ZigZagNet: Fusing Top-Down and Bottom-Up Context for Object Segmentation

Di Lin[1] Dingguo Shen[1] Siting Shen[1] Yuanfeng Ji[1] Dani Lischinski[2]
Daniel Cohen-Or[1] Hui Huang[1*]
[1]Shenzhen University [2]The Hebrew University of Jerusalem

Abstract

Multi-scale context information has proven to be essential for object segmentation tasks. Recent works construct the multi-scale context by aggregating convolutional feature maps extracted by different levels of a deep neural network. This is typically done by propagating and fusing features in a one-directional, top-down and bottom-up, manner. In this work, we introduce ZigZagNet*, which aggregates a richer multi-context feature map by using not only dense top-down and bottom-up propagation, but also by introducing pathways crossing between different levels of the top-down and the bottom-up hierarchies, in a zig-zag fashion. Furthermore, the context information is exchanged and aggregated over multiple stages, where the fused feature maps from one stage are fed into the next one, yielding a more comprehensive context for improved segmentation performance. Our extensive evaluation on the public benchmarks demonstrates that ZigZagNet surpasses the state-of-the-art accuracy for both semantic segmentation and instance segmentation tasks.*

1. Introduction

Object segmentation is a long standing challenging problem in computer vision. It encompasses a variety of tasks including semantic and instance segmentation. Recent advanced segmentation methods have significantly improved the accuracy of object segmentation, leveraging the power of deep convolutional neural networks (CNNs) to learn from large-scale datasets.

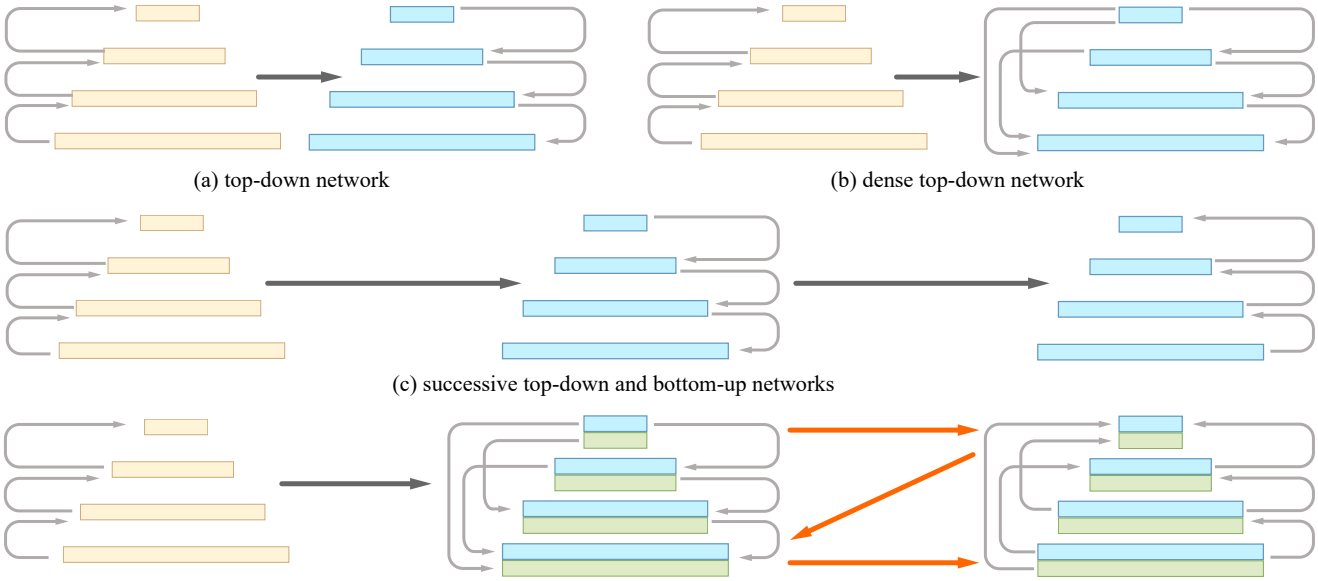
One of the difficulties in localizing instances of objects stems from the fact that objects in natural images may appear at a diversity of scales. Since CNNs [18, 38, 16, 40, 6] consist of convolutional feature maps at various spatial resolutions, recent object segmentation methods [25, 2, 10] have used convolutional feature maps from different CNN levels to represent content at different scales.

Different convolutional feature maps have correlated information, forming a multi-scale context for object segmentation. Thus, most recent approaches further utilize top-down networks [35, 37, 21, 19, 31, 32, 22, 33, 30, 5] (see Figure 1(a)), dense top-down networks [1, 42] (see Figure 1(b)) and successive top-down/bottom-up networks [29, 20, 24] (see Figure 1(c)) to communicate between different levels. The motivation underlying such top-down/bottom-up networks is to propagate multi-scale context across different scales, thereby augmenting the feature maps at different levels. State-of-the art methods [29, 22, 30, 24, 42], however, propagate context information *only along a single direction*.

In this paper, we advocate the idea of exchanging and combining top-down and bottom-up context to enrich the context information encoded by each feature map. In this scheme, the top-down network propagates high-level large scale semantic information down to shallow network layers, while the bottom-up network encodes the smaller scale visual details into deeper network layers. Unlike one-directional network architectures [29, 25, 2, 10, 37, 22, 33, 30, 24, 1, 42], our approach iteratively fuses feature maps between the top-down and bottom-up networks, gradually refining the aggregated multi-scale context information.

More specifically, we introduce *ZigZagNet*, a new scheme for fusing multi-scale context information, illustrated in Figure 1(d). The backbone network (left) extracts a progression of convolutional feature maps for the top-down network (middle), where each pair of feature maps is connected with a top-down pathway. Here, each feature map is sensitive to the context information of all higher-level feature maps. The top-down network produces a new set of feature maps, which are fed into the bottom-up network (right). Similarly to the top-down pathways, the bottom-up pathways enhance each feature map with all lower-level feature maps. The feature maps produced by the top-down and bottom-up networks are exchanged in a zig-zag fashion, and fused to aggregate the context information from all levels. The resulting feature maps are then used by a new round of top-down and bottom-up context propagation. Finally, the fused feature maps at the last stage are used for the segmentation task.

*Hui Huang is the corresponding author of this paper.



(a) top-down network

(b) dense top-down network

(c) successive top-down and bottom-up networks

(d) ZigZag network

Figure 1. Different approaches for propagating multi-scale context information. The top-down networks (a) and (b) use the deeper layers to augment the shallower ones. The feature maps produced by the top-down network are further passed to the bottom-up network (c). In contrast to (a)-(c), ZigZagNet (d) exchanges feature maps between the top-down and bottom-up networks to achieve a richer encoding of multi-scale context. The orange blocks on the left represent the feature maps of backbone networks. The blue and green blocks represent the feature maps produced at different stages. For conceptual illustration, we omit some overlapping pathways and only show a subset of the dense pathways between feature maps in (d). Figure 2 depicts the ZigZagNet architecture in more detail.

In addition to exchanging information between the top-down and the bottom-up networks, each of the two networks employs a novel *Region Context Encoding* (RCE) scheme, which captures context information of multi-scale subregions of the feature maps. We subdivide each feature map into regions, which propagate information to each other. By using different subdivisions of the feature map to compute contextual features, we achieve a richer encoding of the context in multi-scale subregions. The encoded context is propagated via dense pathways, modeling relationships between subregions of different feature maps.

Our ZigZagNet architecture is applicable to an array of object segmentation tasks. We show its effectiveness by evaluating it on the public benchmarks for semantic segmentation (e.g., PASCAL Context dataset [28] and PASCAL VOC 2012 dataset [9]) and also for instance segmentation (e.g., COCO dataset [23]). We surpass state-of-the-art performance on the PASCAL Context dataset [28] and PASCAL VOC 2012 validation set [9]. On the PASCAL VOC 2012 test set, our performance is competitive with that reported by Chen et al. [5], who use a private JFT dataset [17, 6, 39] to pre-train the backbone network. On the challenging COCO dataset [23], our approach is applied with different backbone networks and detectors, yielding consistent improvement of the segmentation accuracy. We compare our single model to the previous methods individually, and we achieve the state-of-the-art result on the COCO test-dev set. Our code package and models will be publicly available.

2. Related Work

The literature on image segmentation is vast [26, 3, 43, 21, 1, 42, 20, 22, 15, 24]. In the following, we will mainly survey semantic segmentation [9, 28] and instance segmentation [14, 23] techniques, which are closely related to our work in the sense that they combine convolutional feature maps from different levels in order to aggregate multi-scale context information.

Semantic Segmentation Semantic segmentation methods aim to provide pixel-wise labels for objects. Fully convolutional networks (FCNs) [26] have been used for semantic segmentation and achieved tremendous progress. Due to the down-sampling operations, the convolutional feature maps have progressively coarser resolutions. Thus, using high-level feature maps [3, 43, 4] for semantic segmentation inevitably loses spatial context of objects. To resolve this problem, dilated convolution (also known as atrous convolution) has been used to preserve the resolutions of feature maps in more detail. However, atrous convolution produces many high-resolution feature maps that require an overly large budget of GPU memory. To save GPU memory and improve the segmentation accuracy, the backbone FCN is

followed by the top-down network [35, 21, 31, 32, 5] (a.k.a., the encoder-decoder network), which is used to propagate the high-level semantic information and combine it with the spatial details of low-level feature maps, yielding a high-resolution feature map with multi-scale context information. Rather than communicating only between adjacent feature maps, dense pathways [1, 42] are used to propagate top-down context between all pairs of feature maps. Still, the high-level feature maps do not have any lower-level context information to enrich their own expressive power.

Unlike the above semantic segmentation methods, we use dense pathways in both top-down and bottom-up directions, affecting the feature maps with all levels of context information. In the most recent work, Lin et al. [20] also exchanges top-down and bottom-up context; however, the feature traffic takes place between adjacent feature maps only. Thus, it requires multiple stages to propagate the context beyond adjacent feature maps, which may decay important information. In contrast, our dense pathways directly communicate all feature maps at each stage of context propagation, enabling a direct, effective augmentation of the feature maps at all levels.

Our dense pathways employ region context encoding to capture the context of subregions in different feature maps. Instead of constructing the context of various subregion as a global representation [27, 24], we compute context information of multi-scale subregions by examining multiple subdivisions of each feature map. Note that our method is different from the traditional spatial pyramid pooling [3, 43, 4], which uses adjacent subregions to produce the context feature. We propagate information between all subregions, providing more effective context for segmentation tasks.

Instance Segmentation In addition to pixel-wise labels, instance segmentation further aims to differentiate individual objects. Similar to the encoder-decoder network used for semantic segmentation, the top-down network [37, 19, 22, 33, 30] has been applied with the FCN backbone for instance segmentation. Unlike semantic segmentation methods [26, 21, 31, 5, 1, 42, 20] that produce a high-resolution feature map for predicting pixel-wise labels, instance segmentation methods [37, 19, 22, 33, 30, 29, 24] use all levels of feature maps to better capture object instances with different scales.

Recently, successive top-down and bottom-up networks [29, 24] have been used to learn more powerful feature maps at different levels. Specifically, the path-aggregation network [24] appends a bottom-up network following the top-down network, building a shortcut for the information propagation between the top-most and the bottom-most feature maps. The hourglass network [29] repeats top-down and bottom-up feature propagation to distill multi-scale context information. Nonetheless, in this network, at each iteration, the feature maps only receive information of the highest-resolution feature map from the previous iteration, inevitably ignoring the context information of lower-resolution feature maps. In our work, we fuse the feature maps produced by the top-down and the bottom-up networks at all levels, employing all of them in all iterations. This implies that during all stages of top-down and bottom-up context exchange, the entire context information is available for learning effective features.

3. ZigZag Network

Feature maps at all levels can benefit from context information that has been aggregated from all scales. However, recent methods only establish one-way connections between the top-down and bottom-up networks (see Figure 1(a) and (c)), where feature maps from adjacent levels propagate context to affect each other. Even though the latest works [1, 42] use dense pathways (see Figure 1(b)) to strengthen context propagation between multi-scale feature maps, only the highest-resolution feature map perceives the entire context.

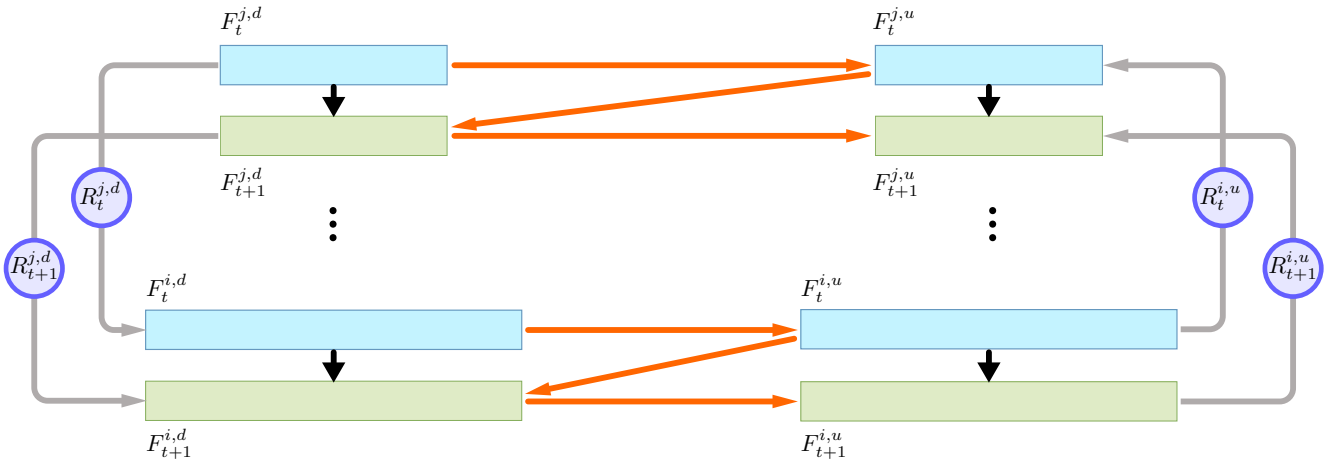
Here, we propose the ZigZagNet architecture, where each feature map is directly enhanced by multi-scale context extracted from all the other maps. More specifically, ZigZagNet consists of two networks, a top-down network, and a bottom-up network, as illustrated in Figure 2. Each network has dense connections between its layers, with each such connection carrying a feature map augmented by its multi-scale context using region context encoding (RCE), as described in more detail in Section 4. In addition to these dense pathways, there are also pathways that exchange information between the two networks, by connecting between feature maps of the same level of the top-down and the bottom-up pyramids (the red arrows in Figure 2). The propagation of context within each network and between the networks is iterated over several stages.

More formally, let t denote the stage ($0 \leq t < T$), and $F_t^{i,d}$ and $F_t^{i,u}$ denote the t-th stage feature maps at the i-th level of the top-down and the bottom-up networks, respectively. At each iteration, we fuse feature maps $F_t^{i,d}$ and $F_t^{i,u}$ to yield $F_{t+1}^{i,d}$, and maps $F_{t+1}^{j,d}$ and $F_t^{j,u}$ to yield $F_{t+1}^{j,u}$. As a result, context information is propagated between the two networks in a zig-zag fashion. Figure 2 illustrates this process, by showing each feature map in two consecutive stages: stage t in blue and stage $t+1$ in green.

Specifically, in stage $t+1$, the top-down network computes the feature map $F_{t+1}^{i,d} \in \mathbb{R}^{H \times W \times C}$ as:

$$F_{t+1}^{i,d} = P_{t+1}^{i,d} + \prod_{j=i+1}^{L} R_{t+1}^{j,d}, \tag{1}$$

where $t = 0, \ldots, T-1$. The total number of stages T is set to 3 in our experiments. We model the feature map $F_{t+1}^{i,d}$



(a) top-down network (b) bottom-up network

Figure 2. Top-down and bottom-up context propagation in ZigZagNet. The gray arrows of the top-down network (a) and bottom-up network (b) represent the dense pathways between different levels of feature maps. The red arrows iteratively exchange the context information between the top-down and bottom-up networks, which generate all levels of feature maps over multiple iterations. Here, we show only two different levels of feature maps to simplify the illustration. The blue and green blocks represent feature maps computed in two successive iterations.

by summing the product of the set of context feature maps from higher levels $\{R_{t+1}^{j,d} | j > i\}$ with a fused feature map $P_{t+1}^{i,d} \in \mathbb{R}^{H \times W \times C}$, defined as:

$$P_{t+1}^{i,d} = \begin{cases} B^i & t = 0, \\ \sigma(W_{t+1}^{i,d} * (F_t^{i,d} + F_t^{i,u})) & \text{otherwise.} \end{cases} \quad (2)$$

Above, $W_{t+1}^{i,d}$ is a convolution kernel and σ denotes the ReLU activation function. Initially (i.e., $t = 0$), we use B^i, the feature map computed by the backbone FCN to construct $F_1^{i,d}$. In the following iterations, we fuse $F_t^{i,d}$ and $F_t^{i,u}$, which are produced by the top-down and bottom-up networks in the previous iteration, by convolving and activating their sum. Thus, differently from the one-way context propagation in previous works, our top-down network receives the previous iteration of top-down as well as the bottom-up context to refine the new feature map $F_{t+1}^{i,d}$. Furthermore, we use region context encoding (RCE) to generate the context feature map $R_{t+1}^{j,d}$ based on the subregions of $F_{t+1}^{j,d}$. As described in Section 4, RCE encodes the relationship of subregions into the context feature maps. By using various scales of subregions, we provide $F_{t+1}^{i,d}$ with richer context.

Similarly, we use the bottom-up network to compute the feature map $F_{t+1}^{j,u} \in \mathbb{R}^{H \times W \times C}$ as:

$$F_{t+1}^{j,u} = P_{t+1}^{j,u} + \prod_{i=1}^{j-1} R_{t+1}^{i,u}, \quad (3)$$

where

$$P_{t+1}^{j,u} = \begin{cases} F_{t+1}^{j,d} & t = 0, \\ \sigma(W_{t+1}^{j,u} * (F_{t+1}^{j,d} + F_t^{j,u})) & \text{otherwise,} \end{cases} \quad (4)$$

Note that here instead of fusing two feature maps from stage t, as done in Eq. (2), we fuse the maps $F_{t+1}^{j,d}$ and $F_t^{j,u}$. This is done, since at this point in the process $F_{t+1}^{j,d}$ is already available, and contains more refined information than $F_t^{j,d}$. Finally, the maps $\{F_T^{i,d}\}$ and $\{F_T^{i,u}\}$ are fused using Eq. (2) to yield the maps $\{P_{T+1}^{i,d}\}$ for segmentation.

Below, we focus on the dense top-down and bottom-up pathways equipped with the RCE to produce context feature maps $R_t^{j,d}$ and $R_t^{j,u}$. For clarity, we omit the notations d, u and t from this point onward.

4. Region Context Encoding

In this section, we elaborate on the region context encoding (RCE) mechanism that connects all subregions of the input feature map, enabling each subregion to diffuse its information flexibly. As illustrated in Figure 3, this is done using multiple parallel branches. We input the feature map produced by the top-down/bottom-up network to each RCE branch. In each branch, we partition the feature map into regular subregions. Next, we perform a weighted sum to aggregate all subregions into a global representation, which is then distributed to all subregions. This allows each subregion to pass information to all subregions of the new feature map. Each branch performs a different subdivision of the feature map, generating subregions at different scales. Finally, we add the feature maps produced by all branches to the input feature, which is propagated as the context feature map R^i to feature maps at other levels. Thus, thanks to the RCE mechanism, subregions of various scales can consequently affect any position of other feature maps.

In more detail, given the feature map $F^i \in \mathbb{R}^{H \times W \times C}$

produced by the top-down/bottom-up network, we convolve and partition it into $K \times K$ subregions. By summing the neurons within each subregion, we produce a feature map $M^i_{K\times K} \in \mathbb{R}^{K\times K\times C}$ as:

$$M^i_{K\times K}(x,y,c) = \sum_{(h,w)\in S(x,y)} F^i(h,w,c),$$
$$x = 1,...,K, \;\; y = 1,...,K, \tag{5}$$

where (x, y) indicates the location of $M^i_{K\times K}$. We use $S(x, y)$ to denote a subregion that includes a set of neurons in $M^i_{K\times K}$. Thus $M^i_{K\times K}(x, y)$ is the feature of the subregion $S(x, y)$. We sum all subregion features by adapting to their importance, yielding a global representation for connecting all subregions. For this purpose, we can simply apply a learnable $K \times K$ convolution with ReLU activation to $M^i_{K\times K}$ without padding. This results in a C-dimensional feature vector (see Figure 3(c)). Another learnable $K \times K$ kernel is used to deconvolve this C-dimensional vector without padding, yielding a new feature map $Q^i_{K\times K} \in \mathbb{R}^{K\times K\times C}$. The $Q^i_{K\times K}$ maps from all branches are then added to the input feature map F^i to yield the feature map $R^i \in \mathbb{R}^{H\times W\times C}$:

$$R^i(h,w) = F^i(h,w) + \sum_{K\in\{3,5,7\}} Q^i_{K\times K}(x,y),$$
$$(h,w) \in S(x,y). \tag{6}$$

In Eq. (6), we empirically use the 3 different subdivisions of F^i (i.e., 3×3, 5×5 or 7×7 subregions) to compute the set of feature maps $\{Q^i_{K\times K}\}$. By dividing the feature maps into more subregions, we dramatically increase the number of parameters to learn, but achieve negligible improvement.

5. Implementation Details

The ZigZagNet network was implemented using the Detectron platform[1]. We use ResNet-101 pre-trained on the ImageNet dataset [8] as the backbone network. The layers *res2*, *res3*, *res4* and *res5* are used as the initial $\{B_1, B_2, B_3, B_4\}$ feature maps in Eq. (2). Three stages of fusing and exchanging context information are performed ($T = 3$). The fused feature maps $\{P^{1,d}_4, P^{2,d}_4, P^{3,d}_4, P^{4,d}_4\}$ are used for the object segmentation tasks. We optimize the network with the SGD solver. We evaluate our method on the semantic segmentation and instance segmentation tasks.

Semantic Segmentation Network Following the training strategy described in [43, 21, 4, 5], we adapt images of the COCO dataset [23] to fine-tune the network at the beginning. The feature map $P^{1,u}_4$, which has the highest spatial resolution is used to regress the pixel-wise categories. We use the softmax loss to penalize the pixel-wise errors.

[1] https://github.com/facebookresearch/Detectron

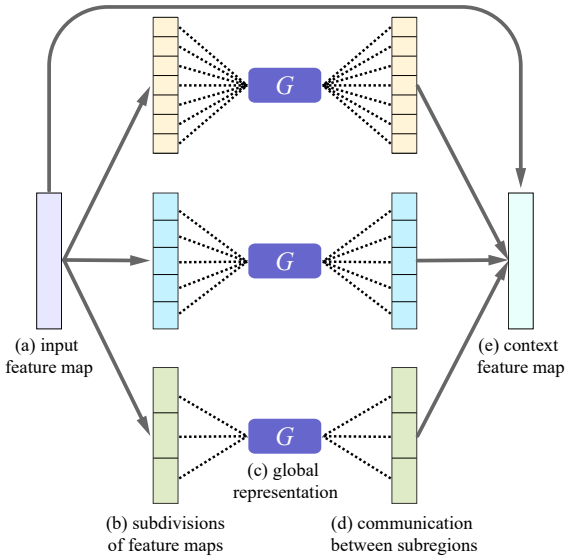


Figure 3. Region context encoding. We use separate convolutional layers to process the input feature map (a), and the results are used to compute features of subregions (b). In each branch, we produce a global representation (c) to connect all subregions. This global representation is used to propagate information between all subregions (d). We use different subdivisions of the input feature map in separate branches. Finally, we add results of all branches to the input feature map, yielding the context feature map (e).

We employ image flipping, cropping, scaling and rotation to prepare the mini-batches for the network's training. Each mini-batch contains 16 images, and each image has the uniform resolution of 473×473. We first use 60K mini-batches with a learning rate of $1e{-}3$, and then decay the learning rate to $1e{-}4$ for additional 60K mini-batches. During testing, we resize each image using four factors (i.e., $\{0.6, 0.8, 1.0, 1.2, 1.4\}$). The five resulting images are fed to the trained model to compute predictions separately. The predictions are averaged to obtain the final result.

Instance Segmentation Network We use all of the fused feature maps (i.e., $\{P^{1,d}_4, P^{2,d}_4, P^{3,d}_4, P^{4,d}_4\}$) to compute the mask for each object [34, 22]. According to the size of the object's bounding box [22], we select one of the feature maps for extracting the ROI feature [12, 15] as the object representation. We use three loss functions, i.e., softmax loss for classification, smooth L_1 loss for bounding-box regression and pixel-wise softmax loss for mask regression.

We rescale the image's shorter edge to 800, keeping the aspect ratio of the image for training and testing the network. Each mini-batch has 8 images. We use the warmup strategy [13] at the beginning of fine-tuning the network. During training, we decay the learning rate by using 0.01, 0.001 and 0.0001, along with 200K, 60K and 40K mini-batches. We use the NMS with a threshold of 0.5 to reduce overlapping segmentation results.

context exchange	feature fusion	dense pathways	mIoU
			82.5
✓		✓	83.6
	✓	✓	84.2
✓	✓		84.9
✓	✓	✓	**86.0**

Table 1. Ablation experiments on the PASCAL VOC 2012 validation set. Segmentation accuracy is measured by mIoU (%).

6. Experiments

We evaluate our method on three public benchmarks, i.e., PASCAL VOC 2012 [9], PASCAL Context [28] and COCO [23] datasets. We use the PASCAL VOC 2012 [9] and PASCAL Context [28] datasets to evaluate the semantic segmentation accuracy in terms of mean Intersection-over-Union (mIoU). For the instance segmentation task, we evaluate our method on COCO dataset [23]. We show the mask average precision (mask AP), which is the standard COCO metric computed over different mask IoU thresholds.

6.1. Results on PASCAL VOC 2012 and Context Datasets

The PASCAL VOC 2012 dataset contains 10,582 training images associated with 20 object categories and background. The PASCAL Context dataset contains 4,998 training images with 59 categories and background. We mainly use the PASCAL VOC 2012 validation set (1,449 images) to evaluate the effectiveness of our approach. We also report segmentation accuracies on the PASCAL VOC 2012 test set (1,456 images) and the PASCAL Context validation set (5,105 images) for comparisons with state-of-the-art methods.

Ablation Study of ZigZagNet Our ZigZagNet models the bidirectional interaction between the top-down and bottom-up networks, iteratively refining different levels of feature maps. The network has dense pathways equipped with the RCE to enrich the context information. We conduct an ablation study by removing the critical components, and examine the effect on the segmentation accuracy. We summarize the results in Table 1.

We fuse the feature maps to achieve the multi-scale context, exchanging the context information between top-down and bottom-up networks. By removing the additional bottom-up network and the dense pathways, we disable the context exchange and feature fusion. Thus the system degrades to the encoder-decoder architecture [5] and obtains the segmentation score of 82.5, significantly lower than the score of 86.0 achieved by our full model.

	method	mIoU
global context	Mostajabi et al. [27]	82.7
	Liu et al. [24]	83.0
	Peng et al. [31]	83.4
spatial pyramid pooling	Zhao et al. [43]	84.1
	Chen et al. [4]	84.7
multi-scale region context	SCF	85.2
	ours	**86.0**

Table 2. Comparisons with various approaches that use context information. "SCF" means summing context feature maps at different levels. Performance is evaluated on the PASCAL VOC 2012 validation set. We report the segmentation accuracy in terms of mIoU (%).

Next, we examine the impact of the feature map fusion on the segmentation accuracy. In ZigZagNet, the top-down and bottom-up networks produce feature maps that are fused to aggregate context at all levels. Without fusing feature maps, the network has only one pass of top-down and bottom-up context propagation. In this case, the feature maps of top-down network learn from higher-level context, but lacking of lower-level context for further refining themselves. It subsequently makes a negative impact on feature maps of the bottom-up network, and obtains the segmentation score of 83.6, once more a significant drop of performance compared to our full approach. By removing the context exchange only, we degrade ZigZagNet to one pass of top-down and bottom-up context propagation. In this case, the feature fusion helps achieve 84.2 IoU, which is better than the one pass of propagation without feature fusion (see the second case in Table 1 that achieves 83.6 IoU). But it still lags far behind our full model (86.0 IoU).

Finally, we study the importance of the dense pathways. The dense pathways employ the RCE to extract multi-scale region context of different levels of feature maps, which are merged to effectively augment the top-down and bottom-up context. With dense pathways removed, we disallow regions beyond adjacent feature maps to form useful context information. This reduces the score to 84.9.

Approaches for Using Region Context We design the RCE to model the relationship between multi-scale subregions. There are other approaches for using context information [27, 24, 31, 43, 4] to enhance the features of subregions. For fair comparisons, we employ these approaches in the ZigZagNet, in place of the RCE. The results are compared in Table 2.

First, we report the results of encoding global context into subregions. Mostajabi et al. [27] and Liu et al. [24] pro-

	VOC12 val set		VOC12 test set		Context val set	
	method	mIoU	method	mIoU	method	mIoU
top-down propagation	Chen et al. [5]	84.6	Chen et al. [4]	86.9	Chen et al. [3]	45.7
	Fu et al. [11]	84.8	Zhang et al. [42]	87.9	Lin et al. [21]	47.3
	Zhang et al. [42]	85.8	Chen et al. [5]	**89.0**	Zhang et al. [41]	51.7
successive propagation	Shah et al. [36]	79.0	Shah et al. [36]	84.3	Liu et al. [24]	50.1
	Fu et al. [11]	84.8	Fu et al. [11]	86.6	Shah et al. [36]	50.8
bidirectional propagation	Lin et al. [20]	85.1	Lin et al. [20]	88.0	Lin et al. [20]	50.3
	ours	**86.0**	ours	88.7	ours	**52.1**

Table 3. Comparisons with other state-of-the-art methods. The performances are evaluated on the PASCAL VOC 2012 validation set, test set and the PASCAL Context validation set. Segmentation accuracy is reported in terms of mIoU (%).

pose to use fully-connected layers to combine all subregions of the feature map as global context. By using different sets of parameters, the fully-connected layer learn global context information that is adaptive to each subregion. Rather than using the fully-connected operation, Peng et al. [31] employ convolutional layers with large kernels to produce global context, which is more transformation-aware. Compared to the above methods, which focus on the global scale of the image, our network leverages the multi-scale subregions to construct richer context, obtaining a higher segmentation score.

Next, we compare our network to approaches that use spatial pyramid pooling to construct context of subregions. Zhao et al. [43] apply spatial pyramid pooling to extract features within multi-scale subregions. Chen et al. [4] use different atrous convolution kernels to achieve learnable pyramid pooling, while saving computation compared to convolving with larger kernels. Note that spatial pyramid pooling [43, 4] computes context of adjacent subregions. Comparably, our method enables the information exchange between all subregions, which generally leads to 1.3 – 1.9 improvement of the segmentation score.

Instead of multiplying different levels of context feature maps in Eqs. (1) and (3), we experiment with the common way of adding different context feature maps to the feature map produced by top-down/bottom-up network. With this change, we observe a performance drop of 0.8 score compared to our full model. A similar observation is made by Zhang et al. [42]. We believe that the multiplication manner better models the interaction between context feature maps at different levels.

Comparisons with State-of-the-Art Methods In addition to the PASCAL VOC 2012 validation set, we report the results of our approach on the PASCAL VOC 2012 test set and the PASCAL Context validation set in Table 3. We compare our network to state-of-the-art approaches, which can be divided into three groups. The first group uses the (dense) top-down network to propagate context information. In the second group, successive top-down and bottom-up networks are used with one-way propagation of context information. Our approach and the context intertwining proposed by Lin et al. [20] belong to the third group, where bidirectional propagation of context is performed. It is noteworthy that ZigZagNet outperforms other methods on the PASCAL VOC 2012 validation set and the PASCAL Context validation set. On the PASCAL VOC 2012 test set, we achieve the score of 88.7 (see per-category accuracies on the PASCAL VOC leaderboard[2]). Our approach is competitive to the network proposed in [5], which uses a private JFT dataset [17, 6, 39] as additional data for training the backbone network. We show several semantic segmentation results of our method in Figure 4.

6.2. Results on COCO Dataset

We test our method on the COCO dataset [23] for instance segmentation. The COCO dataset contains about 120K training images with mask annotations for 80 object categories. We report our results on the COCO validation and test-dev sets, which have about 5K and 20K images respectively.

We use ZigZagNet to output multi-scale feature maps, which are then used by object detectors to extract features for regressing instance masks. Here, we experiment with three widely-used detectors, i.e., FCIS [19], Deformable RCNN [7] and Mask RCNN [15]. These detectors mainly contribute to champions of the COCO instance segmentation challenge from 2016 to 2018. We also evaluate the performance by using different backbone networks, i.e., ResNet-101 and ResNet-152. All results are reported in Table 4. Compared to different baseline models, our network improves the performance by 1 – 3 points. It demonstrates that our ZigZagNet is general to different detectors for achieving the performance gain on instance segmenta-

[2] http://host.robots.ox.ac.uk:8080/anonymous/N1OUN0.html

image ground-truth ours

Figure 4. Six semantic segmentation results produced by our method. The first three rows are from the PASCAL VOC 2012 validation set, and the last three rows are from the PASCAL Context validation set.

	backbone	mask AP w/ ZZNet
FCIS	ResNet-101	29.2 → **32.2**
	ResNet-152	31.7 → **33.4**
Deformable RCNN	ResNet-101	36.1 → **38.2**
	ResNet-152	37.9 → **39.8**
Mask RCNN	ResNet-101	37.5 → **39.5**
	ResNet-152	39.7 → **40.8**

Table 4. Comparisons with popular detectors for instance segmentation. Performance is evaluated on the COCO validation set. Accuracies are reported in terms of mask AP (%).

tion. We show several instance segmentation results of our method in Figure 5.

In Table 5, we compare our method with state-of-the-art models on test-dev set. Without the ensemble of different models and the multi-scale training/testing, all results are achieved by single models based on the ResNet-101 backbone for a fair comparison. Our result is better than others.

7. Conclusions

The latest progress in object segmentation benefits from deep neural networks trained on large-scale datasets and context information provided by multi-scale convolutional

Li et al. [19]	Dai et al. [7]	He et al. [15]	Liu et al. [24]	ours
29.6	35.7	37.1	40.0	**42.0**

Table 5. Comparisons with state-of-the-art single-model methods. Performance is evaluated on the COCO test-dev set. Accuracies are reported in terms of mask AP (%).

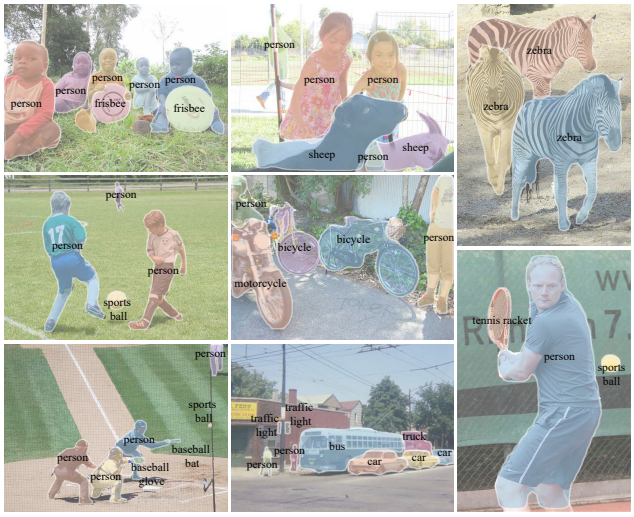


Figure 5. Several instance segmentation results produced by our method. The images are taken from the COCO validation set.

feature maps. In this paper, we have proposed ZigZagNet, where we establish bidirectional connections between the top-down and bottom-up networks. Our network has dense pathways to smooth the information propagation at all levels, encoding richer multi-scale context into the feature maps. The bidirectional connections are critical for fusing and exchanging context, progressively learning how to refine the feature maps with useful information. Our method outperforms the state-of-the-art on several public datasets, showing its effectiveness for object segmentation.

In future work, we plan to explore bidirectional context propagation in 3D segmentation tasks, which exhibit more complex relationship between objects. Additionally, we plan to design more efficient network architectures capable of computing context information at a lower computational cost.

Acknowledgments

We thank the anonymous reviewers for their constructive comments. This work was supported in parts by 973 Program (2015CB352501), NSFC (61702338, 61761146002, 61861130365), Guangdong Science and Technology Program (2015A030312015), Shenzhen Innovation Program (KQJSCX20170727101233642), LHTD (20170003), ISF-NSFC Joint Program (2472/17), and National Engineering Laboratory for Big Data System Computing Technology.

References

[1] P. Bilinski and V. Prisacariu. Dense decoder shortcut connections for single-pass semantic segmentation. In *CVPR*, 2018.

[2] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, 2016.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv*, 2016.

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv*, 2017.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv*, 2018.

[6] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[10] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv*, 2017.

[11] J. Fu, J. Liu, Y. Wang, and H. Lu. Stacked deconvolutional network for semantic segmentation. *arXiv*, 2017.

[12] R. Girshick. Fast r-cnn. In *ICCV*, 2015.

[13] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv*, 2017.

[14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017.

[20] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang. Multi-scale context intertwining for semantic segmentation. In *ECCV*, 2018.

[21] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *arXiv*, 2016.

[22] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[24] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[27] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015.

[28] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

[29] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[30] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. In *CVPR*, 2018.

[31] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, 2017.

[32] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, 2017.

[33] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[35] O. Ronneberger, P. Fischer, and T. Brox. U-Net: convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[36] S. Shah, P. Ghosh, L. S. Davis, and T. Goldstein. Stacked u-nets: a no-frills approach to natural image segmentation. *arXiv*, 2018.

[37] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv*, 2016.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

[39] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[40] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[41] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018.

[42] Z. Zhang, X. Zhang, C. Peng, D. Cheng, and J. Sun. Exfuse: Enhancing feature fusion for semantic segmentation. *arXiv*, 2018.

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv*, 2016.