Bidirectional Pyramid Networks for Semantic Segmentation

Dong Nie[1], Jia Xue[⋆2], and Xiaofeng Ren[1]

[1] Amap, Alibaba Group {dong.nie,x.ren}@alibaba-inc.com
[2] Rutgers University jia.xue@rutgers.edu

Abstract. Semantic segmentation is a fundamental problem in computer vision that has attracted a lot of attention. Recent efforts have been devoted to network architecture innovations for efficient semantic segmentation that can run in real-time for autonomous driving and other applications. Information flow between scales is crucial because accurate segmentation needs both large context and fine detail. However, most existing approaches still rely on pretrained backbone models (e.g. ResNet on ImageNet). In this work, we propose to open up the backbone and design a simple yet effective multiscale network architecture, *Bidirectional Pyramid Network* (BPNet). BPNet takes the shape of a *pyramid*: information flows from bottom (high-resolution, small receptive field) to top (low-resolution, large receptive field), and from top to bottom, in a systematic manner, at every step of the processing. More importantly, fusion needs to be efficient; this is done through an add-and-multiply module with learned weights. We also apply a unary-pairwise attention mechanism to balance position sensitivity and context aggregation. Auxiliary loss is applied at multiple steps of the pyramid bottom. The resulting network achieves high accuracy with efficiency, without the need of pretraining. On the standard Cityscapes dataset, we achieve test mIoU 76.3 with 5.1M parameters and 36 fps (on Nvidia 2080 Ti), competitive with the state of the time real-time models. Meanwhile, our design is general and can be used to build heavier networks: a ResNet-101 equivalent version of BPNet achieves mIoU 81.9 on Cityscapes, competitive with the best published results. We further demonstrate the flexibility of BPNet on a prostate MRI segmentation task, achieving the state of the art with a 45x speed-up.

1 Introduction

Semantic segmentation, detailed semantic understanding of a scene, is a fundamental problem in computer vision. Great progress has been made in recent years for semantic segmentation with convolutional neural networks (CNN), especially with the encoder-decoder architecture [1, 2]. In FCNs [1], convolution layers are stacked with subsampling to form the encoder, and deconvolution

[⋆] Work done during internship at Amap
Code is available at https://github.com/ginobilinie/BPNet

layers are stacked with upsampling to build the decoder. In such network architectures, higher layers are usually thought to capture entire objects, and contexts beyond objects, because they have larger receptive fields and go through more convolution steps; lower layers are more likely to capture local patterns, including part of objects and fine details. However, accurate semantic segmentation requires both large-scale contexts and small-scale details, and the integration of information across scales has been a central topic of investigation. This is particularly important for efficient (real-time) segmentation because we cannot use heavyweight backbones to preserve sufficient information in higher layers.

UNet [2], a successful architecture design that is still popular today, introduces skip connections between corresponding encoder and decoder layers to better model small-scale details in the decoder. This general line of approach has achieved good results with many improvements. VNet [3] proposes to use a convolutional layer to enhance the raw skip connection. AttentionUNet [4] utilizes an attention block to filter unrelated noise to improve the quality of feature fusion. UNet++ [5] redesigns skip pathways to reduce the semantic gap between the feature maps of the encoder and decoder. RefineNet [6] extracts high-resolution semantic information which is both accurate in location and rich in contextual information.

However, such skip connections cannot capture all useful interactions between large-scale information and small-scale information as they flow through the network. One major constraint is that most existing approaches use pre-trained backbone networks in the encoder. Typically, a ResNet (or VGG) backbone network is pre-trained on ImageNet on classification tasks, and then is fine-tuned as part of the semantic segmentation network. Relying on pre-training limits the possibility of adding interactions inside the backbone. Pre-training can also be an issue when we look at different domains such as medical imaging.

In this work we aim to remove the need for pre-training, and open up the backbone model to allow better flow of information across scales and processing steps. We propose a simple yet effective architecture design called *Bidirectional Pyramid Network* (**BPNet**), which symmetrically applies feature fusion between successive layers (as we move up the resolution pyramid toward larger receptive fields) and successive stages (as we apply more convolutions to extract segmentation cues on the same layer). With a very loose analogy, our BPNet design looks similar to the *Pascal's Triangle*, where adjacent numbers are combined to compute the next number. These cross-scale fusion "flows" are bidirectional: not only can they go from higher layers (larger scale, lower resolution) to lower layers (higher resolution); they can also go from lower layers to higher layers, hence facilitating further integration of useful information. In addition to this pyramid architecture, we also find it useful to employ a *parallel unary-pairwise attention* mechanism in order to help capture long-range dependencies and thin structures. An illustration of the BPNet design can be seen in Figure 1.

Our BPNet design can be instantiated in different forms w.r.t. computational requirement, by changing the number of resolution layers (partly corresponding to input image resolution), and the number of channels at each convolution step. We find that systematic information fusion in BPNet is most effective for efficient semantic segmentation, when the network is light- to medium-weight. One lightweight version, BPNet-S3-W32, achieves mIoU 76.3 on Cityscapes running 36 fps on Nvidia 2080 Ti, outperforming BiSeNet and other state-of-the-art real-time networks. One heavyweight version, BPNet-S4, achieves mIoU 81.9 on Cityscapes, competitive with the best published results. We also achieve state-of-the-art results on other benchmarks such as Camvid [7] and PASCAL Context [8]. It is worth noting that we do not use pre-training or external datasets; all our models are trained from scratch on the individual datasets themselves. This is in contrast to related works including HRNet [9], which did not use standard backbones but still pre-trained their networks on ImageNet. Without needing pre-training, our networks can be more versatile; we demonstrate the flexibility of BPNet on a popular medical imaging task on prostate MRI segmentation [10], achieving the state of the art with a 45x speed-up (using 2D convolutions only instead of 3D).

2 Related Work

2.1 Balancing Resolution and Semantics

A widely used semantic segmentation framework is the encoder-decoder [1, 2]. An encoder usually reduces the spatial resolution of feature maps to learn more abstract features. Correspondingly, the decoder recovers the spatial resolution of the input image from encoder so as to generate dense prediction maps. Fully convolutional network [1] is a typical encoder-decoder architecture which utilizes convolutions for encoding and deconvolutions for decoding to perform pixel-wise segmentation. UNet [2] combines shallow and deep features with skip connections to retain more details in the dense predictions. SharpMask [11] proposed a convolution in the skip connection between encoder and decoder layers to reduce the gap between semantics and localization. PANet [12] built a bottom-up connection between lower layers and the topmost layer to enhance the encoder-decoder's feature hierarchy with better localization and small-scale detail in the lower layers. HRNet [9] introduced multi-resolution convolution to fully fuse multi scale information, and the high-resolution pathway can well retain the localization information. In a related work on object detection, [13] proposed a weighted bi-directional feature pyramid network (BiFPN), showing that information flow in both directions (coarse-to-fine, and fine-to-coarse) are useful for feature fusion.

2.2 Context Aggregation

Context aggregation can be used to model long-range information dependency. Zhao et al. [14] proposed a pyramid pooling module to capture global contextual information. Chen et al. [15] used convolutions with different dilated ratios to harvest pixel-wise contextual information in different ranges. Wang et al. [16]

developed a non-local module, which generates a pixel-wise attention mask by calculating pairwise similarity, so as to guide context aggregation. Yuan et al. [17] introduced an object context pooling (OCP) module to explore the relationship between a pixel and the object neighborhood. DANet [18] designed spatial-wise and channel-wise self-attention mechanism to harvest the contextual information. To reduce the computational complexity of non-local module, Huang [19] proposed criss-cross attention module which only computes correlation matrix between each pixel and the corresponding row and column of this pixel. Zhu et al. [20] proposed to sample typical pixels of a feature map as the basis to compute the correlation matrix, reducing computational cost for the non-local module. Li et al. [21] introduced EM algorithms to build a low-rank weight matrix to improve the efficiency of the context aggregation. Although pairwise attention is useful, Li et al. [22, 23] found that long-range information in pairwise attention is usually dominated by features of large-scale patterns and inclined to oversmooth small-scale regions (e.g., boundaries and small objects). They proposed local distribution block to distribute global information adaptively over the feature map.

2.3 Efficient Segmentation

Many algorithms have been designed for efficient segmentation with reasonable accuracy, targeting real-time applications [24–29]. Some works dramatically reduced the resolution of the feature maps to achieve faster inference. For example, in ICNet [24], a cascade network was proposed with multi-scale inputs. Li et al. [25] used cross-level feature aggregation to boost accuracy on a light-weight backbone. Though effective, these methods had difficulty handling some small objects and boundaries of objects. Others tried to design light-weight networks to achieve efficiency. For example, BiSeNet [29] separated semantics and high-resolution details by introducing a spatial path and a semantic path. Different from these methods, we retain a high-resolution representation and encourage interactions between different levels of detail and abstraction.

3 Bidirectional Pyramid Networks

The architecture design of our BPNet is illustrated in Fig 1, including a preliminary convolution step, a bidirectional pyramid network for cross-scale information fusion, then followed by a parallel unary-pairwise attention module for capturing long-range dependency and thin structures.

3.1 Pyramid Architecture

As shown in Figure 1, our pyramid scheme of processing goes in two basic directions (blue arrows): one moves “up” in layers, from higher spatial resolution to lower resolution; the other moves “forward” in stages, maintaining spatial resolution while applying more convolution to extract information.

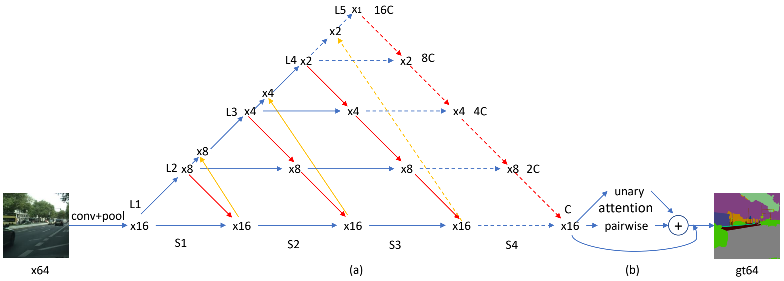

Fig. 1. Illustration of the architecture of the proposed BPNet. This framework consists of a pyramid network and a context aggregation module. (a) The pyramid network contains four stages, i.e., S1, S2, S3 and S4. Each stage is with a bidirectional information flow to boost communication between semantics and resolution, in which, top-down flow (red lines) propagate semantics to high-resolution features and bottom-up flow (yellow lines) passes high-resolution information to rich-semantic features. (b) Context aggregation module learns a pixel-wise unary attention for an emphasis on small patterns and a pairwise attention for long-range information dependency modeling.

A typical instantiation of the BPNet model consists of 4 or 5 layers (3 or 4 steps of subsampling), and 3 or 4 stages of convolution at the bottom layer (highest resolution). If the input resolution is x64 (with x an arbitrary integer), the feature resolution of the lowest layer is x16. At the second lowest layer, the feature resolution is x8, and it goes through 3 stages of convolution at the x8 resolution. As we move up the "pyramid", following common practice, we reduce the feature resolution by half at each step, and increase the number of channels by two (as illustrated by the channel numbers C, $2C$, $4C$, etc. with C an integer). A "bird-eye" view of our network resembles a pyramid, or triangle. In the figure, we show a pyramid of 4 layers (with solid arrows), and a pyramid of 5 layers (with the added stage L5, and the dotted arrows on the right side of the triangle).

Note that this basic architecture (without the cross-scale flows, in red and yellow) is already different from a typical UNet structure (or, for that matter, that of a typical feature pyramid in object detection). There is no clearly defined encoder and decoder. We not only pass information, laterally, from various steps of the subsampling (L1, L2, to L5, left side of the triangle, reducing resolution) to the corresponding steps of upsampling (right side of the triangle, increasing resolution); for each lateral link, we add a varying number of stages, or convolution stages. As we move up the layers, fewer processing steps are needed laterally, as the information has already been through a number of convolutions in the subsampling process.

Top-down information flow. With aforementioned basic architecture, we describe how we design information cross-scale flow in a systematic way. One key component is *top-down* information flow. As shown in Fig 1 (in red arrows), information flows "down" the pyramid at each processing step. For example, the features at L2 (at resolution x8 and after one step of convolution with subsampling) are fed down the hierarchy to be integrated with S1, which is one step of convolution at the lowest layer (maintaining feature resolution x16). Similarly, the features at L3 (at resolution x4 and after two steps of convolution with subsampling) are fed down the hierarchy to the layer below, to be integrated with the output of one lateral step from L2. Other top-down flows are designed similarly across the pyramid.

Empirically, the number of layers (in the resolution hierarchy) and the number of stages (processing steps at the lowest layer) tend to be the same, which results in a "perfect" triangle pattern. In all the models we use in this work, the triangles are "perfect", as they produce good results across board. It is worth noting that the number of layers and the number of stages do not have to be exactly the same. We have experimented with "skewed" triangles and they can be effective under certain circumstances (such as when the input resolution is high but we want a lighter weight model).

Bottom-up information flow. The top-down flows in our pyramid network enhances processing at the high resolutions (low layers) with more semantic and abstract information. However, the information flow does not have to be in only one direction. We can also add bottom-up information flows, as illustrated by yellow arrows in Fig. 1. For bottom-up flows, higher-resolution features (after top-down fusion) are fed upward to be integrated with lower-resolution features at the higher layers. This design completes our bidirectional pyramid network: information is free to flow laterally, upward, or downward, and they are fused at every step of the processing. In the ablation studies, we will show that both top-down flows and bottom-up flows are useful and improve accuracy without a heavy computational cost.

Feature fusion strategy. The bidirectional information flow in the pyramid network brings features with different characteristics together, where feature fusion plays a central role.

Typically, one of three feature fusion strategies is used in semantic segmentation: element-wise *addition*, element-wise *multiplication*, and *concatenation*. *Concatenation* is more flexible as it allows learned combination of the features at a later step, with a computational cost. On the other hand, element-wise *addition* and *multiplication* are more elementary operations: simple to compute, and do not increase the feature dimension.

We focus on how to effectively use the two elementary operations: *addition* and *multiplication*. Intuitively, *addition* can be viewed as an OR operation, combining individual signals from any of the two inputs; and *multiplication* can be viewed as an AND operation, selecting shared signals from both inputs. Using either of these two operations alone may not be sufficient for feature fusion. Therefore, we propose to use a combined fusion block *add-multiply-add* (**AMA**), a weighted combination of these operations, as described below and illustrated in Fig. 2.

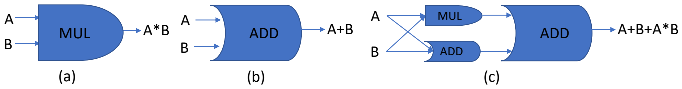

Fig. 2. Illustration of our feature fusion block *add-multiply-add* (**AMA**), which is more expressive than either **add** or **mul**, but does not increase feature dimension like **concat**.

Let A and B represent two input feature vectors at any fusion step of our pyramid network, A from the lower level (high-res, representing detail), and B from the higher level (low-res, representing context). Let A_i, B_i represent channel i of features A and B. Let $F(B)$ denote a transformation of B, including convolution, nonlinear activation, and also upsampling if there is a resolution mismatch between A and B. For element-wise addition **add**, our fusion function is $Y_i = A_i + F(B)_i$. Detail information is "summed" with context information directly. Intuitively, because context information in B has a low resolution, a direct sum tend to produce blurred boundaries. For element-wise multiplication **mul**, our fusion function is $Y_i = A_i \cdot F(B)_i$. Intuitively, this allows information both in A and B to reinforce each other, but unique signals in either A or B could be suppressed.

We find that either **add** or **mul** is not sufficient for feature fusion. Inspired by polynomial feature expansion, and related works such as Factorization Machine [30], we propose a simple yet effective feature fusion block called *add-multiply-add* (**AMA**):

$$Y_i = \alpha_i A_i + \beta_i F^a(B)_i + F^{ma}\left(A_i \cdot F^m(B)_i\right) \quad (1)$$

where F^a, F^m and F^{ma} are three transformations (convolutions) that bring the signals together, and α_i and β_i are learned weights.

In our ablation studies, we validate that the **AMA** fusion block is indeed more powerful and useful than either **add** or **mul**. As a comparison, we also explore a *concatenation* block **concat**, where the fusion function is $Y_i = conct(A_i, F(B)_i)$. Although *concatenation* is more expressive and incorporates *addition*, it cannot directly represent **multiplication**, and we find that it performs less well than the proposed **AMA** fusion, even with a higher computational cost.

3.2 Parallel Unary-Pairwise Attention

While our pyramid architecture is effective in modeling the fusion of small-scale and large-scale semantic cues, it operates locally and does not directly capture long-range dependency. Therefore, we feed the output of the pyramid model through a *Parallel Unary-Pairwise Attention* (**PUP**) module to further improve the effectiveness of the model.

We first use Asymmetric Pyramid Non-local Block (APNB) [20] to model long-range dependency through **pairwise attention**. APNB utilizes a pyramid sampling module into the nonlocal block to reduce computation and memory consumption. However, we find that pairwise attention context aggregation tends to be biased toward large-scale patterns and may harm small-scale patterns (also observed in [23]).

To mitigate the scale dilemma in APNB, we use a **unary attention** block parallel to the pairwise attention block, as shown in Fig. 3. Specifically, The input feature map first passes through a depth-wise 3×3 convolution, followed by a sigmoid function to transform them to "importance weights". Then we apply a learned importance matrix on the input feature map to generate a position-sensitive attention map. We conduct a simple element-wise addition to combine the position-sensitive attention map (from unary attention) and the long-range context-sensitive attention map (from pairwise attention), achieving effective context aggregation without losing signals for local regions. In our studies, we find that our PUP model, modeling unary and pairwise attention in parallel, performs better than pairwise attention alone.

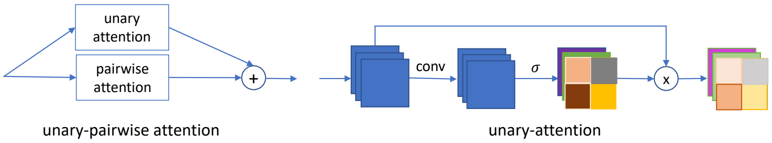

Fig. 3. Schematic illustration of unary-pairwise attention. This block receives a feature map from the pyramid network and outputs a feature map with global information aggregated and local signals retained.

3.3 Model Instantiations and Implementation Details

The model instance with 4 layers (L1 to L4) and 3 stages (i.e., S1 to S3), is called BPNet-S3. The model instance with 5 layers (L1 to L5) and 4 stages (i.e., S1 to S4), is called BPNet-S4.

We employ Kaiming initialization to initialize our proposed network. We use mini-batch stochastic gradient descent (SGD) with batch size 12, momentum 0.9 and weight decay $1e^{-4}$ during training. We apply the "poly" learning rate strategy in which the initial rate is multiplied by $(1 - \frac{iter}{maxiter})^{power}$ each iteration with power 0.9. The initial learning rate is set to $1e^{-2}$. We employ the mean subtraction, random horizontal flip and random scale on the input images to augment the dataset in training process. The scales contains {0.75, 1.0, 1.5, 1.75, 2.0}. Finally, we randomly crop the image into fix size for training. Implementation is done using TorchSeg [31].

4 Experiments

To evaluate the proposed BPNet models, we carry out experiments on Cityscapes dataset, CamVid, Pascal Context, and a medical image dataset (prostate MRI), with Cityscapes being our primary benchmark. Experimental results demonstrate that our BPNet achieves state-of-the-art performance on Cityscapes, particularly for real-time settings. Meanwhile, BPNet can compete with or outperform the state of the art on a number of other benchmarks, includign CamVid, Pascal context and prostate MRI. In this section, we first introduce the datasets, and then proceed to show our main results on Cityscapes, for both real-time and non-real-time settings. We then perform a series of ablation studies, also using the Cityscapes dataset, to valid various design choices in the BPNet. Finally, we report our results on Pascal Context, and a prostate MRI dataset.

4.1 Datasets and Evaluation Metrics

We use the standard *Mean IoU* (mean of class-wise intersection over union) for Cityscapes, Camvid, and Pascal Context. We use the Dice similarity score (DSC) for the prostate MRI dataset, as standard for that benchmark.

- Cityscapes [32] is tasked for urban segmentation, which contains 5,000 pixel-level finely annotated images captured from 50 different cities. Each image is with 1024 × 2048 resolution. The 5,000 finely annotated images are divided into 2,975/500/1,525 images for training, validation, and testing.
- CamVid [7] contains 701 road scenes with image resolution 720 × 960 extracted from driving videos, in which, 701 images are divided into training, validation and testing subsets with 367, 101 and 233 images, respectively. All images are pixel-wise annotated with 11 semantic classes.
- The PASCAL context dataset [8] includes 4, 998 scene images for training and 5, 105 images for testing with 59 semantic labels and 1 background label.
- PROMISE12-challenge dataset [10] is for MRI prostate segmentation, a widely used medical image segmentation benchmark. This dataset contains 50 labeled subjects where only prostate was annotated, and 30 extra subjects hose ground-truth label-maps are hidden from us.

4.2 Experiments on Cityscapes

We primarily carry out our experimental studies on the Cityscapes dataset. We first validate and compare the proposed models (our mediumweight network BPNet-S3) with state-of-the-art real-time semantic segmentation methods that focus on efficiency. We then compare our heavyweight network (BPNet-S4) with state-of-the-art models that focus on accuracy. Furthermore, we conduct ablation studies to explore the impact of the various key components in the BPNet.

Comparison with state-of-the-arts real-time segmentation methods We first consider a lightweight model, i.e., BPNet-S3-W32, the S3 model with the base number of channels C equal to 32. (If not specified, the mediumweight BPNet-S3 has the base number of channels equal to 48). Results of other state-of-the-art real-time semantic segmentation solutions on cityscapes validation and test set (with single-scale inference strategy) are summarized in Table 1. The lightweight model, BPNet-S3-W32, presents the highest mIoU with a fast inference speed. This shows that BPNet-S3-W32 is a good choice for efficiency-demanding segmentation tasks.

Table 1. Semantic segmentation results on Cityscapes. The GFLOPs is calculated on the input size 1024 × 2048. * means FPS tested by ourselves on RTX 2080 TI.

method	params.	GFLOPs	FPS	mIoU	
				val	test
ICNet [24]	7.7	-	37.5*	70.6	69.5
BiSeNet(Res18) [29]	13.4	104.3	41.7*	74.8	74.7
DFANet [25]	7.8	-	58.8*	-	71.3
BPNet-S3-W32	5.1	74.2	36.5	77.2	76.3

Comparison with state-of-the-art segmentation methods. We now show accuracy with our mediumweight and heavyweight models, i.e., BPNet-S3 and BPNet-S4. Results of other state-of-the-art semantic segmentation solutions on cityscapes validation set (with single-scale inference strategy) are summarized in Table 2. Among the approaches, DFN uses ResNet-101 [33] as backbone, Deeplabv3 [15] and PSPNet [14] both use Dilated-ResNet-101 as backbone and Deeplabv3+ [34] use stronger backbone. HRNet [9] utilizes imagenet to train a powerful pretrained model as the backbone for the segmentation tasks. The results show that the proposed BPNet-S3 can achieve similar performance with the DFN and DeepLabv3, but our model complexity is much lower (the number of parameters is about 5 times fewer). More importantly, the computational cost is about 10 times lower. In the meantime, our BPNet-S4 outperforms the DeepLabv3+, PSPNet and HRNetv2-W40. Again, BPNet-S4 has less parameters and needs much less computational resource. BPNet-S4 is also competitive to HRNetV2-W48, without using external data (e.g., Imagenet) for pretraining.

Table 2. Semantic segmentation results on Cityscapes validation with single-scale inference. The GFLOPs is calculated on the input size 1024 × 2048.

method	backbone	params.	GFLOPs	mIoU
DFN [33]	ResNet-101	90.2M	2221.0	78.5
PSPNet [14]	Dilated-ResNet-101	65.9M	2017.6	79.7
DeepLabv3 [15]	Dilated-ResNet-101	58.0M	1778.7	78.5
DeepLabv3+ [34]	Dilated-Xception-71	43.5M	1444.6	79.6
HRNetV2-W40 [9]	-	45.2M	493.2	80.2
HRNetV2-W48 [33]	-	65.9M	747.3	81.1
BPNet-S3	-	11.8M	227.1	78.3
BPNet-S4	-	40.5M	307.5	80.3

In addition, we also evaluate our models on the test set (with multi-scale inference strategy) by submitting inference results to the official evaluation server. We use train+val as training set to train our model and report the mIoU on the test set. From Table 3, We see that BPNet-S4 achieve better mIoU than most of the methods, and achieve competitive performance compared to HRNetv2-W48, again without pre-training, and lower computational complexity.

Table 3. Semantic segmentation results on Cityscapes test (train and train+val as training set, respectively) with multi-scale inference.

method	backbone	use pretraining	mIoU
with train set			
PSPNet [14]	Dilated-ResNet-101	yes	78.4
PSANet [35]	Dilated-ResNet-101	yes	78.6
HRNetV2-W48 [9]	-	yes	80.4
BPNet-S4	-	**no**	80.5
with train+val set			
BiSeNet [29]	ResNet-101	yes	78.9
DFN [33]	ResNet-101	yes	79.3
PSANet [35]	Dilated-ResNet-101	yes	80.1
PADNet [36]	Dilated-ResNet-101	yes	80.3
DenseASPP [37]	WDenseNet-161	yes	80.6
ANN [20]	ResNet-101	yes	81.3
OCNet [17]	ResNet-101	yes	81.7
OCR [38]	ResNet-101	yes	81.8
HRNetv2-W48 [9]	-	yes	81.6
BPNet-S4	-	**no**	81.9

Training Details. BPNet-S3-W32 and BPNet-S4 are trained with 240 epochs from scratch on Cityscapes, taking about 40 and 48 hours with 4 RTX 2080 TI. The training time is not excessive, comparable to that of SOTA methods with pretrained models (e.g., PSPNet, DeepLab V3+, HRNet and so on).

Comparing to Other Methods. Scale and fusion are central topics in computer vision. Our work draws inspiration from many state-of-the-art algorithms, such as HRNet [9] and GridNet [39]. We take HRNet as an example to explain the difference. (1) Feature fusion design is quite different. (2) BPNet is much more efficient than HRNet. (3) Pretraining is not necessary for BPNet to achieve good performance. More details about the difference are introduced in the supplementary material (Sec. 5.3).

Ablation studies To validate design choices in the BPNet, we conduct ablation experiments on the validation set of Cityscapes with different settings. All ablation studies are conducted on BPNet-S3.

Impact of bidirectional information flow. To investigate the effect of bidirectional information flow, we compare the following networks: (a) remove all top-down and bottom-up flows in the pyramid (denoted as 'BaseNet'); (b) add only top-down flow in the pyramid network (denoted as '+Top-Down'); (c) add only bottom-up flow (denoted as '+Bottom-Up') and (d) with both top-down and bottom-up flows (denoted as '+bidirectional').

Table 4. Investigation of bidirectional information flow.

method	fusion strategy	mIoU	*Δ*mIoU
BaseNet	-	73.7	-
+Top-Down	AMA	75.0	+1.3
+Bottom-Up	AMA	74.6	+0.9
+Bidirectional	AMA	76.1	+2.4
+Bidirectional	add	75.4	-0.7
+Bidirectional	mul	75.2	-0.9
+Bidirectional	concat	75.5	-0.6

In Table 4, both "+Top-Down" and "+Bottom-Up" can boost the base network to achieve better performance. Compared to bottom-up information flow, top-down information flow is more beneficial which means providing context to high-resolution processing is more important. With both the top-down and bottom-up links, the network can enjoy even more performance gain, demonstrating the merit of having information flow at every step of the processing, in both upward and downward directions.

Feature fusion strategy. As mentioned in Section 3, three popular feature fusion strategies are **add**, **mul** and **contact**. Our ablation studies focus on comparing these fusion approaches with our proposed **AMA** feature fusion in the bidirectional setting. Table 4 indicates that the proposed **AMA** works best for feature fusion, which outperforms the widely used **concat** as well as **add**.

Impact of parallel unary-pairwise attention. To validate the impact of our parallel unary-pairwise attention (PUP) mechanism for capturing both long-range dependency and thin structure, we conduct experiments with four different designs of attention mechanisms applied on the output of the pyramid network, respectively: (a) pairwise attention only (using APNB [20]) for context aggregation; (b) sequential integration of unary and pairwise attention, unary first, pairwise second; (c) sequential integration of unary and pairwise attention, pairwise first, unary second; (d) the proposed parallel unary-pairwise attention mechanism (PUP). We find that all four attention mechanisms are useful in improving accuracy, with parallel unary-pairwise attention mechanism (PUP) performs the best, significantly better than the two sequential mechanism.

4.3 CamVid

In this subsection, we further validate our lightweight models on the CamVid dataset. The results on test set are listed in Tab. 6. BPNet-S3 can achieve a mIoU

Table 5. Investigation of pairwise attention, unary attention, and their combinations.

method	mIoU	Δ mIoU
pyramid (baseline)	76.1	-
+pairwise	77.5	+1.4
+unary+pairwise (sequential)	77.8	+1.7
+pairwise+unary (sequential)	78.1	+2.0
+PUP (parallel unary-pairwise)	78.3	+2.2

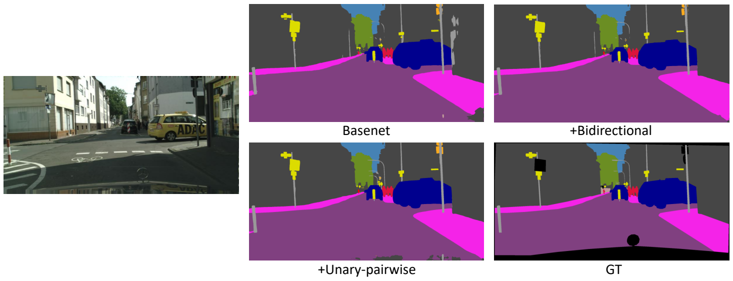

Fig. 4. An visual example of using the proposed modules in the BPNet. The bidirectional model significantly improves over the basenet, by removing wrong predictions on the building (right side), and improving upon one of the three lamps (upper middle). Adding attention, we see improvements over all three lamps with thin structures.

as high as 73.8. With a much smaller number of parameters (5.1M), BPNet-S3-W32 can achieve higher inference speed with competitive accuracy.

Table 6. Semantic segmentation results on CamVid test. Flops computed on 720 × 960.

Dataset	mIoU	params	FLOPs	FPS
SegNet [40]	55.6	29.5	-	6.6*
ICNet [24]	67.1	7.7	-	41.9*
BiSeNet(Res18) [29]	68.7	13.4	34.5	-
BPNet-S3	73.8	11.8	56.9	34
BPNet-S3-W32	69.4	5.1	24.5	52

4.4 PASCAL Context

We keep the same data augmentation and learning rate policy in training are as Cityscapes. We set the initial learning rate to $4e^{-3}$ and weight decay to $1e^{-4}$ [41, 42]. During inference, we follow the standard procedure as suggested in [41, 42]. The comparison of our method with state-of-the-art methods is presented in Table 7. Our network performs competitively to previous state-of-the-arts without tuning of the hyper-parameters (same to those used for Cityscapes).

Table 7. Semantic segmentation results on PASCAL-context, evaluated on 59 classes.

Dataset	backbone	use pretraining	mIoU (59 classes)
PSPNet [14]	Dilated-ResNet-101	yes	47.8
UNet++ [5]	ResNet-101	yes	47.7
EncNet [42]	ResNet-152	yes	52.6
HRNetv2-W48 [33]	-	yes	54.0
BPNet-S4	-	**no**	52.7

4.5 Medical Image Data: Prostate Segmentation

Without the need for pre-training, BPNet has the potential to be useful for domains other than natural images. To show the versatility of BPNet, we conduct additional experiments on the PROMISE12-challenge dataset [10], a popular MRI segmentation benchmark. The detailed comparison is provided in supplementary material due to page limit. Without specific adaptation, we can achieve a high DSC (91.1) in average based on five-fold cross validation, surpassing many existing 3D medical image segmentation algorithms with much less training and inference time. BPNet is competitive with nnUNet, a state-of-the-art 3D convolution model with the highest reported accuracy, but is 45x more efficient. These experimental results indicate that our models may find applications in many domains that need semantic segmentation.

5 Conclusions

We have presented our bidirectional pyramid network for semantic segmentation. Starting from scratch without standard backbones or pre-training, we designed a family of semantic segmentation models with several simple and yet effective components, i.e., pyramid network with top-down and bottom-up information flow, to enhance information interaction between large-scale contexts and small-scale details. We also propose a parallel unary-pairwise attention for context aggregation to help with long-range dependency and thin structure. Competitive results are produced on standard benchmarks and the proposed components are validated to be effective. With efficiency, and without pre-training, we believe our models have the potential to be used for many applications and have room for further improvements.

References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
2. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241

3. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), IEEE (2016) 565–571
4. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 (2018)
5. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer (2018) 3–11
6. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1925–1934
7. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: European conference on computer vision, Springer (2008) 44–57
8. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 891–898
9. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514 (2019)
10. Litjens, G., et al.: Evaluation of prostate segmentation algorithms for mri: the promise12 challenge. MedIA **18** (2014) 359–373
11. Pinheiro, P.O., Lin, T.Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: European Conference on Computer Vision, Springer (2016) 75–91
12. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 8759–8768
13. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. arXiv preprint arXiv:1911.09070 (2019)
14. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2881–2890
15. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40** (2017) 834–848
16. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7794–7803
17. Yuan, Y., Wang, J.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)
18. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3146–3154
19. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 603–612

20. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 593–602
21. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9167–9176
22. Li, X., Zhao, H., Han, L., Tong, Y., Yang, K.: Gff: Gated fully fusion for semantic segmentation. arXiv preprint arXiv:1904.01803 (2019)
23. Li, X., Zhang, L., You, A., Yang, M., Yang, K., Tong, Y.: Global aggregation then local distribution in fully convolutional networks. arXiv preprint arXiv:1909.07229 (2019)
24. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 405–420
25. Li, H., Xiong, P., Fan, H., Sun, J.: Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 9522–9531
26. Li, X., Zhou, Y., Pan, Z., Feng, J.: Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In: Proceedings of the IEEE Conference on computer vision and pattern recognition. (2019) 9145–9153
27. Orsic, M., Kreso, I., Bevandic, P., Segvic, S.: In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2019) 12607–12616
28. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 1314–1324
29. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). (2018) 325–341
30. Rendle, S.: Factorization machines. In: 2010 IEEE International Conference on Data Mining, IEEE (2010) 995–1000
31. Yu, C.: Torchseg. `https://github.com/ycszen/TorchSeg` (2019)
32. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 3213–3223
33. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 1857–1866
34. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). (2018) 801–818
35. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 267–283
36. Xu, D., Ouyang, W., Wang, X., Sebe, N.: Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 675–684

37. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 3684–3692
38. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. arXiv preprint arXiv:1909.11065 (2019)
39. Fourure, D., Emonet, R., Fromont, E., Muselet, D., Tremeau, A., Wolf, C.: Residual conv-deconv grid network for semantic segmentation. arXiv preprint arXiv:1707.07958 (2017)
40. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39** (2017) 2481–2495
41. Ding, H., Jiang, X., Shuai, B., Qun Liu, A., Wang, G.: Context contrasted feature and gated multi-scale aggregation for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2393–2402
42. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2018) 7151–7160