Semantic Segmentation with Context Encoding and Multi-Path Decoding

Henghui Ding, Xudong Jiang, Senior Member, IEEE,
Bing Shuai, Ai Qun Liu, Gang Wang, Senior Member, IEEE

Abstract—Semantic image segmentation aims to classify every pixel of a scene image to one of many classes. It implicitly involves object recognition, localization, and boundary delineation. In this paper, we propose a segmentation network called CGBNet to enhance the segmentation performance by context encoding and multi-path decoding. We first propose a context encoding module that generates context-contrasted local feature to make use of the informative context and the discriminative local information. This context encoding module greatly improves the segmentation performance, especially for inconspicuous objects. Furthermore, we propose a scale-selection scheme to selectively fuse the segmentation results from different-scales of features at every spatial position. It adaptively selects appropriate score maps from rich scales of features. To improve the segmentation performance results at boundary, we further propose a boundary delineation module that encourages the location-specific very-low-level features near the boundaries to take part in the final prediction and suppresses them far from the boundaries. The proposed segmentation network achieves very competitive performance in terms of all three different evaluation metrics consistently on the six popular scene segmentation datasets, Pascal Context, SUN-RGBD, Sift Flow, COCO Stuff, ADE20K, and Cityscapes.

Index Terms—Semantic segmentation, context encoding, gated sum, boundary delineation refinement, deep learning, CGBNet, convolutional neural networks.

I. INTRODUCTION

SEMANTIC segmentation aims to classify every pixel of a given image to one of semantic classes including not only objects (e.g., person, car, cat) but also stuff (e.g., road, grass, sky). It is a dense prediction task whose output has the same resolution with input. Semantic segmentation has been an essential component of computer vision and is in intense demand for practical applications, such as automation devices, virtual reality, self-driving vehicles and etc. It implicitly involves object recognition, object localization and boundary delineation, which requires multi-scale and multi-level visual recognition. A robust segmentation network needs to perform well at all of these implied tasks.

Recently, the great success of Deep Convolutional Neural Networks (DCNN) has largely improved the performance of computer vision tasks [25], such as image classification [66], [68], [28], [30], image deblurring [50] and object detection [56], [57], [21], [22], [40]. However, there are still some limitations when directly applying DCNN to the dense prediction tasks like semantic segmentation [62], [38], [9], [17], [72], [26], [82]. DCNN learns very abstract feature representation of the whole image [80] and hence extracts information of dominant/salient objects of image. This is highly desirable for image classification and object detection. For semantic segmentation, however, spatial information is essential and pixel-level discriminative features are desired. Although most state-of-the-art semantic segmentation frameworks are based on the image classification networks pre-trained on Imagenet [60], it remains an open question of how to better adopt DCNN on semantic segmentation. Herein, we mainly consider three issues when applying DCNN to dense prediction tasks: the various forms of objects/stuff (e.g., salient or inconspicuous), the existence of multi-scale objects and the loss of the accurate spatial boundary information. To address these issues, we propose a new segmentation model with context encoding, multi-path decoding and boundary delineation to enhance the segmentation performance from different levels.

First, semantic segmentation aims at labeling every pixel, which is different from image classification and object detection that target at image-level or region-level labeling. It requires dense prediction with high resolution. Moreover, not only the dominant salient objects but also the stuff and inconspicuous objects should be well parsed. DCNN pretrained on Imagenet [60] prefers image-level abstract features that is not equally discriminative at every spatial position but dominated by salient objects. Therefore, when directly applying DCNN on semantic segmentation, inconspicuous objects are easy to be overwhelmed by salient objects and their information will be weakened or even disregarded. This is undesirable for the scene segmentation. To address this issue, locally discriminative features together with their contexts are essential for scene segmentation. A lot of work devotes to get informative context, e.g. [9], [64], [49], [79]. However, contexts often have smooth representation and are dominated by features of salient objects, which is harmful for labeling inconspicuous objects. Better features for scene segmentation are discriminative context-aware local features, in which each pixel remains its discrimination and also gets information from surroundings.

For this purpose, we propose a context-contrasted local feature, which benefits from both context and local information. The proposed context-contrasted local feature not only exploits the informative context but also spotlights the discriminative local information in contrast to the context. Further, we use a
context-contrasted local (CCL) model to aggregate multi-scale and multi-level context-contrasted local features.

Second, object recognition extracts features for recognizing the whole image, but for segmentation, it is irrational to classify all individual pixels based on a single scale of features. Due to the huge scale variation of objects (e.g., the multi-scale cows in Fig. 1), adaptive scales of features for parsing different pixels are desired, for example, smaller scales of receptive filed for pixels belonging to smaller objects. Previous works address this issue with several different ways. One way is to resize the input image to multiple resolutions and feed them to different (or a shared) networks, then fuse their features, such as approaches in [42], [18], [11], [55]. The aggregation ability of this strategy is limited due to the limited scales of input images used in practice to avoid heavy computational load of this scheme. Another way makes use of multiple levels of features from the middle layers of network, such as approaches in [62], [27], [59], [20]. The intention of this strategy is to exploit multi-scale features. We follow the way of FCN [62] to adopt skip layers to utilize multi-scale features, which is effective and economic. However, in previous works, such as [62], [27], [51], [64], [10], [54], the score maps of skip layers are integrated via a simple sum fusion that ignores different importance of different scales. To address this problem and find an optimal integration choice, we propose a network that controls the information flow of different scale features. It generates control signals to perform a gated sum of the score maps to aggregate multi-scale features selectively. As a selection mechanism is embedded in the multi-scale fusion, more skip layers can join the aggregation to provide richer information for selection. This enhances the aggregation ability of multi-scale features.

Third, detailed spatial information is essential for scene segmentation, especially for the implicit low-level tasks, such as boundary delineation. However, due to the pooling operation or convolution strides of DCNN, lots of spatial information is lost during the encode process. To address this issue, one direction is to retain the spatial information using dilated DCNN, such as DeepLab [9] and PSPNet [82] that remove some of the pooling layers or convolution strides. The dilated DCNN, though performs well, is computationally expensive due to retaining numerous relatively high-resolution feature maps. Another direction is to recover the spatial information during the decode process, such as FCN [62] and RefineNet [41] that employ low-level features. These features contain sufficient low-level visual information that provides textural information and helps make the final prediction finer. However, incorporating very-low-level features (e.g., block2 in ResNet [28]) brings some noise, which adversely affects the high-level tasks such as the segmentation of coherent semantic regions of large-scale objects. In other word, chaotic very-low-level information is not always helpful for every pixels. It should be exploited only for pixels near the textural boundary for accurate boundary delineation while be stifled in the homogenous region. To find a way to appropriately exploit the very-low-level features, we propose a refinement method that gets the textural boundary information from an initial prediction of segmentation network. Then, to refine the initial prediction, the boundary information is fed to the network to incorporate the very-low-level features in the aggregation for pixels near the boundary while suppressing them for other pixels. This method brings very little computation but improves the boundary delineation.

In summary, this paper makes the following contributions: 1) We propose a novel context-contrasted local feature that is tailored for scene segmentation and propose a context-contrasted local (CCL) model to aggregate multi-scale context-contrasted local features. 2) We propose a gated sum scheme to selectively aggregate appropriate-scale features at each spatial location, which is an efficient and effective way to address the issue of the existence of multi-scale objects. 3) We propose a boundary delineation model to refine the prediction of boundary, which encourages the location-informative very-low-level features near the boundaries and suppressed them far from the boundaries. 4) We achieve very competitive performance consistently on the six popular semantic segmentation datasets.

II. RELATED WORK

A. Contextual Modeling

One direction is to apply new layers on the top of the pre-trained DCNN to enhance high-level contextual aggregation. For example, Chen et al. [9] introduced an atrous spatial pyramid pooling (ASPP) to capture useful context information at multiple scales. Visin et al.[70], Ding et al. [15] and Byeon et al.[6] adopted recurrent neural networks to capture long-range context. Zhao et al.[82] employed multiple parallel pooling layers to exploit global information from different regions. Liu et al. [49] proposed to model the mean field algorithm with local convolution layers and incorporate it in deep parsing network (DPN). Yu et al. [79] attached multiple dilated convolution layers after class likelihood maps to exercise multi-scale context aggregation. Sharma et al. [61] proposed a recursive context propagation network to disseminate global information to different local regions. Another way is to use Conditional Random Fields (CRF) [36] to model the context of score maps [10], [9], [83], [42], [49]. For example, Chen et al. [9] adopted CRF to post-process the unary predictions and generate
smoother prediction maps. Zheng et al. [83] proposed CRF-RNN to jointly train CRF with their segmentation networks. Zhang et al. [81] captured global contextual information and selectively highlighted class-dependent feature maps.

B. Multi-scale Aggregation

Due to the huge scale variation of objects, it is difficult to achieve robust segmentation only based on the single scale of features. Multi-scale aggregation is a crucial way to deliver robust parsing maps. There are several methods to achieve multi-scale aggregation. Farabet et al. [18] and Lin et al. [42] adopted multi-resolution input (image pyramid) approach and fuse the corresponding features from multiple resolution. Liu et al. [47] generated multi-scale patches and aggregated the results. Pinheiro et al. [55] inputted multi-size images at different layers of a recurrent convolutional neural network. However, the above approaches are computational expensive and consume large GPU memory. Thus, their aggregation ability for multi-scale features is limited due to the limited scales used in practice to avoid heavy computational load of this scheme. The seminal work FCN [62] introduced the skip layers to locally classify multi-scale feature maps and aggregate their predictions via sum fusion. This is an effective as well as efficient method to integrate different scale features and our work follows this way. Nonetheless, in previous works [62], [27], [51], [64], [10], [54], the score maps of skip layers are fused via a simple sum and hence the different importance of different scales are ignored. To address this issue, we propose a network that facilitates a gated sum to selectively aggregate different scale features. With gated sum fusion, the network can exploit more skip layers from richer scale features in DCNN and customize a suitable integration of different scale features. To the best of our knowledge, our gated sum is the first work to selectively aggregate appropriate scale features in a single network.

C. Boundary Delineation

Boundary delineation is an implicit task in scene segmentation. Due to the diverse shape of objects and complex layout, boundaries are capricious and difficult to be predicted accurately. Conditional Random Fields (CRF) [36] is often adopted as a post-processing method to get a better boundary prediction, such as Deeplab[9]. Apart from CRF. Barron et al. [4] proposed bilateral solver to prompt edge-aware smoothness, and Jampani et al. [33] integrated the bilateral solver in DCNN to jointly train them form data. Peng et al. [54] integrated a residual convolutional architecture in their network to increase the accuracy of boundary region.

In this paper, we propose a boundary delineation refinement (BDR) model, which employs a location-selection scheme for filtration of very-low-level features to refine the spatial boundary. BDR encourages pixels near the boundaries to be assigned more very-low-level information and suppresses it for pixels far from the boundaries. It brings very little computation but enhances the boundary delineation. Furthermore, BDR helps enhance the coalescence of multi-level tasks in a single network.

III. THE PROPOSED SEGMENTATION NETWORK

A. Overall Framework

In this paper, we propose a segmentation network called CGBNet to enhance the segmentation performance by context encoding, multi-path decoding and boundary delineation.

The overall framework of the proposed segmentation network, named as CGBNet (CCL, Gated sum and BDR), is shown in Fig. 2. It contains context-contrasted feature extraction for context encoding, gated multi-scale aggregation and boundary delineation refinement for multi-path decoding. The baseline is a FCN-like architecture with ResNet-101 as backbone network. The proposed context-contrasted local (CCL) model in Fig. 2 generates multi-level and multi-scale context-aware local features. The proposed gated sum denoted by $g^+$ in Fig. 2 selectively aggregates rich scale features extracted in DCNN and CCL. Furthermore, to generate finer boundary delineation in the final prediction, a boundary delineation refinement model (BDR) is proposed to filter the very-low-level features. The proposed CCL, Gated Sum and BDR are presented in details in the following sections.

B. Context-Contrasted Local Feature

Context aims at collecting surrounding information and enlarging the effective receptive field, which can greatly improve the performance of semantic segmentation. DCNN trained for object recognition has already generated relatively high-level context features [66], [28], but its high-level features are learnt for overall abstract representation of the whole image, which focus on the dominant parts and cannot ensure useful context for inconspicuous objects and stuff. Also, they may not be discriminative at some spatial positions as they are trained to discriminate the whole image collectively. Therefore, the context features for object recognition are not directly favorable for scene segmentation that aims at classifying every pixel [16]. Lots of previous works devote to obtain context for semantic segmentation. For example, [9] and [79] use dilated convolutions to aggregate coarse context while [64] and [6] adopt a recurrent neural network to capture long-range dense context. However, either dense context or coarse context is easy to be dominated by the features of salient objects, resulting in weak signals of inconspicuous objects. Comparing to object segmentation that focuses on salient objects, there are richer categories and complex conjunctions between categories in scene segmentation [64]. Due to the complexity of objects and stuff in scene segmentation, indiscriminately collecting context information may bring harmful interference, especially under clutter surroundings. For example, in Fig. 3, compared with the two persons, the cars behind them are inconspicuous objects. The detailed local features collect information around pixel $A$ and are discriminative to other pixels, but they are not aware of global information such as road and building, thus could not obtain robust high-level information for pixel $A$. However, aggregating context for pixel $A$ brings features of salient objects like the persons and hence will be dominated by the features of the persons. Some information of cars may be ignored in the final prediction, resulting wrong labeling for pixels at that location. Also, contexts at different positions are
Herein, it is significant to design tailored context features for scene segmentation. To this end, a context-contrasted local feature is proposed in this paper to perform high-level feature modeling, which injects blur context to local feature to make discriminative context-aware local feature. Specifically, we propose to generate local information and context separately. To this end, a context-contrasted local feature is proposed in this paper to perform high-level feature modeling, which injects blur context to local feature to make discriminative context-aware local feature. Specifically, we propose to generate local information and context separately. To this end, a context-contrasted local feature is proposed in this paper to perform high-level feature modeling, which injects blur context to local feature to make discriminative context-aware local feature.

Comparison with Existing Context Models. DAG-RNN [64] performs contextual modeling by propagating local information in feature maps to encode long-range context. Different from DAG-RNN, CCL exploits multi-scale features for segmentation, and the context-aware local features of CCL are different from those in DAG-RNN. ASPP [9] aggregates multi-scale contexts via combining score maps generated by different context aggregation branches, each of which uses dilated Conv kernels with different stride rates to generate different scale contexts in parallel. Compared with this type of context model, CCL first contextualizes contrasted features at every block to obtain context-aware local features, which combines two different scales in the feature level and takes advantage of both context and local information, then further aggregates multi-scale context-contrasted local features in score level. Moreover, the score maps of CCL are fused via gated sum instead of the simple sum. CRF [36] is ordinarily applied to score maps and boosts consistency of low-level information, while CCL aims at discriminative high-level features. In fact, CRF can also be used as a post-processing step to promote context. The function of context-local forces the networks generating tailored features for scene segmentation. Context shows solicitude for each pixel but brings some unnecessary information while local focuses on vicinal information but ignores other parts, resulting in losing some essential information. Different from the above two, context-local pays more attention to local while collecting coarse context to aggregate pivotal information, thus could get robust high-level features.

It is a mechanism that imitates the human visual behavior. When our human beings look at an object, we always collect the context for that object in a way that our eyes focus on that object in contrast to the blurred surroundings [19]. In other word, we concentrate on that object while we are aware of its surroundings. The architecture of CCL is shown in Fig. 4. It consists of several chained context-local blocks to make multi-level context-contrasted local features. Gated sum (presented in the next section) is adopted in CCL to selectively aggregate different levels of context-contrasted local features.

![Fig. 2. The overall framework of the proposed CGBNet. The proposed context-contrasted local (CCL) model generates multi-level and multi-scale context-aware local features. Gated sum selectively aggregates rich scale features in DCNN and CCL. BDR encourages pixels near the boundaries to be assigned more very-low-level textural information.](image-url)

![Fig. 3. (Best viewed in color) Visualization of different feature information. The local information of pixel A could not aggregate useful contexts, such as road and other cars. However, its contexts will be dominated by the features of the men in the both schemes of dense context and coarse context. The context-local scheme injects blur context to local feature of pixel A to make discriminative context-aware local feature.](image-url)
performance of our segmentation network.

### C. Gated Multi-scale Aggregation

Due to the existence of objects at multiple scales, it is difficult to directly apply DCNN to scene segmentation to obtain appropriate information for all pixels [16]. In this section, we discuss how to extract different scale of features from DCNN.

An efficient and effective way is to add skip layers from the middle layers of DCNN. Based on the encoder-decoder architecture of FCN [62], skip layers being as classifiers are used to exploit multi-scale features in DCNN to generate corresponding segmentation score maps. However, in previous works such as [62], [27], [51], [64], [10], the classification score maps of skip layers are non-selectively fused by sum, which does not take into account the individual differences of these score maps for different pixels. Some pixels prefer scores from features with larger receptive filed, e.g., pixels belonging larger objects while others desire features with smaller receptive filed, e.g., pixels belonging to smaller objects. Moreover, some pixels of complex structures may need to aggregate multi-scale features for better information collection. If these score maps are aggregated indiscriminately, the inapposite or incorrect scores may harm the final prediction. It is better to provide scale selections for every pixel.

To this end, we propose an selection scheme called gated sum to adaptively select the appropriate receptive field for each pixel based on its scale or representation support. This scheme contains inherent position-wise gates to select outputs from skip layers and control the information flow of DCNN. As skip layers are aimed at capturing multi-scale features, a bypass and simple solution is to pick different skip layers for different pixels in FCN framework. With the gated sum fusion, the network customizes a suitable aggregation choice of score maps according to the information of images, corresponding to choosing which scales of features are better and more desirable for each pixel. Moreover, with gated sum fusion, we can add more skip layers to extract richer scale information without posing the problem of inapposite results.

In order to obtain the information to control the gates, such as scale and contextual support, info-skip layers consisting of Conv+Sigmoid are introduced to extract the information from corresponding feature maps and generate information maps with size of $H \times W$, where $H \times W$ is the spatial size of feature maps. Since these information maps, $I_{n,p}$, and score maps, $S_{n,p}$, of skip layers are generated from a same DCNN or CCL, the sequence relationship, e.g. from low level to high level, among feature maps of DCNN and CCL should also be considered. Recurrent Neural Networks (RNN) [23], [46], [24] are effective and efficient to learn such sequence relationship, thus all of the information maps are fed to RNN in sequence to learn the relationship of these information maps. Based on RNN, these information maps can be aware of neighbourhood maps and acquire the sequence relationship among all of the information maps. The proposed scheme of a general gated sum for feature maps of the same size is shown in Fig. 5.

In details, we hypothesize that the information maps of higher layers have already grasped the information of lower layers due to the effect of DCNN, thus the RNN begins with information map of the last layer of our segmentation network. Suppose a gated sum has $N$ score maps in a block that have the same spatial size $H \times W$, $S_{i,n}$, generated by $N$ skip layers from different scale features $F_{i,p}$. We have $S_{n,p} = \mathcal{F}_s(F_{n,p}, \Theta_s)$, where $p$ is the spatial position, $n \in 1, 2, ..., N$, $c \in 1, 2, ..., C$ and $C$ is the number of class labels. $\mathcal{F}_{n,c}$ is the classification function of $n$th skip layer, $\Theta_s$ is its parameters and $F_{n,c}$ denotes the input feature maps with the dimensionality of $H \times W \times \text{# channels}$. For each skip layer, we first generate an information map $I_{n,p}$ of size $H \times W \times I$ from the corresponding feature maps:

$$I_{n,p} = \mathcal{F}_i(F_{n,c}, \Theta_i)$$

where $\mathcal{F}_i$ is the function of $n$th info-skip layer.
Conv+Sigmoid and $\Theta^p_n$ is its parameters. Then these information maps $I^p_n$ are inputted to a RNN in sequence to learn their relationships:

$$h^p_n = \tanh \left( W^n \left( h^{n-1}_p \right) \right)$$

(3)

where $h^p_n$ is the $n$th output of RNN. To make our network efficient, all positions are processed parallely and $W^n$ is shared for all spatial positions. To ensure every information map be aware of global information, the outputs of RNN are concatenated, $H_p = (h^1_p, ..., h^N_p)^T$, and refined with global information:

$$\overline{H}_p = \mathcal{F}_g (H_p, \Theta_g) + H_p$$

(4)

where $\mathcal{F}_g$ is a $1 \times 1$ CONV and $\Theta_g$ is its parameters. Next, $\overline{H}_p$ is splitted, $\overline{H}_p = (\overline{h}^1_p, ..., \overline{h}^N_p)^T$, and used to generate the gates $G^p_n$ for gated sum:

$$G^p_n = N \frac{e^\overline{h}^p_n}{\sum_{i=1}^N e^\overline{h}^i_p}$$

(5)

The sum of $G^p_n$ for each position $p$ is normalized to $N$. Finally, $N$ score maps are selectively fused via gated sum:

$$\overline{S}_p = \sum_{n=1}^N G^p_n \odot \overline{S}^n_p$$

(6)

where $\odot$ denotes element-wise multiplication and $\overline{S}^p_n$ is the output of the gated sum. The embedding of this gated sum (Fig. 5) into the encoder-decoder architecture (Fig. 2) will be discussed in section IV-C.

The gates of gated sum control the information flow of skip layers, i.e. how much the $\overline{S}^n_p$ can pass the gates depends on the value of $G^p_n$. A larger $G^p_n$ means a better feature $n$th skip layer for labeling of the position $p$. While at a smaller $G^p_n$ means that for position $p$, the segmentation result generated by the $n$th skip layer is not desirable and should be inhibited. More importantly, $G^p_n$ is neither fixed value nor directly learned from training data. It is generated from the testing image by the proposed network learned from the training data. Thus, $G^p_n$ is adaptive to different pixels of testing image. The values of $G^p_n$ not only depend on the training data, but also depend on the testing input images and vary according to the feature maps. Therefore, we call them “gates” to differentiate them from the simple fixed or learned “weights”. With gated sum, the network adaptively (to different testing images) selects appropriate score maps from rich scales of features.

D. Boundary Delineation Refinement

Semantic segmentation requires multi-level tasks, including low-level tasks such as boundary delineation, high-level tasks such as object recognition. A well-performed semantic segmentation architecture should be able to effectively deal with different tasks. In this section, we discuss how to recover the boundary information and refine the boundary delineation of segmentation.

Due to the pooling operations and convolution strides, detailed spatial information is lost during the encode process, which does not conform to the goal of scene segmentation. To address this issue, up-sampling and decoding the output of DCNN through learnable deconvolutional filters [53] is feasible and efficient. But the deconvolution alone cannot recover the textual spatial information that is already lost due to the pooling operations and convolution strides. To recover the spatial knowledge, it is necessary to exploit the very-low-level feature information that contains sufficient textual visual information (e.g. corners, edges, etc) and supplement these information for the accurate boundary prediction. Compared with the gated sum that mainly selects different scale features for the task of segmenting different scales of objects, boundary delineation requires higher resolution (lower-level) features to provide finer informative textual boundary knowledge. For example, in Fig. 6, to delineate the boundary layout of object 1, the parsing of pixel B is essential. Although the gated sum can select different scales of feature for pixel A and pixel C, as the processing of upsample, the spatial size is enlarged and the very-low-level information (e.g. corners, edges, etc) becomes more and more important for getting finer prediction around pixel B. Therefore, providing meticulous edge information for parsing pixel B is necessary. However, the very-low-level features are not suitable for high-level tasks such as object recognition as incorporating very-low-level features brings some noise information for the high-level tasks, especially for the positions relatively far from boundary.

To get meticulous boundary information, it is necessary to make use of the very-low-level features that embody textual visual information and have higher signal-to-noise ratio (SNR) near the boundary. On the other hand, the very-low-level features contain noise information and have lower SNR far away from the boundary, which are not robust for object recognition. Therefore, we propose a boundary delineation refinement (BDR) model that suppresses the very-low-level features far away from the boundary where their SNR is lower, and engages them near the boundary where the SNR is higher.

In detail, we propose a refinement method that first gets the boundary map $B^0_p$ from the initial prediction of segmentation network, $B^0_p = 1$ for pixels at the boundary and $B^0_p = 0$ otherwise. Since the boundary generated from the initial prediction is not perfectly aligned with the real edges, directly using this hard mask may repeat mistakes of the initial prediction. To address this issue, isotropic low-pass Gaussian filters with different standard deviation $\sigma^m$ are adopted to diffuse the
initial hard boundary map, which yields soft boundary masks $B^m_p$, $m = 1, 2, 3$, as shown in Fig. 7 for $m = 3$, where wider boundary with different soft values is produced. The 1D profile $b^m_x$ of the soft boundary masks $B^m_p$ for a single line initial boundary can be expressed as

$$b^m_x = \frac{1}{\sigma^m \sqrt{2\pi}} e^{-\frac{(x-x_0)^2}{2\sigma^2}}$$

where $t$ is the position of the initial textural boundary, $\sigma^m \in \{1, 3, 5\}$, $m \in \{1, 2, 3\}$, sequentially corresponding to three very-low-level features (from lower to higher) used in BDR. Fig. 8 plots these 3D profiles $b^m_x$ of the soft boundary masks $B^m_p$. It shows that all these three soft boundary masks decrease their values with the increase of the distance from the initial boundary. Moreover, as shown in Fig. 8 at $x = t, t + \Delta_1, t + \Delta_2$, nearer the initial boundary, lower-level features have higher mask values than higher-level features while farther away from the initial boundary, higher-level features have higher mask values than lower-level features. These masks are then used to select the very-low-level features that contain textural boundary information as a supplement to the up-sampled high-level score maps from the gated sum:

$$ \hat{S}^m_p = \sum_{m=1}^{M} B^m_p \circ \hat{S}^{c,m}_p + \hat{S}^i_p $$

where $B^m_p$ is the $m$th soft boundary mask, $\hat{S}^{c,m}_p$ is the score maps from $m$th very-low-level features, $\hat{S}^i_p$ is the boundary refined score maps, $\hat{S}^m_p$ is the up-sampled output of gated sum. To remove the possible isolated noise introduced by using the very-low-level features, a simple median filter is adopted in testing process to smooth the final prediction.

The proposed BDR encourages pixels near the boundaries to be assigned more very-low-level information and suppresses it far from the boundaries. It brings very little computation yet enhances the boundary delineation.

IV. EXPERIMENTS

We evaluate our segmentation framework on six public scene segmentation datasets, Pascal Context, SUN-RGBD, Sift Flow, COCO Stuff, ADE20K and Cityscapes.

A. Implementation Details

We use truncated ResNet-101 (pre-trained on ImageNet [60]) as our backbone model. In detail, pool5 and layers after it are discarded and a convolutional adaption layer that decrease the feature channels from 2048 to 512 is placed on the top of truncated ResNet-101 to reduce parameters. The number of layers in CCL can be modified according to inputs. All experiments of this paper have six. We upsample the score maps with deconvolution (transpose convolution).

Our Network is trained end-to-end with SGD with fixed momentum 0.9 and weight decay 0.0005. Following [9], we employ the “poly” learning rate, $Lr_c = Lr_i \times (1 - \frac{iter}{max\_iter})^{power}$, where the $Lr_c$ is current learning rate and $Lr_i$ is the initial learning rate. The initial learning rate is set to be $10^{-3}$ and the power is set to 0.9. The iteration number is set to 15K for Pascal Context, 13K for SUN-RGBD and 20K for COCO Stuff. Batch size is 10 during training and the statistics of batch normalization layer is updated after the final iteration. The parameters of new layers are randomly initialized with Gaussian distribution (variance $10^{-2}$) and trained with higher learning rate ($\times 3$). For batch processing, all images are resized to have maximum extent of 512 pixels and padded with zero to 512 $\times$ 512 pixels during training. We randomly flip the images horizontally to augment the training data. Batch Normalization (BN) [32] is used to accelerate training.

We evaluate our network with three performance metrics: Global Pixel Accuracy (GPA), Average Class Accuracy (ACA) and Mean Intersection-over-Union (IoU). Mathematical definitions please refer to [62].

B. Multi-scale Context-Contrasted Local Features

In section III-B we introduced context-contrasted local (CCL) model to integrate multi-level context-aware local features. To evaluate the key principle (i.e. multi-scale context-contrasted local features) of CCL, we simplify our context-local network architecture CCL to LA, and LA$^d$. LA abandons the context parts (dilated Conv) of CCL and LA$^d$ doubles the hidden dimensionality of LA. The performance of these models are listed in Table I. Their performance gap clearly demonstrates the benefit brought by the proposed CCL model.

First, compared with LA that is conventional convolutional feature, CCL aggregates specialized context-contrasted local features that not only leverages the informative context but also exploits the discriminative local information in contrast to the context. In consequence, CCL outperforms LA by a
noticeable margin, which clearly shows the significance of the context-contrasted local features for scene segmentation.

It’s crucial to introduce new parameters that fill the domain gap during the fine-tuning of segmentation networks from classification networks. However, we believe that the network architecture outweighs the magnitude of parameters for boosting the performance. To validate this, we increase the hidden dimension of LA from 512 to 1024, which is denoted by $LA^d$ in Table I. The parameter quantity of $LA^d$ is then the same as CCL, but $LA^d$ does not improve the performance of LA and even slightly make it worse. This convinces us that the noticeable performance boost is mainly contributed by the architecture of the context-contrasted local features, not from the simple increase of network parameters. And we study the effect of the number of CCL blocks, as shown in TABLE III, the CCL-6 achieve best performance and thus we adopt 6 blocks in CCL. Further tests of CCL can be seen in TABLE VI.

We compare with other context models in a controlled experiment and summarize their performance in Table I. The parameter quantity of $OM$ is then the same as CCL, but $LA^d$ does not improve the performance of LA and even slightly make it worse. This convinces us that the noticeable performance boost is mainly contributed by the architecture of the context-contrasted local features, not from the simple increase of network parameters. And we study the effect of the number of CCL blocks, as shown in TABLE III, the CCL-6 achieve best performance and thus we adopt 6 blocks in CCL. Further tests of CCL can be seen in TABLE VI.

We compare with other context models in a controlled experiment and summarize their performance in Table II. The proposed CCL noticeably outperforms others, which demonstrates the significance of CCL.

### C. Embed Gated Sum into Encoder-Decoder Architecture

Gated sum is a selection mechanism to pick appropriate scale of features. But for the encoder-decoder architecture, the spatial sizes of distinct blocks are not the same, e.g. 16 × 16 for block 5 and 64 × 64 for block 3 in Fig. 2. This causes difficulty of aggregating all the score maps in one run. The most straightforward solution is upsampling all the score maps to the same resolution before the aggregation. However, this consumes a large amount of resources. Therefore, in this work, we first propose a general scheme of gated sum shown in Fig. 5 and then embed it into the encoder-decoder architecture in Fig. 2. For the embedding, we adopt the gated sum within each block where the feature maps possess the same spatial resolution. Then the output of gated sum is upsampled to higher resolution to participate in the gated sum in block with higher resolution. Meanwhile, to pass the information maps form block to block, the last output of RNN is also upsampled to generate the gates for the upsampled score map and the inputs of the next RNN.

We visualize the learned weights of gated sum in Fig. 9. As shown in Fig. 9, for the higher layer, the gated weights are relatively larger in some larger-scale area, such as the background stuff in first two images and the bigger cow in third image. For the lower layer, the gated weights are weaker in the large-scale area but stronger in smaller-scale area, such as the smaller aeroplane and boundary of larger aeroplane in the first image.

We also present an ablation experiment of the gated sum in Table VI. As shown in Table VI, the gated sum improves the performance visibly. Comparing ResNet-101 to ResNet-50, we see that the performance gain brought by the gated sum will be higher if there are more score maps for selecting.

### TABLE I

<table>
<thead>
<tr>
<th>Method</th>
<th>GPA</th>
<th>ACA</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>73.5%</td>
<td>53.9%</td>
<td>42.5%</td>
</tr>
<tr>
<td>Baseline + LA</td>
<td>75.8%</td>
<td>57.6%</td>
<td>45.9%</td>
</tr>
<tr>
<td>Baseline + LA$^d$</td>
<td>75.7%</td>
<td>56.6%</td>
<td>45.4%</td>
</tr>
<tr>
<td>Baseline + CCL</td>
<td>76.6%</td>
<td>61.1%</td>
<td>48.3%</td>
</tr>
</tbody>
</table>

$LA$ is local aggregation generated by removing the context part of CCL. LA$^d$ doubles the hidden dimensionality of LA from 512 to 1024, thus its parameter quantity is the same as CCL.

### TABLE II

<table>
<thead>
<tr>
<th>Networks</th>
<th>CA</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>None</td>
<td>42.5%</td>
</tr>
<tr>
<td>Baseline + CRF[36]</td>
<td>CRF</td>
<td>43.2%</td>
</tr>
<tr>
<td>Baseline + DAG-RNN[64]</td>
<td>DAG-RNN</td>
<td>44.1%</td>
</tr>
<tr>
<td>Baseline + ASPP[9]</td>
<td>ASPP</td>
<td>44.9%</td>
</tr>
<tr>
<td>Baseline + CCL</td>
<td>CCL</td>
<td>48.3%</td>
</tr>
</tbody>
</table>

Segmentation networks are adapted to encode-decode architecture with rich skip layers, the stride rates (dilation factors) of the four branches in ASPP are revised to {1, 3, 4, 6} respectively. For fair comparisons, gated sum is not adopted, and they only differentiate each other in terms of context aggregation (CA).

### TABLE III

<table>
<thead>
<tr>
<th>CCL-#</th>
<th>CCL-1</th>
<th>CCL-3</th>
<th>CCL-5</th>
<th>CCL-6</th>
<th>CCL-7</th>
<th>CCL-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>44.3%</td>
<td>46.8%</td>
<td>48.1%</td>
<td>48.3%</td>
<td>48.3%</td>
<td>48.2%</td>
</tr>
</tbody>
</table>

Images Lower Layer Higher Layer

**Fig. 9. Visualize of gated weights:** here we visualize gated weights of Block 3 in Fig. 2, of which the resolution is relatively high and suitable to generate significantly visible weights map. We select the first layer and last layer in the gated sum of Block 3, noted as higher layer and lower layer respectively.
Images | Initial Prediction | Hard Mask | Soft Mask | Refined Prediction | Ground Truth
--- | --- | --- | --- | --- | ---

**Fig. 10.** (Best viewed in color) **Boundary Delineation Refinement** helps to obtain textural boundary information from low-level features but avoid bringing to much noise information.

**TABLE IV**
ABLATION STUDY OF BDR IN BOUNDARY REGIONS

<table>
<thead>
<tr>
<th>BDR</th>
<th>Boundary Region IoU</th>
<th>Overall IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>40.2%</td>
<td>51.6%</td>
</tr>
<tr>
<td>✓</td>
<td>41.6%</td>
<td>52.2%</td>
</tr>
</tbody>
</table>

**TABLE V**
EFFECT OF USING THE VERY LOWER-LEVEL FEATURES

<table>
<thead>
<tr>
<th>Methods</th>
<th>Very low-level</th>
<th>BDR</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN-8s</td>
<td>X</td>
<td>n/a</td>
<td>48.1%</td>
</tr>
<tr>
<td>FCN-4s</td>
<td>✓</td>
<td>X</td>
<td>47.8%</td>
</tr>
<tr>
<td>FCN-4s</td>
<td>✓</td>
<td>✓</td>
<td>48.5%</td>
</tr>
</tbody>
</table>

The very low-level features (block2 in ResNet) brings detailed textural boundary information but also noise that are not suitable for the high-level tasks. Note the context aggregation model of FCN is replaced by CCL.

**D. Segmentation result comparisons of Boundary Delineation Refinement**

In Table. V, we show the effect of using the very-lower-level features (e.g. block2 in ResNet, Pool2 in VGG16). As shown in Table. V, although FCN-4s utilizes more skip connections from the very lower-level features, its segmentation results are inferior than FCN-8s [62]. The very low-level features contain informative textural layout but also noise. Using them at all pixels in the multiple scale aggregation adversely affects the segmentation performance. To address this issue, we propose a boundary delineation refinement model to selectively utilize the low-level features: encourages the very low-level features near the boundaries but suppress them far from the boundaries.

We also present qualitative segmentation result comparisons of Boundary Delineation Refinement in Fig. 10. As shown in Fig. 10, the boundary delineation refinement model improves the details of boundary visibly. To verify the effect of BDR among boundaries, we also evaluate our model only in boundary regions, which locate close to boundaries (distance ≤ 10 pixels). As shown in TABLE IV, although the overall performance brought by BDR is only 0.6%, the performance gain in the boundary regions reaches 1.4%. This agree with the goal of BDR and also confirms its effectiveness.

**TABLE VI**
ABLATION EXPERIMENTS OF PROPOSED METHODS

<table>
<thead>
<tr>
<th>Baseline Model</th>
<th>MS</th>
<th>BDR</th>
<th>Gated Sum</th>
<th>CCL</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.7%</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>46.3%</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>48.1%</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>48.5%</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>49.2%</td>
</tr>
<tr>
<td>ResNet-101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>42.5%</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>48.3%</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>51.6%</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>52.2%</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>53.4%</td>
</tr>
</tbody>
</table>

MS means multi-scale testing. The proposed CCL, gated sum and BDR improve the performance visibly.

**TABLE VII**
RUNTIME COMPARISON

<table>
<thead>
<tr>
<th>Model</th>
<th>BDR</th>
<th>Gated Sum</th>
<th>CCL</th>
<th>Time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-101</td>
<td></td>
<td></td>
<td></td>
<td>138.9</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>140.8</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>158.7</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>159.1</td>
</tr>
</tbody>
</table>

Runtime comparison on a single NVIDIA Tesla K80. It shows that the proposed network marginally increase the running time from the baseline.
E. Ablation Study and Runtime comparison

An ablation experiment of the proposed models is presented in Table VI. We provide an insight into the effect of the proposed methods by a detailed experimental study that illustrates how the segmentation results are affected by each of the proposed methods as well as their combinations. As shown in Table VI, the proposed CCL, gated sum and BDR improve the performance respectively, which shows the significance of this work.

The average inference time of proposed segmentation networks on Pascal Context images are listed in Table VII (on a single NVIDIA Tesla K80). It shows that the proposed network marginally increase the running time from the baseline.

F. Comparison with the State-of-the-Arts

Pascal Context [52] contains 10103 images from Pascal VOC 2010, and these images are re-annotated as pixel-wise segmentation maps. There are 4998 images for training and 5105 images for testing in Pascal Context. We use the most common 59 categories in this dataset for evaluation. A few examples on validation set of Pascal Context are shown in Fig. 11. Compared with the baseline, our segmentation network performs better at global information, salient objects, stuff and inconspicuous objects and has a robust adaptability to multi-scale objects. Quantitative results of Pascal Context are shown in Table VIII. It shows that our segmentation network outperforms the state-of-the-arts by a large margin across all evaluation metrics.

SUN-RGBD [67] provides pixel-wise labeling for 37 categories. It has 10335 indoor images which are from SUN3D [74], NYUDv2 [65], Berkeley B3DO [34] and the newly captured images. The training set has 5285 images and the test set contains 5050 images. We only use the RGB modality as input for training and testing. Quantitative results of SUN-RGBD are reported in Table IX. It shows that our segmentation network outperforms the state-of-the-arts by a large margin for all the three evaluation metrics.

Sift Flow [44] contains 2688 images labeled by 33 classes. We use the standard split protocol thus 2488 images are used for training and 200 images are used for testing. Quantitative results of Sift Flow are shown in Table X. Our segmentation
networks outperforms state-of-the-arts across all evaluation metrics.

**COCO Stuff** [7] contains 10000 images from Microsoft COCO dataset [43], out of which 9000 images are for training and 1000 images for testing. The unlabeled stuff pixels in original images of Microsoft COCO are further annotated with additional 91 classes in COCO Stuff. Herein, this dataset contains 171 categories including objects and stuff annotated to each pixel. Quantitative results of COCO Stuff are shown in TABLE XI. On this more challenging dataset with much more number of classes, our proposed scene segmentation network outperforms the state-of-the-arts by a large margin across all evaluation metrics, which demonstrates the capacity of the proposed CGBNet to handle complex dataset. Recently, COCO Stuff is updated to 2017 version, in which more annotated images for segmentation are provided and the split is adjusted to 40000 training images and 5000 validation images. We also test our segmentation network on this new challenging dataset, shown in TABLE XI. There is no other published result on this newly enlarged dataset for comparison. The results of DSSPN that trained with extra data (marked with † in TABLE XI) are better than ours. DSSPN builds a semantic neuron graph by explicitly incorporating the semantic concept hierarchy into network construction. In such a way, DSSPN could exploit intrinsic taxonomy and semantic hierarchy of segmentation datasets and overcome the issue of the label discrepancy among different datasets, which effectively arms the segmentation network with the ability of training different datasets within a unified segmentation model. DSSPN is orthogonal to our methods. They can be applied to the segmentation network together. We believe that, if not considering about the GPU memory limitation, our approach could be further improved with such semantic concept hierarchy architecture in term of classification. We could train our model more effectively across several segmentation datasets using DSSPN.

**ADE20K** [84] is a recently released dataset for scene parsing which has 20210 training, 2000 validation and 3352 test images. This dataset contains 150 categories including object and stuff annotated to each pixel. We report our results on 2000 validation images in TABLE XII. We achieve state-of-the-art results on this dataset.

**Cityscapes** [12] is a recent street scene dataset which contains 5000 images with pixel-level fine annotations. There are 2975 training images, 500 validation images and 1525 testing images. And 19 classes (e.g. roads, bicycles and cars) are considered for evaluation on the testing sever provided by the organizers. The test results are shown in TABLE XIII.

### TABLE X

**Performance Comparison on Sift Flow Dataset**

<table>
<thead>
<tr>
<th>Networks</th>
<th>GPA</th>
<th>ACA</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu et al. [45]</td>
<td>76.7%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tighe et al. [69]</td>
<td>75.6%</td>
<td>41.1%</td>
<td>-</td>
</tr>
<tr>
<td>Farabet et al. [18]</td>
<td>78.5%</td>
<td>29.6%</td>
<td>-</td>
</tr>
<tr>
<td>Pinheiro et al. [55]</td>
<td>77.7%</td>
<td>29.8%</td>
<td>-</td>
</tr>
<tr>
<td>Sharma et al. [61]</td>
<td>79.6%</td>
<td>33.0%</td>
<td>-</td>
</tr>
<tr>
<td>Yang et al. [76]</td>
<td>79.8%</td>
<td>48.7%</td>
<td>-</td>
</tr>
<tr>
<td>FCN-8s [62]</td>
<td>85.9%</td>
<td>53.9%</td>
<td>41.2%</td>
</tr>
<tr>
<td>DAG-RNN + CRF [64]</td>
<td>87.8%</td>
<td>57.8%</td>
<td>44.8%</td>
</tr>
<tr>
<td>Context + CRF [42]</td>
<td>88.1%</td>
<td>53.4%</td>
<td>44.9%</td>
</tr>
<tr>
<td><strong>CGBNet</strong></td>
<td><strong>89.7%</strong></td>
<td><strong>58.5%</strong></td>
<td><strong>46.8%</strong></td>
</tr>
</tbody>
</table>

*The proposed CGBNet segmentation network outperforms state-of-the-arts across all evaluation metrics.*

### TABLE XI

**Performance Comparison on COCO Stuff Dataset**

<table>
<thead>
<tr>
<th>Networks</th>
<th>GPA</th>
<th>ACA</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN [7]</td>
<td>52.0%</td>
<td>34.0%</td>
<td>22.7%</td>
</tr>
<tr>
<td>DeepLab [10]</td>
<td>57.8%</td>
<td>38.1%</td>
<td>26.9%</td>
</tr>
<tr>
<td>DAG-RNN [64]</td>
<td>62.2%</td>
<td>42.3%</td>
<td>30.4%</td>
</tr>
<tr>
<td>RefineNet-Res101 [41]</td>
<td>65.2%</td>
<td>45.3%</td>
<td>33.6%</td>
</tr>
<tr>
<td>DSSPN-Res101 [39]</td>
<td><strong>68.5%</strong></td>
<td><strong>47.0%</strong></td>
<td><strong>36.2%</strong></td>
</tr>
<tr>
<td><strong>CGBNet</strong></td>
<td><strong>67.4%</strong></td>
<td><strong>49.9%</strong></td>
<td><strong>36.9%</strong></td>
</tr>
</tbody>
</table>

*The proposed CGBNet outperforms the state-of-the-arts by a large margin across all evaluation metrics. Methods trained with extra data are marked with †.*

### TABLE XII

**Parsing Performance of Different Networks on ADE20K Dataset.**

<table>
<thead>
<tr>
<th>Networks</th>
<th>GPA</th>
<th>ACA</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SegNet [2]</td>
<td>71.0%</td>
<td>31.1%</td>
<td>21.6%</td>
</tr>
<tr>
<td>Cascaded-SegNet [84]</td>
<td>71.8%</td>
<td>37.9%</td>
<td>27.5%</td>
</tr>
<tr>
<td>FCN [62]</td>
<td>71.3%</td>
<td>40.3%</td>
<td>29.4%</td>
</tr>
<tr>
<td>DilatedNet [79]</td>
<td>74.1%</td>
<td>47.9%</td>
<td>32.3%</td>
</tr>
<tr>
<td>DAG-RNN [64]</td>
<td>73.9%</td>
<td>49.0%</td>
<td>33.5%</td>
</tr>
<tr>
<td>Cascaded-DilatedNet [84]</td>
<td>74.5%</td>
<td>45.4%</td>
<td>34.9%</td>
</tr>
<tr>
<td>RefineNet-Res152 [41]</td>
<td>-</td>
<td>-</td>
<td>40.7%</td>
</tr>
<tr>
<td>PSPNet-Res101 [82]</td>
<td>80.6%</td>
<td>-</td>
<td>42.0%</td>
</tr>
<tr>
<td>DSSPN-Res101-Universal† [39]</td>
<td>81.1%</td>
<td>-</td>
<td>43.7%</td>
</tr>
<tr>
<td>EncNet [81]</td>
<td>81.7%</td>
<td>-</td>
<td>44.7%</td>
</tr>
<tr>
<td><strong>CGBNet</strong></td>
<td><strong>82.1%</strong></td>
<td><strong>58.2%</strong></td>
<td><strong>44.9%</strong></td>
</tr>
</tbody>
</table>

*The proposed CGBNet outperforms the state-of-the-arts by a large margin across all evaluation metrics.*

V. CONCLUSION

In this paper, we address the challenging task of scene segmentation. Scene segmentation aims at parsing an image into a set of coherent semantic regions and classifying each pixel to one of classes. Therefore, the context and multi-scale aggregation are crucial to achieve good segmentation. However, DCNN designed and trained for image classification tends to extract abstract features of dominated objects, which weakens or even disregards some essentially discriminative information for inconspicuous objects and stuff. To address this issue, we propose a novel context-contrasted local feature to leverage the useful context and spotlight the local information in contrast to the context. The proposed context-contrasted local feature greatly improves the image segmentation performance, especially for inconspicuous objects and stuff. Adding skip layers is a common effective and efficient way to exploit multi-scale features, but the existing approaches indiscriminately fuse the score maps of skip layers via a simple summation. To achieve an optimal multi-scale aggregation, we propose a scheme of gated sum to selectively aggregate multi-scale features for each pixel. The parameters of gates are...
TABLE XIII

<table>
<thead>
<tr>
<th>Networks</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF-RNN</td>
<td>62.5%</td>
</tr>
<tr>
<td>FCN [62]</td>
<td>65.3%</td>
</tr>
<tr>
<td>SCNN [37]</td>
<td>66.3%</td>
</tr>
<tr>
<td>DPFN [49]</td>
<td>66.8%</td>
</tr>
<tr>
<td>DilatedNet [79]</td>
<td>66.8%</td>
</tr>
<tr>
<td>LRR [20]</td>
<td>69.7%</td>
</tr>
<tr>
<td>DeepLab-v2+CRF [9]</td>
<td>70.4%</td>
</tr>
<tr>
<td>Context + CRF [42]</td>
<td>71.6%</td>
</tr>
<tr>
<td>RefineNet-RS101 [41]</td>
<td>73.6%</td>
</tr>
<tr>
<td>DUC [71]</td>
<td>77.6%</td>
</tr>
<tr>
<td>PSPNet [82]</td>
<td>78.4%</td>
</tr>
<tr>
<td>DFN [78]</td>
<td>79.5%</td>
</tr>
<tr>
<td>DenseASPP(DenseNet-161) [77]</td>
<td>80.6%</td>
</tr>
</tbody>
</table>

The proposed CGBNet segmentation networks outperform state-of-the-arts.

generated from the testing image by the proposed networks learnt from the training data. Thus, they are adaptive not only to the training data, but also to the specific testing image. We further propose a boundary refinement model to delineate the prediction of boundary. It encourages the very low-level features near the boundary and suppresses them far from it because the very-low-level features in terms of the signal-to-noise ratio often tend to be informative near the boundary but noisy far from it. Without bells and whistles, our segmentation network achieves state-of-the-art performance consistently on the six popular scene segmentation datasets used in the evaluation, Pascal Context, SUN-RGBD, Sift Flow, COCO Stuff, ADE20K and Cityscapes.

ACKNOWLEDGMENT

This research was supported by Singapore Ministry of Education Academic Research Fund AcRF Tier 3 Grant no: MOE2017-T3-1-001.

REFERENCES


Henghui Ding received the B.E. degree from Xi’an Jiaotong University, Xi’an, China, in 2016. He is currently pursuing the Ph.D. degree with the Rapid-Rich Object Search (ROSE) Laboratory, School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. His research interests include computer vision and machine learning.

Xudong Jiang (M02SM06) received the B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, and the Ph.D. degree from Helmut Schmidt University, Hamburg, Germany. After several years teaching as a Lecturer in UESTC, he was with the Institute for Infocomm Research, A*STAR, Singapore, as a Lead Scientist, and the Head of the Biometrics Laboratory from 1998 to 2004, where he developed a system that achieved the most efficiency and the second most accuracy at the International Fingerprint Verification Competition in 2000. He joined Nanyang Technological University (NTU), Singapore, as a Faculty Member, in 2004, where he has also served as the Director of the Centre for Information Security from 2005 to 2011. He is currently a tenured Associate Professor with the School of EEE, NTU. He holds 7 patents and has authored over 150 papers with 37 papers in the IEEE journals, including 11 papers in IEEE TIP and 6 papers in IEEE TPAMI. His research interests include signal/image processing, pattern recognition, computer vision, machine learning, and biometrics. He has been an IFS Technical Committee Member of the IEEE Signal Processing Society, and he has served as Associate Editors for IEEE TIP, IEEE SPl, and IET Biometrics.

Ai Qun Liu received the B.E. degree from Xi’an Jiaotong University, Xi’an, China, the M.Sc. degree from the Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree from the National University of Singapore, Singapore in 1982, 1988, and 1994, respectively. He is currently a Professor with School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore. He serves as an Editor, Co-Editor, and Editorial Board Member of several scientific journals. He is the author or coauthor of more than 130 publications including peer-reviewed journal papers. Dr. Liu has won several awards including the IES awards.

Bing Shuai received the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2018. He is currently a Research Scientist at Amazon AWS Rekognition. His research interests include computer vision (specifically in) scene segmentation, video segmentation, and video action detection.

Gang Wang received the B.Eng. degree in electrical engineering from the Harbin Institute of Technology and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He was an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University. He had a joint appointment at the Advanced Digital Science Center, Singapore, as a Research Scientist, from 2010 to 2014. He is currently a Researcher/Senior Director and a Distinguished Scientist at Alibaba AI Labs. He was a recipient of MIT Technology Review Innovators Under 35 Award (Asia). He is an Area Chair of ICCV 2017, CVPR 2018 and ICCV 2019. He is an Associate Editor of TPAMI.