Information Sciences 539 (2020) 277–294

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Cross-Attentional Bracket-shaped Convolutional Network for semantic image segmentation

Cam-Hao Hua [a], Thien Huynh-The [b], Sung-Ho Bae [a,*], Sungyoung Lee [a,*]

[a] Department of Computer Science and Engineering, Kyung Hee University (Global Campus), 1732 Deogyeongdae-ro, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Republic of Korea
[b] ICT Convergence Research Center, Kumoh National Institute of Technology, 61 Daehak-ro, Gumi-si, Gyeongsangbuk-do 39177, Republic of Korea

ARTICLE INFO

Article history:
Received 10 December 2019
Received in revised form 5 June 2020
Accepted 7 June 2020
Available online 20 June 2020

Keywords:
Semantic image segmentation
Bracket-shaped Convolutional Neural Network
Cross-Attentional mechanism

ABSTRACT

As perception-related applications are of great importance in industrial production and daily life nowadays, solutions for understanding given images semantically receive numerous attention from the literature. To this end, significant accomplishments have been reached for such pixel-wise segmentation problem thanks to novel manipulations of integrating global context into local details in convolutional neural networks. However, this strategy in the existing work did not exhaustively exploit middle-level features, which carry reasonable balance between fine-grained and semantic information. Therefore, this paper introduces a Cross-Attentional Bracket-shaped Convolutional Network (CAB-Net) to leverage their contribution to the tournament of constructing pixel-wise labeled map. In concrete, fine-to-coarse feature maps of interest from the backbone network are densely combined by an efficient fusion of channel-wisely and spatially attentional schemes in crossing manner, namely Cross-Attentional Fusion, to embed semantically rich features into finer patterns. Continuously, these newly decoded outputs repeat the same procedure round-by-round until shaping a final feature map having finest resolution for complete scene understanding. Consequently, the proposed CAB-Net achieves competitive mean Intersection of Union performance on PASCAL VOC 2012 (83.6% without MS-COCO pre-training), CamVid (76.4%) and Cityscapes (78.3%) datasets.

1. Introduction

Over the last few years, tremendous growth in power of computational resources and quantity of visual data has enabled deep learning to be intensively exploited in diverse computer vision tasks for further industrial applications. To this end, Convolutional Neural Network (CNN), one of the most well-known branches of deep learning, has attracted numerous researchers thanks to its significant performance boost in various problems ranging from categorizing overall content [33,13,8,14] to labeling every single pixel [28,1,31] of images. Specifically, the former is basically referred to as classification task at image level, which can be applied into human activity recognition [17,18], disease progression identification [15], to name a few. Meanwhile, the latter is called semantic segmentation which recognizes predefined objects at pixel level for complete scene understanding.

* Corresponding authors.
E-mail addresses: hao.hua@oslab.khu.ac.kr (C.-H. Hua), thienht@kumoh.ac.kr (T. Huynh-The), shbae@khu.ac.kr (S.-H. Bae), sylee@oslab.khu.ac.kr (S. Lee).

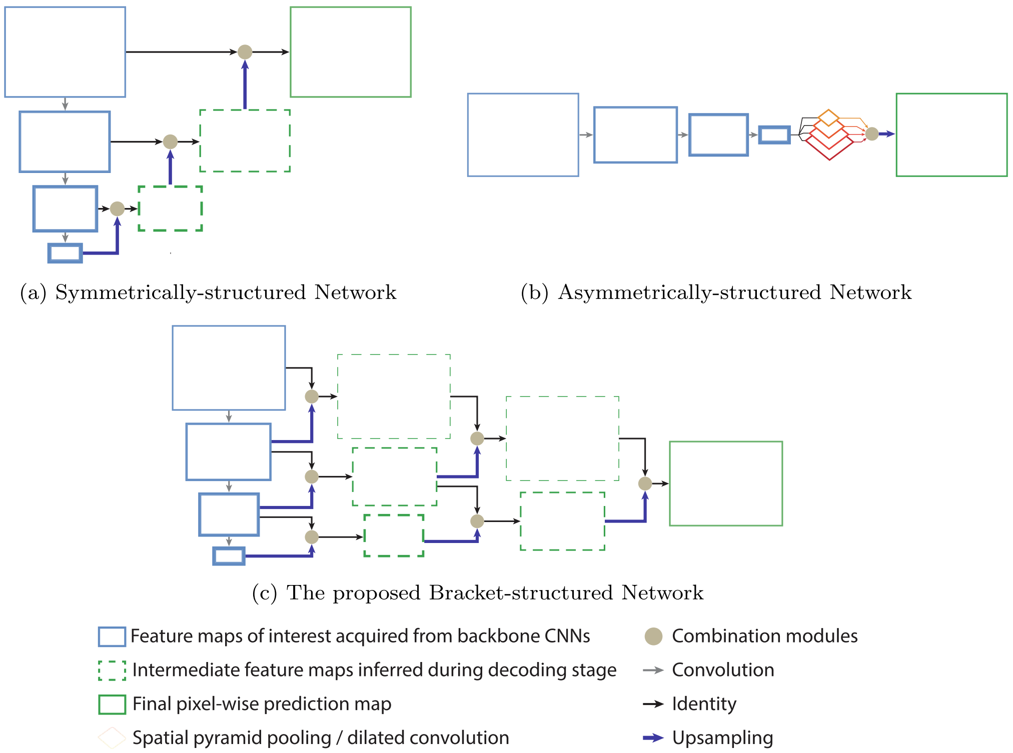
https://doi.org/10.1016/j.ins.2020.06.023
0020-0255/© 2020 Elsevier Inc. All rights reserved.

In fact, such per-pixel labeling problem remains an open research area due to the following reason: the recently rapid development of perception-related applications (e.g., medical image analysis, augmented reality, computational photography, autonomous driving) requires higher pixel-wise categorization performance for retrieving more comprehensive knowledge from the given scenes. For example, segmenting regions of interest semantically from a medical image can provide valuable information such as tumor density (by pixel level) to the physicians for better diagnosis and treatment [2]. As a result, a large amount of semantic segmentation models has been proposed and benchmarked with large-scale datasets [10,4,9] for being efficiently applied into the aforementioned technologies.

Generally, to tackle such pixel-wise grouping problem, most existing approaches have utilized CNN primarily designed for classifying images like VGG [33], ResNet [13], Xception [8], to name a few, as the backbone network to exhaustively exploit its powerful feature representation. In concrete, features learned at shallow layers are finely patterned but weakly semantic due to partial view on the original inputs. Oppositely, features acquired at deeper layers depict abstract appearance (a.k.a., coarse pattern) but carry semantically-rich information due to multiple subsampling stages and larger field of view on the input images, respectively. In other words, following the feedforward process of the CNNs, wherein spatial resolution of the learned feature maps gradually decreases while corresponding channel dimension increases significantly, local details and global contextual information are extracted successively. Besides that, since semantic segmentation framework aims to generate densely labeled output having spatial size same as that of the input, it emerges the following research question: *How to design an optimal upsampling strategy being able to balancedly combines local information (finely patterned features) with global context (semantically rich features) acquired from shallow-to-deep layers of the backbone CNN?*

To address this, Fully Convolutional Network (FCN) [28] – the pioneering model of end-to-end trainable segmentation architecture - utilized skip-connection mechanism to fuse contextual information captured from middle- to high-level layers. Accordingly, to resolve existing drawbacks as well as improve the segmentation performance, numerous efforts have been made in the literature. In terms of network topology, there are two major groups, i.e., *symmetrically-* [1,31,20,25,34,27,29,21,3,26] and *asymmetrically-structured* [23,7,43,38,44,45,24,42,39,40,11] frameworks as shown in Fig. 1a and 1b, respectively.

Regarding the upsampling strategy, Fig. 1a (i.e., *symmetrically-structured* network group) conceptually shows that only the lowest-resolution feature map inferred from the backbone CNN is upsampled step-wisely to form into the highest-resolution prediction map. In addition, during this progress, all the intermediate upsampled features are refined by counterparts learned at encoding stage via certain combination mechanisms. Note that the whole structure of this approach can be also referred to as a U-/Ladder-shaped architecture. Similarly, demonstration of the *asymmetrically-structured* network group in


(a) Symmetrically-structured Network

(b) Asymmetrically-structured Network

(c) The proposed Bracket-structured Network

Feature maps of interest acquired from backbone CNNs
Intermediate feature maps inferred during decoding stage
Final pixel-wise prediction map
Spatial pyramid pooling / dilated convolution
Combination modules
Convolution
Identity
Upsampling

Fig. 1. Conceptual diagrams of (a) *symmetrically-structured* network, (b) *asymmetrically-structured* network, and (c) the proposed *Bracket-structured* network. Spatial and channel dimension of the feature maps are represented by corresponding perimeter and border thickness, respectively.

Fig. 1b delivers the same idea in which the decoding process initiates from the coarsest feature map for further spatial pyramid pooling and upsampling steps. It can be observed from these architectures that feature maps obtained at middle layers of the backbone CNN are not utilized significantly. Clearly, they just perform a single role of excluding contextual ambiguities from the corresponding upsampled versions in the *symmetrically-structured* group. Meanwhile, they even contribute nothing during the decoding stage in the *asymmetrically-structured* group.

Accordingly, motivated by the fact that the middle-level features are not exploited thoroughly in the existing work, this paper proposes a Cross-Attentional Bracket-shaped Convolutional Neural Network, namely CAB-Net, to leverage their contributions to the process of retrieving final pixel-wise prediction map. In concrete, we hypothesize that each middle-level feature map keeps a reasonable balance between fine-grained details and semantic information, which is capable of simultaneously refining pixel-wise context of coarser-resolution feature maps and eliminating ambiguities existent at finer-resolution versions. Hence, as conceptually depicted in Fig. 1c, not only the coarsest one, every feature map of interest (except for the one with highest spatial dimension) is now upsampled and then combined with the adjacent higher-resolution version to produce finer outputs. Continuously, these newly decoded features repeat the same procedure round-by-round until shaping the final feature map of finest resolution. Notably, to strengthen the semantic contexts in the results of the combination between two adjacent feature maps, we embed into the mergers a cross-attentional scheme inspired from SENet [14]. Briefly, major contributions provided by this study are encapsulated as below:

- We propose a Bracket-shaped CNN to leverage the exploitation of middle-level feature maps by exhaustively pairing adjacent ones through attention embedded combination modules. Such routine repeats round-by-round until the final prediction map of densely enriched semantic contexts is retrieved.
- We define an effective approach of combining two neighboring feature maps having different resolutions by adopting a cross-attentional fusion mechanism, namely CAF module. The major objective is to properly fusion semantically rich information (of the lower-resolution inputs) with finely patterned features (of the higher-resolution versions) for the outputs.
- We train and evaluate the proposed architecture on popular semantic segmentation datasets including PASCAL VOC 2012 [10], CamVid [4], and Cityscapes [9], on which the performance is competitive with well-known deep learning models in the literature.

It is worth noting that compared to the existing work [16], this paper offers four remarkable activities: (i) utilizes ResNet-101 [13] instead of VGG-16 [33] as the backbone CNN to extract more informative features for the decoding stage; (ii) introduces the powerful CAF module which suits the proposed Bracket-shaped network more reasonably than the counterpart in [16]; (iii) provides in-depth analyses on the Bracket structure's capability of leveraging information from middle-level features; and (iv) experiments with larger-scale datasets for a more comprehensive evaluation.

The rest of this paper is organized as follows. The review of related work utilizing deep learning is detailedly given in Section 2. Next, Section 3 describes the proposed CAB-Net thoroughly. Afterwards, evaluation on benchmark datasets and corresponding discussions are elaborated in Section 4. Finally, this work is concluded in Section 5.

2. Related work

2.1. Symmetrically-structured networks

Models belonging to this group mainly follow the framework of symmetric encoder-decoder. Conceptually, backbone CNNs pre-trained on large-scale dataset for classification are often utilized as the encoder for gradually extracting from local to global features. Subsequently, the decoder is constructed in layer-wise reversed manner based on the encoder's inherent architecture to progressively integrate semantic contexts into local details in the final per-pixel segmentation map. It is obvious that involving extracted features at the encoding stage to the upsampling process at the decoder can significantly boost the pixel-wise labeling performance.

Typically, SegNet [1] made use of max pooling indices from pooling layers at the backbone VGG [33] to directly locate pixels of lower-resolution feature maps in the corresponding upsampled versions. Then, the convolution layers with specific settings same as the counterparts at the encoder are subsequently applied. This strategy enables important features to be sustained throughout the network but clearly ruins the correlation between neighboring pixels.

Meanwhile, for the purpose of maintaining localization precision while being able to learn meaningful contextual information, various combination styles between corresponding feature maps at the decoder and encoder in the upsampling process were introduced. They can be either simple concatenation technique as in U-Net [31] or specialized modules as in G-FRNet [20] and GFF [25]. These schemes are shown to yield promising performance in many benchmarks but require high footprint for training due to large depth-sized tensors. Accordingly, Tian et al. [34] defined an efficiently data-dependent upsampling scheme to reduce the necessity of exhaustively involving high-dimensional features in the backbone CNN during the decoding process.

On the other hand, instead of concatenation, Feature Pyramid Network [27], SwiftNetRN-18 [29] and LDN [21] introduced a Lateral Connection Module (LCM), wherein an upsampled feature map is element-wisely added to the corresponding version extracted from the encoder before being fed into learnable convolution filters. This module is executed step-by-step until forming the final prediction map. Also based on the core of pixel-wise summation, Bilinski et al. [3] proposed the scheme of Dense Decoder Shortcut Connections (containing Encoder Adaptation, Fusion, and Semantic Feature Generation modules) to enhance meaningful contexts captured from features at multiple scales. Similarly, RefineNet [26] further took into account additional refinement units (consisting of Residual Convolution Unit and Chained Residual Pooling) to comfort the training process and acquire global contextual information accurately.

2.2. Asymmetrically-structured networks

Deep learning models categorized into this group contain a specialized upsampling strategy for aggregating contextual information from multiple strides without involving multi-level feature maps of the encoder.

Typically, attaching Recurrent Neural Network (RNN) to the pretrained CNN is an alternative way since the RNN can robustly represent the dependencies of pixel-level information with respect to global context through an evolutionary process of learning from hidden states. In concrete, RLS [23] presented the series of densely horizontal-vertical sweeping and level set method, respectively, for such evolutionary learning strategy.

Concurrently, there is another suggestion that equipping each neuron with a larger field of view on lower-level feature maps enables the semantically rich information to be captured more effectively without sacrificing spatial resolution abundantly. Thus, recently proposed models like DeepLab [7], FSSNet [43], DenseASPP [38], PSPNet [44], and SSPP-ES [45] utilized dilated (atrous) convolution layers, which have larger receptive field but similar amount of trainable weights compared to those of the original versions. Subsequently, aggregating the extracted features learned from various dilation rates, so-called spatial pyramid mechanism, is the key factor earning impressive segmentation performance in these networks.

On the other hand, to avoid facing the complicated padding issue caused by the dilated convolution, a depth-wisely attentional mechanism in PAN [24], EncNet [42], BiSeNet [39], and DFN [40] is additionally exploited along with their hybrid architectures. In specific, PAN [24] introduced a Global Attention Upsample module, wherein average spatial-based pooling is applied to the features acquired at high-level layers of the encoder. Furthermore, Zhang et al. [42] took advantages of dilation approach as well as attention scheme to design an EncNet, which is composed of (i) a backbone CNN with dilated convolutions for extracting features and (ii) a Context Encoding for embedding semantic details back into the encoded features, to accurately classify every pixel. Besides that, a similar two-stream approach called BiSeNet [39] is introduced in the literature. It consists of cost-efficient Feature Fusion and Attention Refinement Modules in the main and auxiliary context paths, respectively, for the improvement of both accuracy and inference speed. Meanwhile, DFN [40] was proposed to enhance consistent appearance of segmented objects, for which Channel Attention Blocks were designed to re-weight feature responses of the finer-resolution maps by semantically richer context in the adjacent coarser ones. Furthermore, DANet [11] applied both spatial- and channel-based attention schemes onto the deepest-level feature map in parallel, of which the outputs are summed for subsequent learning layers followed by a final softmax classifier.

In this study, for jointly learning valuable information from the adjacent feature maps, we adopt not only the channel-wisely but also the spatially attentional blocks to seamlessly combine semantically rich context with finely patterned features while ensuring an effective training process. Notably, our work utilizes the two types of attention mechanism in crossing manner for the connections between all-level features along the decoding stage, which is different from the aforementioned DANet [11].

3. Proposed methodology

This section delivers details of the proposed CAB-Net, with corresponding demonstration in Fig. 2, for semantic segmentation as follows. We firstly elaborate the decoding process of Bracket-shaped structure for generating the pixel-wise prediction map. Then, an in-depth description of the proposed CAF module that combines two adjacent feature maps of interest is delivered.

3.1. Bracket-shaped Convolutional Neural Network

It is worth noting that the proposed Bracket-shaped decoder can be easily fitted to any classification-based CNNs. In this paper, we employ ResNet-101 [13] pretrained with ImageNet dataset [32] as the default backbone CNN (encoder) of the proposed architecture in Fig. 2 to extract meaningful features from the inputs. Accordingly, four encoded feature maps of specialized convolution blocks are taken into account for the Bracket-shaped decoder. Note that spatial resolution of these features is reduced by half (i.e., they have strides of 4, 8, 16, and 32, respectively) while their channel dimension gets significantly deeper after each convolution block along the feedforward process. To be convenient, the selected features are respectively named *convmap-1* (with spatial size having stride of 4 compared to that of the original input and depth of d1), *convmap-2* (8 and d2), *convmap-3* (16 and d3), and *convmap-4* (32 and d4) as manifested in Fig. 2.

Next, every of those feature maps, except for the finest-resolution one (i.e., *convmap-1*), combines with the adjacent higher-resolution version through the CAF module to generate an output having same dimension as that of the latter. In other words, the utilization of all the middle-level feature maps (e.g., *convmap-2* and *convmap-3*) is leveraged since each

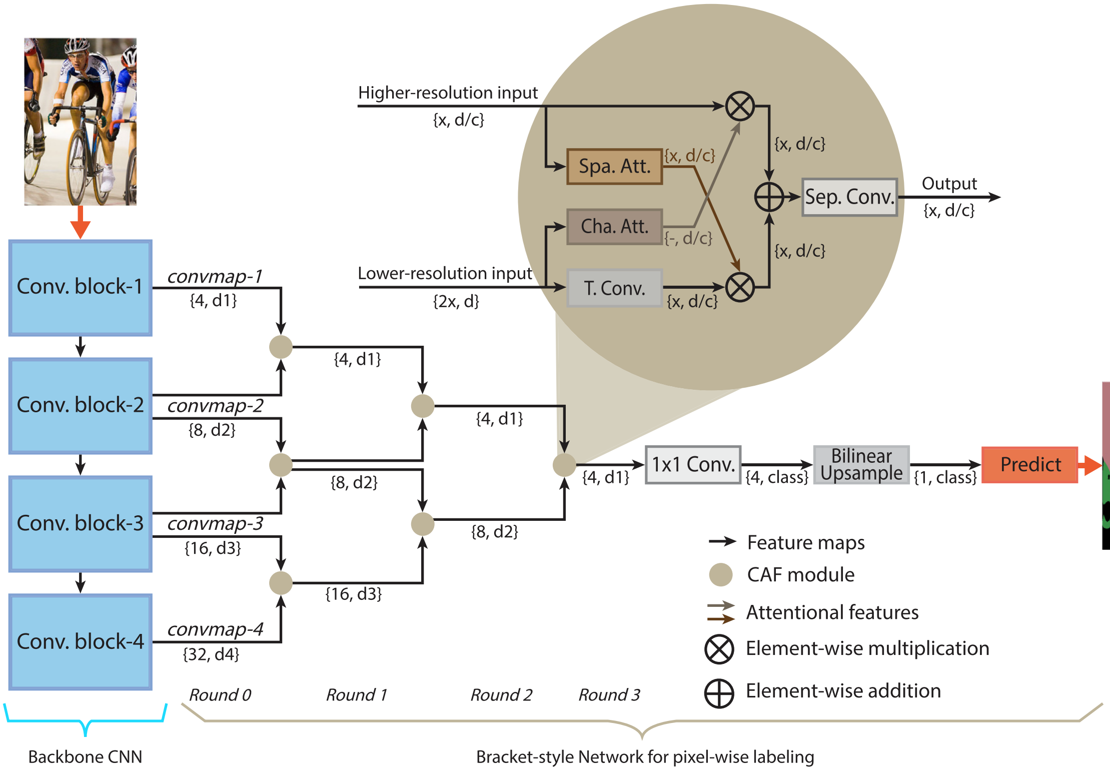


Fig. 2. Architecture of the proposed CAB-Net. Given an input image fed into the backbone CNN containing series of predefined convolution blocks, final outputs of these blocks have strides of 4, 8, 16, and 32, respectively. Subsequently, these chosen feature maps (namely *convmap-1*, *convmap-2*, *convmap-3*, *convmap-4*) are utilized in the decoding process for pixel-wise labeling. In brief, these fine-to-coarse feature maps (represented by black arrows) are densely combined via the Cross-Attentional Fusion modules to produce outputs, which continuously pass through the same procedure until one final prediction map is retrieved. As for the obtained segmentation map, every pixel is assigned an object class within the predefined number of training classes. Since every inferred feature map fuses with its adjacent finer-resolution map at each round and the total number of feature maps decreases by one round-by-round, such process is named Bracket-shaped network. Note that the symbol $\{x, d\}$ attached to each arrow indicates the corresponding feature map having stride of x (i.e., its spatial dimension is $1/x$ as large as that of the input image) and d channels. Meanwhile, x = – (dash) means that the spatial size equals to 1×1. Besides that, 'T. Conv.' and 'Sep. Conv.' stand for *Transpose* and *Separable Convolution* layer while 'Spa. Att.' and 'Cha. Att.' represent Spatially and Channel-wisely Attentional blocks, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

one simultaneously plays two roles, i.e., (i) integrating global context at a certain level to the final prediction map by upsampling itself, and (ii) refining semantically richer information of upsampled version of the adjacent coarser-resolution map by embedding its finer patterned features. Hence, it is clear that given n encoded feature maps chosen from the backbone CNN, such connection style infers $n-1$ outputs at the first round of the proposed Bracket-shaped decoder. Subsequently, as such routine iterates, total number of semantic feature maps decreases by one while average spatial dimension increases round-by-round until the final pixel-wise prediction map is retrieved.

In specific, let $\mathbf{F}_i^r$ be the i^{th} feature map at r^{th} round, where $i = 1, \ldots, n-r$ and $r = 0, \ldots, n-1$. Note that $i = 1$ indicates the feature map having highest resolution and $i = n-r$ corresponds to the lowest. Accordingly, $\mathbf{F}_1^0$ refers to as *convmap-1* and $\mathbf{F}_4^0$ corresponds to *convmap-4* at the initial 0^{th} round as presented in Fig. 2. Then, the feature maps of next rounds are continuously determined by

$$\mathbf{F}_i^r = \mathcal{C}(\mathbf{F}_i^{r-1}, \mathbf{F}_{i+1}^{r-1}), \qquad r \geqslant 1 \tag{1}$$

where $\mathcal{C}(.)$ is the CAF module, which is fully depicted at Section 3.2. It is obvious that until the $(n-1)^{\text{th}}$ round, the final prediction map containing finely patterned features fulfilled by semantically rich context is acquired.

Fundamentally, there are two apparent advantages of using the Bracket structure: (i) missing or ambiguous details are suppressed significantly since every upsampled feature map is always refined by the equivalent-sized version of finer-grained information; and (ii) semantically rich information is densely enhanced in the final per-pixel segmentation map because such upsampling plus dense mixture strategy is applied for all fine-to-coarse feature maps at all rounds during the decoding stage.

3.2. Cross-attentional fusion module

Obviously, the ultimate purpose of the upsampling process in a semantic segmentation architecture is to ensure that visual details in the upsampled version of certain coarse-resolution feature map are capable of bearing the semantic information reasonably. To achieve this, refining local ambiguities appearing in the upsampled ones by effectively involving well-representational knowledge in the corresponding encoder's feature maps plays a critical role in many model designs.

To efficiently coordinate with the capability of the proposed Bracket-structured decoder, we define the CAF module built upon the attentional mechanism (inspired from Refs. [6,14]) followed by *Separable Convolution (Sep. Conv.)* layers [8] as illustrated in the copper circle in Fig. 2. Concretely, each CAF unit comprehensively exploits contextual information from the two inputs of different scales by Channel-wisely Attentional (Cha. Att.) and Spatially Attentional (Spa. Att.) blocks. The former depth-wisely re-weights the lower-level features of the higher-resolution input by using semantically richer features of the lower-resolution counterpart. The latter spatially re-calibrates features of the upsampled lower-resolution input by utilizing finer patterns of the higher-resolution one. As a consequence, fusioning the acquired cross-attentional information can infer fruitful feature maps for the dense prediction.

The first block (a.k.a., Cha. Att.) is executed according to the fact that the coarser and deeper feature map possesses much more informative context along the depth dimension than the finer and shallower one does. Therefore, it is beneficial to the final performance when conducting the impact of that channel-wise semantic information on the fine-grained features in feedback-like manner. To address this, we employ a depth-wise calibration strategy inspired from the attention mechanism in Ref. [14] as demonstrated in Fig. 3a. It is worth noting that the DFN [40] also adopts such scheme. Specifically, all feature responses are re-weighted through a step of cross-channel learning on the global pooling information, which is acquired from the considered feature map itself [14] or that concatenated with the adjacent scale [40], a.k.a. self-attention. Differently, our approach collects informative attributes across channels of the lower-resolution input only in order to depth-wisely enhance corresponding responses of the higher-resolution one, a.k.a. cross-attention. As shown in Fig. 3a, each channel of the coarser-resolution input, of which the spatial and depth size are $\frac{1}{2x}$ as large as that of the original image and d respectively, is averaged spatially to form a vector having length of d. Accordingly, this vector, namely $\mathbf{g} \in \mathrm{R}^d$, compactly carries reasonable information in channel-wise manner as follows

$$\mathbf{g} = \left[g_1(\mathbf{F}_{i+1}^{r-1}), \ldots, g_d(\mathbf{F}_{i+1}^{r-1})\right]^T \tag{2}$$

where $g_d(.)$ is the *Channel Pool* operation taking place on d^{th} channel of a considered feature map f, of which the corresponding formulation is

$$g_d(\mathbf{F}_{i+1}^{r-1}) = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{F}_{i+1\,h,w,d}^{r-1} \tag{3}$$

where (h, w) indicates pixel coordinates of the considered feature map $\mathbf{F}_{i+1}^{r-1}$ having spatial resolution of $H \times W$. Consequently, every channel of the lower-resolution input has its own representative response in the d-length vector $\mathbf{g}$. Next, to correspondingly express the relative importance degree of each channel onto that of the higher-resolution input, we firstly filter the vector $\mathbf{g}$ by two *Fully Connected (FC)* layers with *ReLU* activation in the middle so as to learn cross-channel relation-

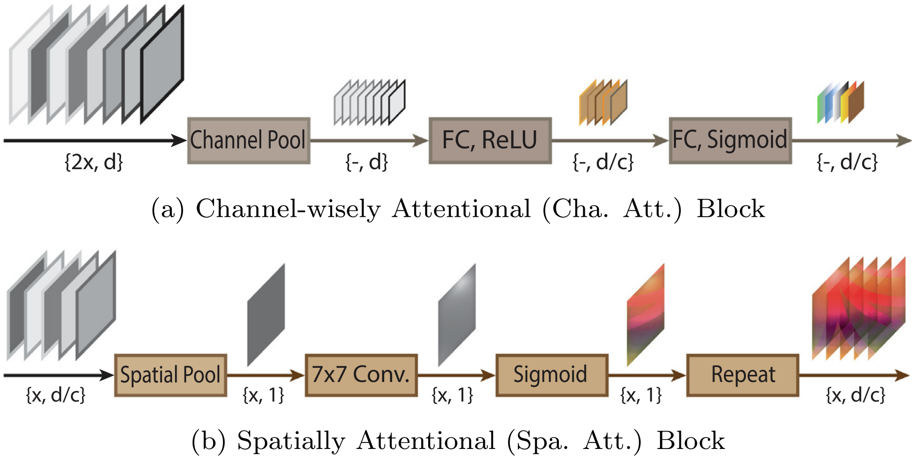


Fig. 3. Details of operators in attentional schemes, i.e., 'Spa. Att.' and 'Cha. Att.'. Note that 'FC' stands for *Fully Connected* layer and '7 × 7 Conv.' indicates one *Convolution* layer having kernel size of 7 × 7.

ships. Notably, the size of applied hidden layers is set to be identical to the number of the higher-resolution input's channels. These learning operations are equivalent to the following equation

$$\boldsymbol{g}_{ca} = \mathbf{W}_{fc_2}\left(ReLU(\mathbf{W}_{fc_1}\boldsymbol{g} + \mathbf{B}_{fc_1})\right) + \mathbf{B}_{fc_2} \tag{4}$$

where $\{\mathbf{W}_{fc_1} \in \mathrm{R}^{\frac{d}{c}\times d},\ \mathbf{B}_{fc_1} \in \mathrm{R}^{\frac{d}{c}}\}$ and $\{\mathbf{W}_{fc_2} \in \mathrm{R}^{\frac{d}{c}\times\frac{d}{c}}, \mathbf{B}_{fc_2} \in \mathrm{R}^{\frac{d}{c}}\}$ are learnable parameters of the first and second *FC* layers, respectively, and $\boldsymbol{g}_{ca}$ is the yielded channel-wise attention feature vector of $\frac{d}{c}$ length. Then, the *Sigmoid* activation $\sigma(.)$ is utilized to rescale values of elements in the vector $\boldsymbol{g}_{ca}$ within the range from 0 to 1. Subsequently, the resulting channel-wisely attentional features are used to modify the responses of the higher-resolution input $\mathbf{F}_i^{r-1} \in \mathrm{R}^{2H\times 2W\times\frac{d}{c}}$ in depth-wise manner as below

$$\mathbf{F}_{i_{ca}}^{r-1} = \left\{ \mathbf{F}_i^{r-1}{}_{:,:,\delta} \otimes \sigma(\boldsymbol{g}_{ca})_\delta | \delta = 1,\ldots,\tfrac{d}{c} \right\} \tag{5}$$

where $\otimes$ symbolizes the element-wise multiplication and $\mathbf{F}_{i_{ca}}^{r-1} \in \mathrm{R}^{2H\times 2W\times\frac{d}{c}}$ is the channel-wisely attentional version of $\mathbf{F}_i^{r-1}$.

The second block (a.k.a., Spa. Att.) is utilized based on the fact that higher-resolution input possess finer spatial patterns, which is profitable for the refinement of local details in the upsampled version of the lower-resolution input. Therefore, we opt for expressing important spatial features of the finer-resolution input into the upsampled partner in the CAF module through a spatially-attentional block exhibited in Fig. 3b. In particular, the finer-resolution input $\mathbf{F}_i^{r-1} \in \mathrm{R}^{2H\times 2W\times\frac{d}{c}}$ is fed into a *Spatial Pool* operation, in which responses at every pixel (h, w) are averaged across channel dimension as follows

$$\mathbf{F}_{isp}^{r-1}{}_{h,w} = \frac{c}{d}\sum_{z=1}^{\frac{d}{c}} \mathbf{F}_i^{r-1}{}_{h,w,z} \tag{6}$$

wherein $\mathbf{F}_{isp}^{r-1} \in \mathrm{R}^{2H\times 2W\times 1}$ is the corresponding output of this operation. Subsequently, one trainable *Convolution* layer having kernel size of 7 × 7 with padding of 3, namely $\mathbf{W}_{7\times 7}$, followed by the *Sigmoid* activation $\sigma(.)$ is adopted to quantify the locally spatial dependencies as below formulation

$$\mathbf{F}_{i_{s77}}^{r-1} = \sigma(\mathbf{W}_{7\times 7} * \mathbf{F}_{isp}^{r-1}) \tag{7}$$

where $*$ and $\mathbf{W}_{7\times 7} \in \mathrm{R}^{1\times 7\times 7\times 1}$represent the convolution operator and learnable parameters of the above-mentioned kernel, respectively, and $\mathbf{F}_{i_{s77}}^{r-1} \in \mathrm{R}^{2H\times 2W\times 1}$ is defined as the spatially attentional features. It is worth noting that this one-channel map is then repeated by $\frac{d}{c}$ times to be same depth size as that of the higher-resolution input of the CAF module. Simultaneously, the lower-resolution input $\mathbf{F}_{i+1}^{r-1}$ is upsampled into $\mathbf{F}_{i_u}^{r-1}$ using the *Transpose Convolution (T. Conv.)* layer having stride of 2 and the number of filters identical to channel dimension of the higher-resolution input. Notably, as shown in Fig. 2, given *d* as the channel size of the lower-resolution input, then that of the higher-resolution one is *c* times smaller. The upsampling operation can be expressed as below

$$\mathbf{F}_{i_u}^{r-1} = upsample(\mathbf{F}_{i+1}^{r-1}) = \mathbf{W}_u *^u \mathbf{F}_{i+1}^{r-1} + \mathbf{B}_u \tag{8}$$

where $*^u$ is fractionally-strided convolution operation, $\mathbf{W}_u \in \mathrm{R}^{\frac{d}{c}\times 3\times 3\times d}$ corresponds to trainable weights in $\frac{d}{c}$ transposed convolution filters having size of $3 \times 3 \times d$, and $\mathbf{B}_u \in \mathrm{R}^{\frac{d}{c}}$ stands for trainable biases. Finally, from (7) and (8), the spatially attentional version, denoted as $\mathbf{F}_{i_{sa}}^{r-1}$, of the originally upsampled map $\mathbf{F}_{i_u}^{r-1} \in \mathrm{R}^{2H\times 2W\times\frac{d}{c}}$ is obtained by multiple operations of Hadamard product as follows

$$\mathbf{F}_{i_{sa}}^{r-1} = \left\{ \mathbf{F}_{i_u}^{r-1}{}_{:,:,\delta} \otimes \mathbf{F}_{i_{s77}}^{r-1} | \delta = 1,\ldots,\tfrac{d}{c} \right\} \tag{9}$$

To this end, both semantically rich information and finely patterned features are exhaustively exploited in cross-attentional manner, i.e., $\mathbf{F}_{i_{ca}}^{r-1}$ and $\mathbf{F}_{i_{sa}}^{r-1}$, respectively. The next step is to integrate them by a simple pixel-wise addition scheme, of which the total result is continuously fed into the *Sep. Conv.* as follows

$$\mathbf{F}_i^r = \mathbf{W}_{sc} * ReLU(\mathbf{F}_{i_{ca}}^{r-1} \oplus \mathbf{F}_{i_{sa}}^{r-1}) \tag{10}$$

where $\oplus$ signifies the element-wise addition, $\mathbf{W}_{sc} = \{\mathbf{W}_{df} \in \mathrm{R}^{\frac{d}{c}\times 3\times 3}, \mathbf{W}_{pf} \in \mathrm{R}^{\frac{d}{c}\times 1\times 1\times\frac{d}{c}}\}$ denotes the sequential execution of $\frac{d}{c}$ depth-wise convolution filters with 3×3 size and $\frac{d}{c}$ point-wise convolution filters with $1 \times 1 \times \frac{d}{c}$ size. It is also worth noting that the *Sep. Conv.* layer defined in this CAF module includes three consecutive operations, i.e., *ReLU* activation, *Sep. Conv.*, and *Batch Normalization* layer [19] (which was not shown in (10) for simplicity). Obviously, compared to using the normal 3×3 convolution, such kind of filter can reduce the number of trainable parameters per layer from $\frac{d}{c} \times 3 \times 3 \times \frac{d}{c}$ to $\frac{d}{c}(3 \times 3 + \frac{d}{c})$ while effectively maintaining the capability of shrinking unexpected artifacts appearing in such decoding process. Remarkably, we have also found that additionally taking the fusion in (10) with $\mathbf{F}_i^{r-1}$ and $\mathbf{F}_{i_u}^{r-1}$ is not necessary due to trivial performance improvement while being subject to more computation.

In a nutshell, instead of simply adding upsampled version of the coarser input to the naïve finer-resolution one (which may hinder the precise integration of semantically rich features into spatial dimension), taking into account the proposed CAF module can improve the efficiency of context acquirement and corresponding pixel-wise localization.

4. Experiments

In this section, the proposed CAB-Net is intensively experimented on PASCAL VOC 2012 [10], CamVid [4], and Cityscapes [9] datasets to show its effectiveness for applications of vision-based object localization and autonomous driving, to name a few, in the industry. Particularly, we introduce the benchmark datasets at first, then provide details of training configurations, and finally discuss the experimental results of the ablation studies and comparisons with the state-of-the-art methods.

4.1. Benchmark datasets

4.1.1. PASCAL VOC 2012 [10]

This dataset aims to represent 20 semantic object categories common in real world (i.e., groups of person, animal, vehicle and indoor context). Originally, there are 1464 training, 1449 validation, and 1456 testing images of various sizes in this challenge. Moreover, we follow the procedure of [28,1] wherein additional annotations from Semantic Boundaries Dataset [12] are included for increasing the total number of training images to 10,582. Afterwards, the proposed CAB-Net is further fine-tuned with the original training plus validation set before being benchmarked by the test set on a designated testing server.

4.1.2. CamVid [4]

The name of this dataset stands for "Cambridge-driving Labeled Video Database". It is the collection of various road scenes recorded in 10 min by a dashboard camera, which acts as the eyes of an autonomous car. Accordingly, all 701 obtained 720×960 video frames are pixel-wisely labeled given 32 semantic categories. However, to be comparable with previous work, our experiment uses the split of 367 training, 101 validation and 233 testing images with 12 finalized ground-truth labels to evaluate the proposed model.

4.1.3. Cityscapes [9]

This dataset also represents things that an autonomous car should 'see' for understanding urban street scenes semantically. It offers a large pool of 5000 and 20,000 1024×2048 images with fine and coarse annotations, respectively, corresponding to 19 semantic classes through a 50-city itinerary. In this paper, only the set of fine annotations with 2975 training, 500 validating, and 1525 testing images is utilized for the evaluation of the proposed CAB-Net.

4.2. Training configurations

In this work, the proposed CAB-Net is trained using PyTorch framework [30] with two NVIDIA GTX 1080Ti GPUs. The training images are augmented by following strategies: scaling with random factor in {0.5, 0.75, 1.0, 1.25, 1.5, 1.75}; random cropping to pre-specified size (513×513 for PASCAL VOC 2012, 360×480 for CamVid, and 768×768 for Cityscapes); randomly horizontal flipping; and channel-wise normalization with zero mean and standard deviation of one. Besides that, random Gaussian noise and rotation in $[-10^\circ, 10^\circ]$ are further involved in experiments with PASCAL VOC 2012. Moreover, weight decay coefficient is set to $1e-5$ for promoting the proposed model's generalization capability. Note that we use the batch size of 12, 16, and 6 for PASCAL VOC 2012, CamVid, and Cityscapes, respectively.

About the backbone CNN, we apply the powerful ResNet-101 [13] as mentioned before. Concretely, final outputs of the 1st, 2nd, 3rd and 4th residual blocks (with d1 = 256, d2 = 512, d3 = 1024, d4 = 2048, respectively) are taken into account for the decoder.

Then, we adopt the optimization strategy of Chen et al. [7] to minimize the total softmax loss with respect to the CAB-Net's parameters. In short, stochastic gradient descent with momentum of 0.9 is applied together with the 'poly' learning rate decay schedule, wherein learning rate at the i^{th} iteration equals to the initial learning rate (which is set at 0.01 in this work) multiplied by $\left(1-\frac{i}{\max_i}\right)^{0.9}$. Correspondingly, pretrained weights of the backbone network are fine-tuned with the contemporary learning rate multiplied by 0.01.

Finally, we train the CAB-Net with PASCAL VOC 2012, CamVid, and Cityscapes in 50, 500, and 250 epochs, respectively. The mean Intersection of Union (mIoU) metric is used for performance evaluation. In particular, let us denote p_{xy} as the pixel belonging to ground truth label x is predicted to be of label y, and L as the total number of labels, the mIoU is determined by

$$mIoU = \frac{1}{L}\sum_{x=1}^{L}\frac{p_{xx}}{\sum_{y=1}^{L}p_{xy} + \sum_{y=1}^{L}p_{yx} - p_{xx}}$$

Moreover, a multi-scale test strategy is conducted for the final comparison with the state-of-the-arts, which also report their experimental results applying the same procedure. In concrete, every original test image and its variously scaling (i.e., with factors of {0.5, 0.75, 1.25, 1.5, 1.75} compared to the original size) and horizontal-flipping versions are fed into the built network. The final prediction scores are then averaged from those of all the obtained outputs. Compared to the single-scale test approach, the multi-scale one is capable of boosting mIoU by 1.0–3.5% approximately depending on the dataset as reported in Tables 3–5, but trading off a much more expensive computation.

4.3. Ablation Study

For the ablation study, we use the training plus augmentation (10,582 images) and validation (1449 images) sets of PASCAL VOC 2012 for the evaluation. In this section, we firstly study how the channel-wisely and spatially attentional mechanisms coordinates with the Bracket-shaped architecture. Next, we examine the impact of backbone CNNs with various capacities on the segmentation performance. Finally, we demonstrate how the proposed CAB-Net represents semantic details along the decoding process through the visualization of manifold feature maps.

4.3.1. The coordination between Bracket-shaped Network and CAF-based Connections for leveraging middle-level features

Let's consider a middle-level feature map $\mathbf{F}_i^r$ (with $1 < i < n - r, \forall r$) which plays different roles in two adjacent CAF modules because of the Bracket-shaped connection manner, i.e., the lower-resolution input of $\mathcal{C}_1(\mathbf{F}_{i-1}^r, \mathbf{F}_i^r)$ and the higher-resolution input of $\mathcal{C}_2(\mathbf{F}_i^r, \mathbf{F}_{i+1}^r)$. In the CAF block $\mathcal{C}_1, \mathbf{F}_i^r$ contributes its finer representation via the learnable *T. Conv.* layer and the depth-based semantic information via the channel-wisely attentional mechanism to be adjusted by and re-calibrate the remaining input $\mathbf{F}_{i-1}^r$, respectively. Meanwhile, in the CAF module $\mathcal{C}_2, \mathbf{F}_i^r$ takes a reversed role in which its finely patterned features and neural units are employed to spatially refine and be re-calibrated in depth-wise manner by the partner $\mathbf{F}_{i-1}^r$, respectively. Consequently, each middle-level feature map is exhaustively exploited as both roles of coarser- and finer-resolution features for comprehensively embedding semantic into fine-grained details on the tournament of inferring the final pixel-wise prediction map.

We quantitatively prove the advantage of cross-attentional mechanism in the Bracket-style decoding procedure by validating various settings in Table 1. Compared to the baseline combination, performance improvement introduced by the embedded attentional mechanisms is considerable with 1.13% for spatial-based and especially 2.72% for channel-based attentions. Furthermore, by combining these two attention types in crossing manner, the mIoU is further elevated by 1.0% approximately. This implies the powerful coordination between Bracket-structured network and the CAF-based connections for leveraging the capability of embedding semantically contextual information into finely patterned features.

Moreover, from the reported number of parameters in Table 1, it can be realized that the utilization of one simple 7×7 convolution kernel in each Spa. Att. block has nearly no impact on the model complexity. Meanwhile, employing *FC* layers in Cha. Att. modules increases the number of trainable parameters by approximately 15.8% due to large channel size of processed tensors. Obviously, it is worth trading off such minor complexity increment for an overall mIoU improvement of 3.64%, which is significant in the semantic segmentation problem.

4.3.2. The contribution of backbone CNN to final performance

To this end, we further exploit the contribution of backbone CNN to the final performance in terms of mIoU. Specifically, VGG-16 [33] and Xception-65 [8] are utilized as an alternative to our main backbone ResNet-101 [13]. Besides that, the shallower version, i.e., ResNet-50, is included in this experiment for providing further insights into the impact of varying-deep features on the pixel-wise segmentation performance.

In general, the more capacities an architecture has, which means superior representations of deep features are achieved, the better the segmentation performance in terms of mIoU gets. As reported in Table 2, the gap of mIoU between employing ResNet-101 and VGG-16 as the backbone CNN rises to 5.13%. Correspondingly, the model complexity is enlarged as the total number of trainable parameters increases, which is determined by two major factors, i.e., backbone CNN's capacity and depth sizes of the feature maps involved to the Bracket-shaped decoding stage. The former is enumerated in the fourth column of Table 2. It is worth noting that despite possessing more layers than those in ResNet-50, Xception-65 has fewer trainable parameters thanks to the full usage of *Sep. Conv.*, which is more cost-efficient that the conventional version as described in Section 3.2. Regarding the latter, information of considered channel dimension is given in the second column of Table 2,

Table 1
mIoU (%) on Pascal VOC 2012 [10] validation set and number of parameters with various settings of attentional mechanism.

Settings		mIoU (%)	No. parameters
Cha. Att.	Spa. Att.		
		76.73	33.66M
	✓	77.86	33.66M
✓		79.45	38.97M
✓	✓	80.37	38.97M

Table 2
mIoU (%) on Pascal VOC 2012 [10] validation set and number of parameters with different backbone CNNs.

Backbone CNN	Depth sizes {d1, d2, d3, d4}	mIoU (%)	No. parameters		
			Backbone	Bracket	Total
VGG-16 [33]	{128, 256, 512, 512}	75.24	14.72M	7.13M	21.85M
Xception-65 [8]	{128, 256, 728, 2048}	77.96	20.81M	21.06M	41.87M
ResNet-50 [13]	{256, 512, 1024, 2048}	78.27	23.51M	38.97M	62.48M
ResNet-101 [13]		80.37	42.50M	38.97M	81.47M

{d1, d2, d3, d4}: Depth sizes of backbone CNN's feature maps involved to the Bracket-shaped decoding process (abbreviated as 'Bracket' in the fifth column).

from which those retrieved from ResNet have largest sizes compared to the counterparts. This leads to the increment of more hidden nodes and convolution kernels for *FC* and *Sep. Conv.* layers, respectively, in the CAF modules. Therefore, applying ResNet as the backbone CNN results in a much larger number of learnable parameters in the Bracket-structured decoding process (more than 1.85 times compared to the others) as well as the whole architecture (more than 1.5 times) accordingly.

However, as aforementioned that semantically-rich details are essentially encoded in channel-wise manner, the deeper features acquired from ResNet are capable of contributing more generalized and informative context to the decoding step than those of VGG or Xception. In addition, since the proposed Bracket-shaped decoding procedure exhaustively involves such varying-scale feature maps through multiple rounds, depth-wisely representational abilities of those features are marked as strongly influential attributes benefiting the final segmentation performance. Consequently, it can be observed from Table 2 that employing ResNet-50 as the backbone network introduces a slightly better mIoU (despite fewer layers) while the 101-layer version improves by 2.41% (which is significant in this domain) in comparison with the usage of Xception-65.

4.3.3. Representation of feature maps with respect to different attentional schemes

In this part, we introduce the visual representation of key feature maps for semantic segmentation, comprising $\mathbf{F}_1^1$, $\mathbf{F}_1^2$, and $\mathbf{F}_1^3$, with respect to different attentional schemes. Given an image fed into the proposed CAB-Net, responses in the chosen features are averaged over corresponding channel dimension. Then, those pixel intensities are scaled to the range of [0, 255] as illustrated by two example cases in Fig. 4.

Clearly, since $\mathbf{F}_1^1$ is only decoded by low-level semantic information (from $\mathbf{F}_1^0$ and $\mathbf{F}_2^0$) in our CAB-Net, using naive upsampling followed by element-wise fusion still results in ambiguous features for next rounds. In contrast, applying any attentional mechanisms initializes more meaningful focuses (with high pixel intensities) on object details as shown in the last three rows in the first and fourth column of Fig. 4. Then, in the second round, features $\mathbf{F}_2^2$ inferred by the non-attention strategy (first row in Fig. 4) continue to hardly manifest the regions of interest. Although the Bracket-shaped network structure is able to smoothly embed semantically rich features to spatial context round-by-round, the representation of predefined object categories is still not optimal.

Accordingly, the utilization of spatially and channel-wisely attentional modules has strengthened the capability of expressing vital features and diminishing trivial ones. On the one hand, using Spa. Att. is able to precisely orientate the expressiveness while effectively maintaining spatial context as shown in $\mathbf{F}_1^1$ and $\mathbf{F}_1^2$ (see third row compared to those in the second and fourth rows). On the other hand, involving Cha. Att. blocks can leverage the contribution of semantic details encoded along depth dimension, which plays an important role for class discrimination. However, as can be observed from $\mathbf{F}_1^3$ in the third and fourth rows of Fig. 4, Spa. Att. blocks face difficulty of distributing semantically rich features (of which the pixels should have high intensities) over space. Meanwhile, Cha. Att. blocks show its weakness in highlighting extracted spatial features.

Finally, with the proposed cross-attentional scheme described in Section 3.2, the advantages of both Spa. and Cha. Att. modules are comprehensively combined for better differentiation and localization of objects' features. In specific, feature maps in the last row of Fig. 4 perform the best coordination between semantically rich features and corresponding spatial context. For instance, compared to the counterparts, semantic features of the horse's body in the right $\mathbf{F}_1^2$ are expressed and localized more impressively, which leads to better representation of attentional responses in the subsequent $\mathbf{F}_1^3$.

4.4. Comparison with state-of-the-art methods

4.4.1. PASCAL VOC 2012

The experimental performance on test set is quantitatively reported in Table 3. It can be observed that the proposed approach achieves competitive mIoU of 83.6% compared with that of the state-of-the-arts. Regarding the class-wise results, our CAB-Net attains the top performance with significant margin (up to 3.7%) for 11/20 semantic objects ranging from small to large scale. Meanwhile, state-of-the-art results of the remaining labels are shared between deep models applying dilated convolution-based operators such as EncNet [42], PSPNet [44], and WideResNet [37]. Another noteworthy methodology called Tree-structured Kronecker CNN (TCKN) [36] adopted Kronecker product as the custom convolutional layers, which

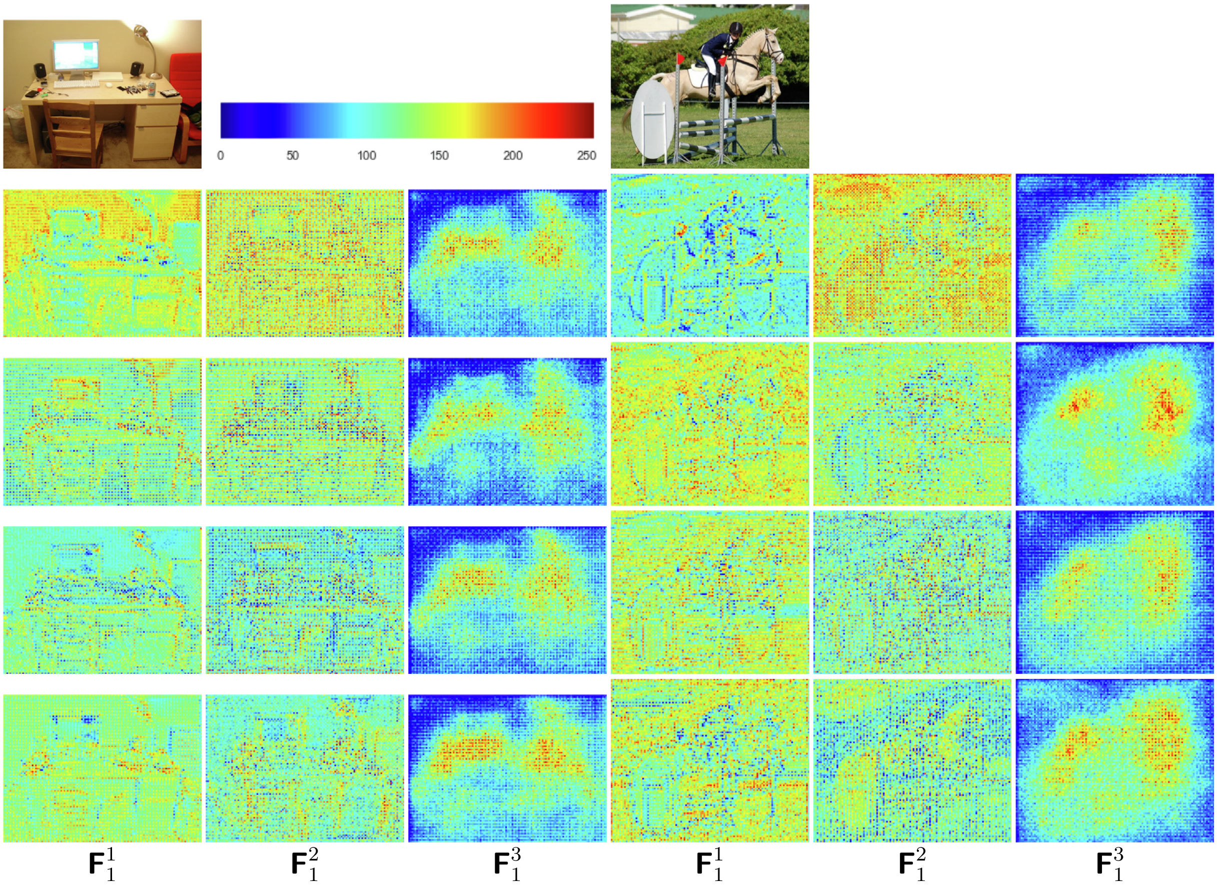


Fig. 4. Representation of key feature maps (i.e., $\mathbf{F}_1^1$, $\mathbf{F}_1^2$, and $\mathbf{F}_1^3$ from left to right) extracted by our CAB-Net with respect to different attentional schemes. Note that the responses presented in the feature maps are averaged over the depth dimension. Top row: example raw images in PASCAL VOC 2012 [10] validation set and a color-intensity indicator; 2nd row: no Cha. Att. and Spa. Att.; 3rd row: only using Spa. Att.; 4th row: only using Cha. Att.; and last row: applying both Cha. Att. and Spa. Att. in the connection blocks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

nails the second-best overall performance in Table 3. Differently, by employing the *Sep. Conv.* layers under the unique Bracket-shaped structure, the proposed model is able to give superior achievements over those networks. In specific, the exhaustive employment of middle-level features during the inference process is able to continuously refine the integration of semantic context to high-resolution representation. As a consequence, the details of various scales are managed more efficiently. Besides that, thanks to the cross-manner operation of multiple CAF blocks along the bracket-style decoder, the proposed architecture outperforms the DANet [11] (which uses dual attention applied only to the highest-level feature in parallel fashion) by 1.0%, which is significant in such a competitive semantic segmentation topic.

Moreover, typically visual results exhibited in Fig. 5 have shown the effectiveness of the CAB-Net in partitioning multiple categories of different scales. Additionally, compared to the outputs introduced by B-Net-VGG-LCM [16], the proposed network can reason better pixel-wise labeling performance, especially the *bird* and *chair* classes. However, our model still fails in precisely segmenting objects which contain interior gaps (void labeled regions in ground-truth map) such as the light-brown chair and the horse's body parts overlapped by fence in the fourth and fifth rows of Fig. 5, respectively. In addition, a very small-sized airplane located at right side of the input image in the first row is not segmented in the prediction map. We argue that the largest-resolution feature maps involved in the proposed decoding procedure have stride of 4 is the major reason for several improper representations of those small spaces for complete scene learning. Accordingly, taking into account features with stride of 2 may bring in finer details for better local context learning. Nevertheless, such approach compromises on the usage of smaller mini-batch size during training due to higher number of operations and model complexity, which subsequently makes the overall performance even worse.

4.4.2. CamVid

It can be observed from the Table 5 that the proposed CAB-Net obtains state-of-the-art mIoU of 76.4%. Regarding per-class performance, our network reaches state-of-the-art class-wise IoU in 10 (*building*, *tree*, *sky*, *car*, *sign-symbol*, *road*, *pedes-*

Table 3
Comparison of per-class IoU and mIoU (%) on Pascal VOC 2012 [10] test set. **Boldface** numbers indicate the best performance at each class.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU (%)
FCN [28]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
B-Net-VGG-LCM [16]	92.0	42.9	92.3	73.3	77.5	91.4	86.4	91.5	42.7	81.9	61.6	84.4	85.8	88.4	90.1	65.5	86.4	60.0	86.1	72.5	78.5
G-FRNet [20]	91.4	44.6	91.4	69.2	78.2	95.4	88.9	93.3	37.0	89.7	61.4	90.0	91.4	87.9	87.2	63.8	89.4	59.9	87.0	74.1	79.3
DDSC [3] (ss)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	81.2
WideResNet [37]	94.4	72.9	94.9	68.8	78.4	90.6	90.0	92.1	40.1	90.4	71.7	89.9	93.7	**91.0**	89.1	71.3	90.7	61.3	87.7	78.1	82.5
PSPNet [44]	91.8	71.9	94.7	71.2	75.8	95.2	89.9	**95.9**	39.3	90.7	71.7	**90.5**	94.5	88.8	89.6	**72.8**	89.6	64.0	85.1	76.3	82.6
DANet [11]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	82.6
DFN [40]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	82.7
EncNet [42]	94.1	69.2	**96.3**	**76.7**	**86.2**	96.3	**90.7**	94.2	38.8	90.7	73.3	90.0	92.5	88.8	87.9	68.7	**92.6**	59.0	86.4	73.4	82.9
TKCN [36]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	83.2
CAB-Net (ss)	94.9	69.2	92.9	73.1	79.5	95.8	87.4	94.1	39.7	87.5	72.4	90.6	92.4	86.4	89.1	68.3	90.8	64.3	88.7	**78.5**	82.5
CAB-Net	**96.0**	**75.6**	94.3	69.1	79.9	**97.1**	89.8	94.8	**40.4**	**91.2**	**74.6**	89.4	**94.7**	87.2	**91.7**	69.6	92.1	**65.5**	**88.8**	76.9	**83.6**

ss: single-scale testing strategy; –: no data recorded in the original work.

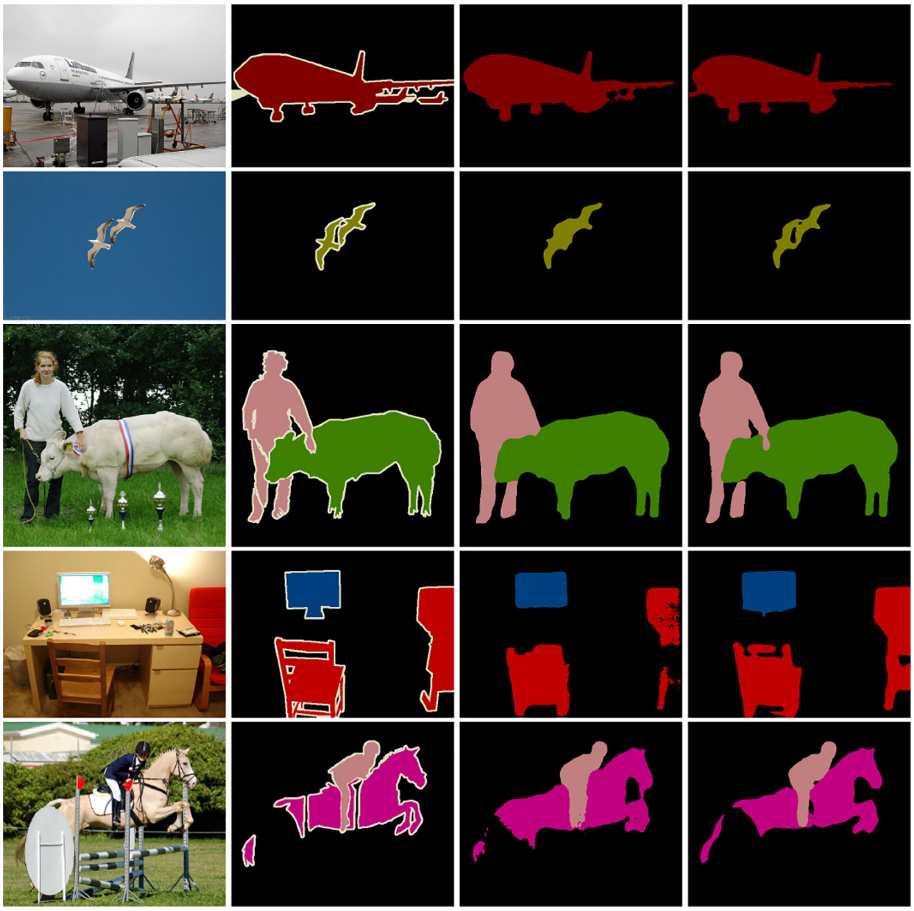


Fig. 5. Several qualitative results on Pascal VOC 2012 [10] validation set. Left to right: original images, ground-truth labels, results of B-Net-VGG-LCM [16], and our CAB-Net.

trian, *fence*, *sidewalk*, and *bicyclist*) out of totally 11 semantic labels. Remarkably, significant margins ranging diversely from 0.2% to 22.7% are gained at these categories in comparison with the corresponding second places. These achievements show that the exhaustive utilization of middle-scale feature maps in terms of Bracket-structured manner can effectively embed semantic information to the representation of medium- to small-sized objects (e.g., *tree*, *pedestrian*, *sign-symbol*) while providing precise annotations for large-sized ones (e.g., *building*, *car*). However, the segmentation performance for the *pole* label is still lower than that of B-Net-VGG-LCM [16] by 6.5%. Apparently, due to the heavily imbalanced class issue, it is challenging to produce the performance higher than 50% for this category, even in the existing work. In addition, despite the fact that feature maps with stride of 2 are also taken into account in Ref. [16] allows fine details like *pole* to be acquired more effectively, the corresponding mIoU is significantly lower than that of the proposed CAB-Net by 10%. This arguably implies that, as discussed at previous sub-section, the involvement of too large-sized tensors during the decoding process should encounter the issue of training convergence and following non-optimal test performance.

Besides that, several visual results compared with those of B-Net-VGG-LCM [16] and the corresponding ground-truth maps are illustrated in Fig. 6. Obviously, the proposed architecture is able to reduce the wrong labeling between *truck* (in purple) and *building* (in red) as displayed in the 2^{nd} row; *sidewalk* (in blue) and *road* (in magenta) as shown in the 1^{st} and 3^{rd} row, respectively. This infers that the discrimination between similar-sized objects is performed better thanks to the usage of a more robust backbone CNN, the newly defined connection module called CAF and its powerful coordination with the Bracket-style decoding structure.

4.4.3. Cityscapes

The quantitative and qualitative benchmark results of this dataset from the evaluation server are presented in Table 4 and Fig. 7, respectively. The proposed CAB-Net achieves a competitive mIoU of 78.3%, where the performance of semantically recognizing motion objects like *car, truck, bus*, and *motorbike* is superior over that of the state-of-the-art methods by a large margin (up to 1.4%). The performance of remaining categories, except for small-scale *traffic light* and *sign symbol*, has average

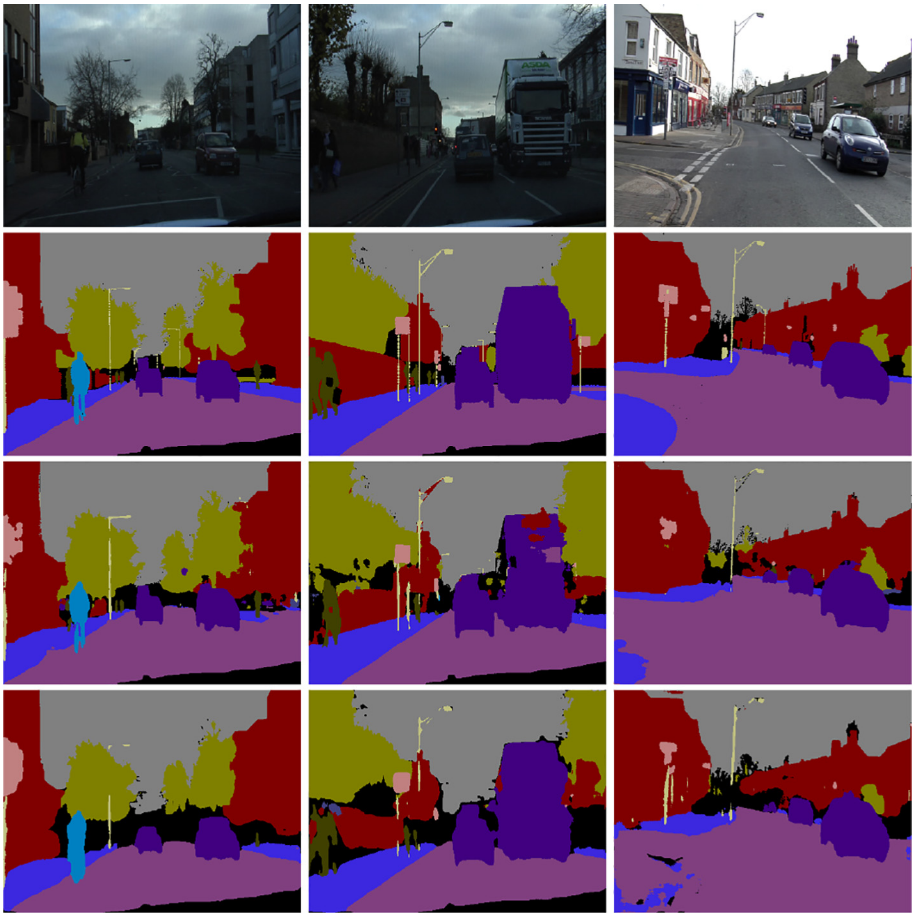


Fig. 6. Several qualitative results on the CamVid [4] dataset. First row: original images; 2nd row: corresponding ground truth labeled maps; 3rd row: results inferred by B-Net-VGG-LCM [16]; and last row: our CAB-Net.

lower IoU of 1.5% approximately in comparison with that of the state-of-the-arts such as PSPNet [44] and SSPP-ES [45]. Note that the latter can be considered as an upgraded version of the former thanks to the supplementation of functional modules like scale-aware spatial pyramid pooling, encoder mask, and scale attention [45]. Clearly, while these varying-rate dilated convolution-oriented approaches can capture contextual information well for segmenting static things impressively, our technique is more robust at tackling motion instances of different scales thanks to the comprehensive utilization of middle-level features. In particular, despite facing the issue of handling very small-sized objects in high-resolution images, the proposed cross-attentional mechanism still enables our model to reasonably locate diverse representations of medium- to large-scale targets such as the moving instances. Simultaneously, with the dense combination scheme between the decoded feature maps by the Bracket-structured network, the localization of those categories is continuously refined for the optimal pixel-wise labeling as depicted in Fig. 7. Furthermore, compared to the sibling B-Net-VGG-LCM [16], our CAB-Net, with the remarkable improvements in terms of backbone network as well as attentional connection scheme, can label the objects more accurately (e.g., the representations of *sidewalk* category in the second row of Fig. 7).

4.4.4. Computational complexity

Finally, we compare the computational complexity in terms of inference speed and total number of parameters with several existing methods belonging to both *symmetrically*- and *asymmetrically-structured* network families. In each group, approaches with different ultimate objectives, i.e., focusing more on either labeling accuracy or inference rapidity, are involved in the discussion. Accordingly, high-resolution images (1024 × 2048) in Cityscapes dataset [9] are taken into account for this experiment, from which the comparison details are given in Table 6. The proposed CAB-Net is run on a Linux OS-based desktop computer equipped with Intel® Core™ i7-7700 CPU at 3.6 GHz × 8, NVIDIA GeForce GTX 1080Ti GPU, and 32 GB RAM, which yields the inference speed of 13 frames per second (fps). Meanwhile, the B-Net-VGG-LCM [16] reaches 20 fps because channel sizes of the feature maps utilized from the backbone VGG-16 are considerably shallower.

Regarding the *symmetrically-structured* topology, both SegNet [1] and SwiftnetRN-18 [29] have faster segmentation speeds of 4 and 26 fps than our model due to the employment of much lower-capacity CNNs, i.e., VGG-16 [33] and

Table 4
Comparison of per-class IoU and mIoU (%) on Cityscapes [9] test set. **Boldface** numbers indicate the best performance at each class.

Approach	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU (%)
SegNet [1]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	56.1
FSSNet [43]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	58.8
FCN [28]	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
DeepLab-CRF [7]	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
RefineNet [26]	98.2	83.3	91.3	47.8	50.4	56.1	66.9	71.3	92.3	70.3	94.8	80.9	63.3	94.5	64.6	76.1	64.3	62.2	70.0	73.6
BiSeNet [39]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	74.7
SwiftNetRN-18 [29]	98.3	83.9	92.2	46.3	52.8	63.2	70.6	75.8	93.1	70.3	95.4	84.0	64.5	95.3	63.9	78.0	71.9	61.6	73.6	75.5
B-Net-VGG-LCM [16]	98.4	84.8	92.4	55.1	55.5	62.1	71.7	76.3	93.3	71.4	95.0	85.1	67.9	95.6	60.5	72.0	62.4	67.3	74.9	75.9
DUC–HDC [35]	98.5	85.5	92.8	**58.6**	55.5	65.0	73.5	77.9	93.3	72.0	95.2	84.8	68.5	95.4	70.9	78.8	68.7	65.9	73.8	77.6
PSPNet [44]	**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
SSPP-ES [45]	98.5	85.9	**93.1**	51.8	**59.6**	**67.8**	74.9	78.3	**93.5**	72.2	**95.5**	**86.8**	**72.1**	95.6	68.9	81.3	74.8	70.2	76.6	**78.8**
CAB-Net (ss)	98.3	83.4	92.3	52.2	54.8	62.0	70.8	75.4	93.3	70.9	95.2	85.1	68.4	95.4	65.4	80.3	**80.7**	67.7	73.7	77.1
CAB-Net	98.5	85.4	92.8	55.6	59.1	63.3	70.9	75.6	93.4	71.1	95.2	86.4	71.3	**95.9**	**72.3**	**82.2**	72.3	**70.4**	76.5	78.3

ss: single-scale testing strategy; –: no data recorded in the original work.

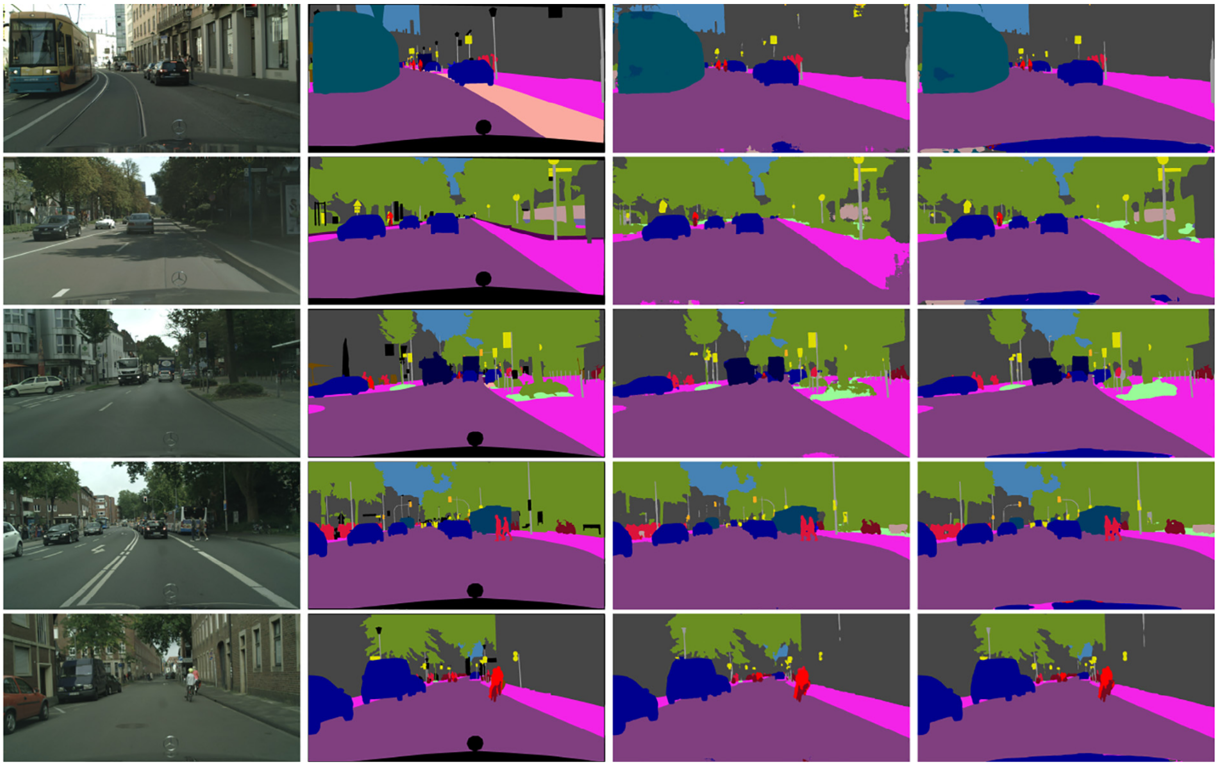


Fig. 7. Several qualitative results on Cityscapes [9] validation set. Left to right: original images, ground-truth labels, B-Net-VGG-LCM [16], and our CAB-Net.

Table 5
Comparison of per-class IoU and mIoU (%) on CamVid [4] test set. **Boldface numbers** indicate the best performance at each class.

Approach	Building	Tree	Sky	Car	Sign-symbol	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	mIoU (%)
SegNet (3.5K dataset) [1]	–	–	–	–	–	–	–	–	–	–	–	60.1
DeepLab-LFOV [5]	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6
Dilation8 [41]	82.6	76.2	89.9	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3
Dilation+FSO-DF [22]	84.0	77.2	91.3	85.6	49.9	92.5	59.1	37.6	16.9	76.0	57.2	66.1
B-Net-VGG-LCM [16]	81.4	75.3	92.8	82.5	42.8	89.2	60.8	47.8	**36.3**	66.4	54.8	66.4
G-FRNet [20]	82.5	76.8	92.1	81.8	43.0	94.5	54.6	47.1	33.4	82.3	59.4	68.0
BiSeNet [39]	83.0	75.8	92.0	83.7	46.5	94.6	58.8	53.6	31.9	81.4	54.0	68.7
DDSC [3] (ss)	–	–	–	–	–	–	–	–	–	–	–	70.9
LDN121 16→2 [21]	–	–	–	–	–	–	–	–	–	–	–	75.8
CAB-Net (ss)	88.7	87.2	94.9	91.0	60.5	**94.9**	57.4	60.4	26.8	**85.4**	55.4	73.0
CAB-Net	**91.1**	**88.9**	**95.7**	**93.0**	**64.8**	94.7	**66.5**	**70.5**	29.8	85.3	**60.3**	**76.4**

ss: single-scale testing strategy; –: no data recorded in the original work.

Table 6
Comparison of mIoU, inference speed, and number of model parameters for input image with resolution of 1024×2048 in Cityscapes [9] dataset. **Boldface** numbers indicate the best performance at each criterion.

Network structure	Approach	NVIDIA GPU	mIoU (%)	Inference speed (fps)	No. parameters
Symmetric	SegNet [1]	Titan X	56.1	17	29.46M
	SwiftnetRN-18 [29]	GTX 1080Ti	75.5	39	11.80M
Asymmetric	PSPNet [44]	GTX 1080Ti	**78.4**	7	65.60M
	BiSeNet [39]	Titan Xp	74.7	**65.5**	49.00M
Bracket	B-Net-VGG-LCM [16]	GTX 1080Ti	75.9	20	25.92M
	CAB-Net	GTX 1080Ti	78.3	13	81.47M

ResNet-18 [13], respectively. In concrete, for such kind of symmetric encoder-decoder, the process of inferring pixel-wise labeled map mainly relies on the inherent structure of backbone CNN. Therefore, applying shallower network that extracts features having smaller depth size to be involved in the decoding stage requires fewer parameters and operations throughout the whole architecture. However, there is a huge trade-off with the mIoU-based performance, wherein our CAB-Net greatly outperforms the SegNet [1] and SwiftnetRN-18 [29] by 22.2% and 2.8%, respectively. On the other hand, it is noteworthy that the B-Net-VGG-LCM [16], another representative of Bracket-style structure, attains higher mIoU (of 19.8%) and processing rate (of 3 fps) while having same backbone CNN but fewer parameters (of 12%) in comparison with those of the SegNet [1]. The major reason is that its decoder is the reverse replication of the original VGG-16 [33], which is obviously more expensive than the connections between several selective feature maps only in the Bracket-shaped structure or the SwiftnetRN-18 [29].

Compared to PSPNet [44] and BiSeNet [39] in *asymmetrically-structured* group, the proposed CAB-Net contains more number of parameters but still accomplishes noticeable results in the remaining criteria. Particularly, the inference speed introduced by our method (13 fps) is nearly twice as many as that of the PSPNet [44] (7 fps) while the gap of attained mIoU is trivial (0.1%). We argue that the primary cause is the manifold utilization of the deepest feature maps in ResNet-101 [13] for various pooling rates followed by conventional convolutional layers in that approach. Such strategy heavily elaborates the volume of operations (comprising multiply, add, max-value calculations, etc), which subsequently reduces the segmentation speed. Meanwhile, the involvement of efficient attentional and *Sep. Conv.* layers to lower-depth features round-by-round in our technique conducts cheaper operation burden despite carrying more learnable parameters. On the contrary, since the BiSeNet [39] targets at processing rapidity more favorably, it is built upon the lightweight backbone ResNet-18 [13] with an attached dual network stream for amalgamating global context and local details in a cost-efficient way. Hence, the inference speed is impressive with approximately 65 fps but compromising poorer mIoU with a gap of 3.6% compared to that of the proposed CAB-Net. In a nutshell, it can be realized that the mIoU, inference speed, and model complexity in terms of parameters' amount are strongly correlated criteria, to which the preference certainly depends on predefined purposes of each deep learning model.

5. Conclusion

In this paper, we pointed out how the proposed Bracket-structured network jointly works with the CAF modules to balancedly combine fine-grained with semantically rich features. Such Bracket-style decoding procedure along with the cross-attentional manner can finalize the prediction map more accurately as middle-level feature maps are exhaustively exploited. They are employed for not only continuously refining semantically rich features of higher-level layers, but also incessantly embedding reasonable contexts to finely patterned features of lower-level layers during the tournament of per-pixel labeled map inference. Accordingly, with promising performance in the experiments, the proposed model is expected to effectively represent semantic categories from raw images for further information-based processes in realistic vision-oriented applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Cam-Hao Hua: Conceptualization, Methodology, Software, Investigation, Writing - original draft. **Thien Huynh-The:** Validation, Formal analysis, Writing - review & editing. **Sung-Ho Bae:** Supervision, Resources, Visualization, Writing - review & editing. **Sungyoung Lee:** Supervision, Resources, Funding acquisition.

Acknowledgements

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-0-01629) supervised by the IITP (Institute for Information & communications Technology Promotion). This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00655). This research was supported by NRF-2016K1A3A7A03951968 and NRF-2019R1A2C2090504.

References

[1] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (12) (2017) 2481–2495.

[2] S. Banerjee, S. Mitra, B.U. Shankar, Automated 3D segmentation of brain tumor using visual saliency, Information Sciences 424 (2018) 337–353.

[3] P. Bilinski, V. Prisacariu, Dense Decoder Shortcut Connections for Single-Pass Semantic Segmentation, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 6596–6605..

[4] G. J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and Recognition Using Structure from Motion Point Clouds, in: Computer Vision – ECCV 2008, Springer, Berlin Heidelberg, 2008, pp. 44–57..

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected CRFs, in: 3rd International Conference on Learning Representations ICLR, 2015.
[6] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T. Chua, SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6298–6306..
[7] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 834–848.
[8] F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807..
[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes Dataset for Semantic Urban Scene Understanding, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3213–3223..
[10] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012..
[11] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 3146–3154..
[12] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, J. Malik, Semantic contours from inverse detectors, in: 2011 International Conference on Computer Vision, 2011, pp. 991–998.
[13] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778..
[14] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141..
[15] C.-H. Hua, T. Huynh-The, K. Kim, S.-Y. Yu, T. Le-Tien, G. Park, J. Bang, W. Khan, S.-H. Bae, S. Lee, Bimodal learning via trilogy of skip-connection deep networks for diabetic retinopathy risk progression identification, International Journal of Medical Informatics 132 (2019), 103926.
[16] C.-H. Hua, T. Huynh-The, S. Lee, Convolutional Networks with Bracket-Style Decoder for Semantic Scene Segmentation, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp. 2980–2985..
[17] T. Huynh-The, C. Hua, D. Kim, Encoding Pose Features to Images With Data Augmentation for 3-D Action Recognition, IEEE Transactions on Industrial Informatics 16 (5) (2020) 3100–3111, ISSN 1941-0050..
[18] T. Huynh-The, C.-H. Hua, T.-T. Ngo, D.-S. Kim, Image representation of pose-transition feature for 3D skeleton-based action recognition, Information Sciences 513 (2020) 112–126, ISSN 0020-0255.
[19] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, vol. 37, 2015, pp. 448–456..
[20] M. A. Islam, M. Rochan, N. D. B. Bruce, Y. Wang, Gated feedback refinement network for dense image labeling, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4877–4885..
[21] I. Kreso, J. Krapac, S. Segvic, Efficient Ladder-style DenseNets for semantic segmentation of large images, CoRR abs/1905.05661, http://arxiv.org/abs/1905.05661..
[22] A. Kundu, V. Vineet, V. Koltun, Feature space optimization for semantic video segmentation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3168–3175..
[23] T.H.N. Le, K.G. Quach, K. Luu, C.N. Duong, M. Savvides, Reformulating level sets as deep recurrent neural network approach to semantic segmentation, IEEE Transactions on Image Processing 27 (5) (2018) 2393–2407.
[24] H. Li, P. Xiong, J. An, L. Wang, Pyramid attention network for semantic segmentation, in: British Machine Vision Conference 2018, BMVC, 285, 2018..
[25] X. Li, H. Zhao, L. Han, Y. Tong, K. Yang, GFF: Gated fully fusion for semantic segmentation, CoRR abs/1904.01803, http://arxiv.org/abs/1904.01803..
[26] G. Lin, F. Liu, A. Milan, C. Shen, I. Reid, RefineNet: multi-path refinement networks for dense prediction, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 1, 1.
[27] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944..
[28] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.
[29] M. Orsic, I. Kreso, P. Bevandic, S. Segvic, In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 12607–12616..
[30] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in: NIPS Autodiff Workshop, 2017..
[31] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, 2015, pp. 234–241..
[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252.
[33] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: 3rd International Conference on Learning Representations ICLR, 2015.
[34] Z. Tian, T. He, C. Shen, Y. Yan, Decoders matter for semantic segmentation: data-dependent decoding enables flexible feature aggregation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 3126–3135..
[35] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, G. Cottrell, Understanding convolution for semantic segmentation, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 1451–1460..
[36] T. Wu, S. Tang, R. Zhang, J. Cao, J. Li, Tree-structured Kronecker convolutional network for semantic segmentation, in: 2019 IEEE International Conference on Multimedia and Expo (ICME), ISSN 1945-7871, 2019, pp. 940–945..
[37] Z. Wu, C. Shen, A. van den Hengel, Wider or deeper: revisiting the ResNet model for visual recognition, Pattern Recognition 90 (2019) 119–133.
[38] M. Yang, K. Yu, C. Zhang, Z. Li, K. Yang, DenseASPP for semantic segmentation in street scenes, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3684–3692..
[39] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, N. Sang, BiSeNet: Bilateral segmentation network for real-time semantic segmentation, in: Computer Vision – ECCV 2018, ISBN 978-3-030-01261-8, 2018, pp. 334–349..
[40] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, N. Sang, Learning a discriminative feature network for semantic segmentation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1857–1866..
[41] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: 4th International Conference on Learning Representations ICLR, 2016.
[42] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, A. Agrawal, Context encoding for semantic segmentation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7151–7160..
[43] X. Zhang, Z. Chen, Q.M.J. Wu, L. Cai, D. Lu, X. Li, Fast semantic segmentation for scene perception, IEEE Transactions on Industrial Informatics 15 (2) (2019) 1183–1192.
[44] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, 6230–6239..
[45] F. Zhou, Y. Hu, X. Shen, Scale-aware spatial pyramid pooling with both encoder-mask and scale-attention for semantic segmentation, Neurocomputing 383 (2020) 174–182, ISSN 0925–2312..