6th Int. Symposium on Artificial Intelligence and Robotics (ISAIR). Virtual, August 2021

BASNet: Improving semantic segmentation via boundary-assistant symmetrical network

Yong Qiang[a], Quan Zhou[a], Huimin Shi[a], Xin Jin[b], Weihua Ou[c] and Longin Jan Latecki[d]

[a]College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China;
[b]Department of Computer Science and Technology, Beijing Electronic Science and Technology Institute, Beijing, China;
[c]School of Big Data and Computer Science, Guizhou Normal University, Guiyang, China;
[d]Department of Computer and Information Sciences, Temple University, Philadelphia, USA

ABSTRACT

Recently, boundary information has gained more attention in improving the performance of semantic segmentation. This paper presents a novel symmetrical network, called BASNet, which contains four components: the pre-trained ResNet-101 backbone, semantic segmentation branch (SSB), boundary detection branch (BDB), and aggregation module (AM). More specifically, our BDB only focuses on processing boundary-related information using a series of spatial attention blocks (SABs). On the other hand, a set of global attention blocks (GABs) are used in SSB to further capture more accurate object boundary information and semantic information. Finally, the outputs of SSB and BDB are fed into AM, which merges the features from SSB and BDB to boost performance. The exhaustive experimental results show that our method not only predicts the boundaries of objects more accurately, but also improves the performance of semantic segmentation.

Keywords: Semantic segmentation, Boundary detection, Attention Block, Symmetrical network, ResNet-101 backbone

1. INTRODUCTION

Recent studies[1–4] have demonstrated that deep neural networks (DNNs) can learn more powerful mid-level image representations in various vision-related tasks, e.g., image classification,[1] object detection,[5] image semantic segmentation,[6,7] content-based image retrieval,[4] etc. In these tasks, image semantic segmentation is a fundamental task in the field of computer vision, which plays an important role in many real-world applications, such as autonomous driving,[8] robotics,[9,10] and medical segmentation.[11,12] In recent years, convolutional neural networks (CNNs) have achieved remarkable progress over all segmentation benchmarks. A mainstream architecture is to convert the fully connected layer into a fully convolutional layer so that the CNN architecture used for image classification can be adapted to the task of semantic segmentation.[13–16] However, FCN-based networks have following disadvantage for dense estimation problem: The spatial resolution of the output feature map is greatly reduced due to the downsampling operations (e.g., pooling, convolution stride, etc.). This motivates the generation of new CNN architectures[17,18] to restore the spatial resolution of the network output. Yet these methods ignore the boundary information of objects, resulting in rough segmentation of object shapes and boundaries, which influences the performance of semantic segmentation.

In order to relieve above problems, some methods[6,7,19,20] have been proposed to assist semantic segmentation using the results of boundary detection. For example, GSCNN[6] proposes a two-stream network by explicitly merging shape information into the feature map. In addition, a dual task loss is adopted to optimize semantic mask and boundary prediction, synchronously. DFN[7] designs a boundary network with deep supervision to refine the semantic boundary of prediction. However, these methods still have the following limitations: (1) When using boundary clues to assist semantic segmentation,[6,20] the extracted boundary features may contain

Corresponding author: Quan Zhou, quan.zhou@njupt.edu.cn. This work is partly supported by NSFC (No. 61876093, 61801242, 61701252, 61671253), and NSFJS (No. BK20181393).

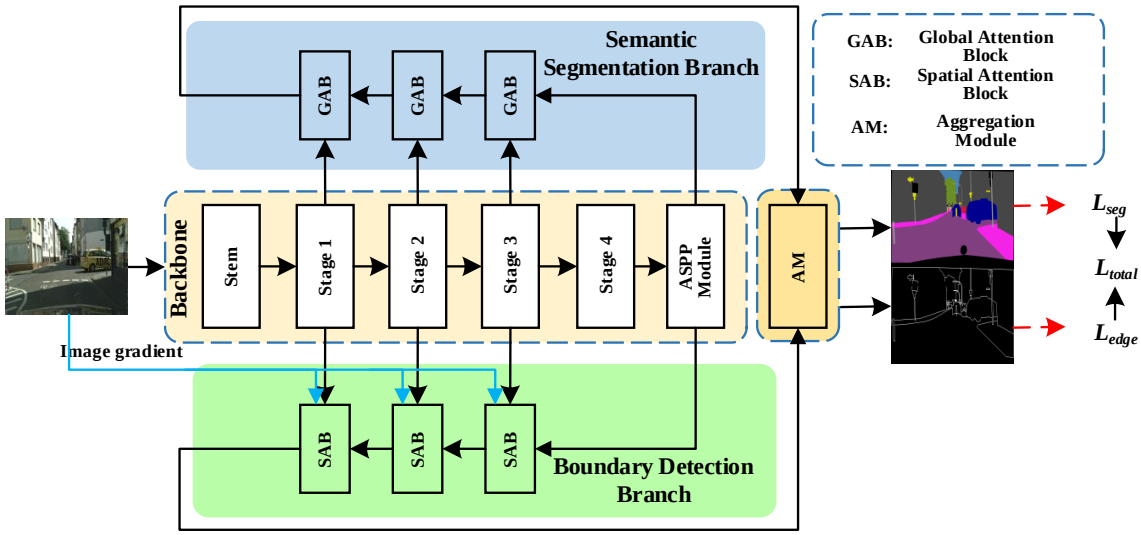

Figure 1. Overall architecture of our BASNet. The BASNet includes four components: backbone network, semantic segmentation branch (SSB), boundary detection branch (BDB), and aggregation module (AM). The backbone is combined by a pre-trained ResNet-101[1] and an ASPP[14] module. The blue line means that the input image passes through a Canny[21] operation and then concatenated with the input image, and finally sent to BDB. (Best viewed in color)

non-boundary parts (e.g., background, object inside parts, etc.) that are not beneficial to accurately identifying the boundaries of objects. Moreover, some methods[7,19] lose part of the boundary due to the limitation of receptive field, where the non-boundary features eventually affect the performance of semantic segmentation; (2) Some previous methods[6,7] often ignore some important feature information in the backbone network stages, especially spatial information. Yet semantic features from high-level stages and boundary features from low-level stages are equally important for semantic segmentation.

To deal with these shortcomings, this paper designs a symmetrical network, called BASNet, using object boundary to improve the performance of semantic segmentation. As shown in Fig. 1, our BASNet mainly consists of three parts: semantic segmentation branch (SSB), boundary detection branch (BDB), and aggregation module (AM). In BDB, a series of SABs are used to restore feature resolution, where each SAB encodes object boundary using spatial attention to enhance boundary features. Similarly, our SSB uses a set of GABs to recover the feature resolution. Compared with SAB, the GAB also uses channel attention to encode the importance of each channel, which provide useful channel information. At the end, the outputs of SSB and BDB are fed into AM to output final results of semantic segmentation and boundary detection. In summary, this paper has three contributions as follows: (1) We propose a novel symmetrical network to solve semantic segmentation task using boundary information and achieve new state-of-the-art results on Cityscapes[22] dataset. (2) In the SSB, we present a series of GABs to improve the accuracy of object localization and object classification by aggregating channel information and spatial information of each stage. (3) In the BDB, we introduce a set of SABs to correctly identify the contour of objects using the spatial information of each stage and the image gradient information.

The remainder of this paper is organized as follows. After a brief introduction of related work in Sec. 2, we elaborate on the details of our BASNet in Sec. 3. Experimental results are given in Sec. 4, and Sec. 5 provides conclusion remarks and future work.

2. RELATED WORK

2.1 Semantic segmentation

Full convolutional network (FCN[13]) based methods[13–16] has made great progress in semantic segmentation. In,[14]last two downsample layers are removed to obtain dense prediction and dilated convolution operations are

employed to enlarge the receptive field. SegNet,[17] DenseNet[18] adopt encoder-decoder structures to sequentially recover spatial resolution of the network ouput.

Recently, some works[7,23,24] attempt to improve the performance of semantic segmentation through attention mechanism. In,[7] the channel attention is used to fuse the channel information of each stages. CCNet[23] decomposes a standard non-local module into two sequenced cross attention blocks. DANet[24] enable a single feature for any pixel to interact with all other pixels. Inspired by these approaches, our BASNet introduces a global attention block (GAB) to aggregate the important feature information of each stage.

2.2 Boundary for segmentation

Recently, some methods[6,7,19,25,26] exploit the boundary information to improve the segmentation. BFP[19] introduced a boundary aware feature propagation (BFP) module to harvest and propagate the local features within their regions isolated by the learned boundaries in the UAG-structured image. GSCNN[6] exploits the duality between the segmentation predictions and the boundary predictions with a two-branch mechanism and a regularizer. In,[7] a boundary network with deep supervision is proposed to refine the semantic boundary of prediction. However, these methods are not beneficial to accurately identifying the boundaries of objects due to the limitations of extracting boundary features and improper processing of non-boundary parts.

In this paper, a series of SABs in BDB are introduced to focus on processing boundary-related information, where each SAB enhances the boundary features through spatial attention, and suppresses the response of non-boundary pixels at the same time using image gradient information. In addition, we also propose a joint loss function that refines both semantic segmentation result and boundary prediction result.

3. OUR METHOD

3.1 Network architecture

The overall architecture of our BASNet is depicted in Fig. 1. In order to obtain high-quality semantic segmentation outputs, we adopt ResNet-101,[1] pre-trained on ImageNet,[27] and add the Atrous Spatial Pyramid Pooling (ASPP[14]) module, as the backbone to abstract deep features. Following 6,7,24,28, we employ holding-resolution version of ResNet-101 using dilation convolutions, where all the feature maps in the last three stages have the same spatial size. It retains more details without adding extra parameters. Moreover, some methods[14,15] have proved that the ASPP module is an important part of state-of-the-art semantic segmentation, where the feature from the dilated residual network is fed into ASPP to generate the output feature map of the backbone.

After we gather features from stage1, stage2, stage3 and the ASPP module, our BDB uses a series of SABs to recover the feature resolution step by step, where each SAB can enhance the boundary features and suppresses the non-boundary features using spatial attention and image gradient. At the same time, similar to BDB, a set of GABs in SSB are used to recover the feature resolution. Furthermore, each GAB extracts semantic information and spatial information of different stages by channel attention and spatial attention. As done for BDB and SSB, their outputs are fed into AM to fuse the information of these two branches. Finally,

the outputs of AM, which have predicted channel-wise semantic maps and boundary maps, later receive their supervisions from the semantic ground truth maps and the boundary ground truth maps (Following 6), respectively.

3.2 Semantic segmentation branch

As shown in Fig. 2, our GAB has two inputs: the feature map $\mathbb{F}_l$ comes from low-level stage, and the feature map $\mathbb{F}_h$ comes from high-level stage. Note the Sec. 3.1 has shown that the spatial resolution of stage2 $\sim$ 4 and ASPP module remains unchanged, while the spatial resolution of stage1 is twice the above. Therefore, the spatial resolution of $\mathbb{F}_l \in \mathbb{R}^{C\times H\times W}$ is twice $\mathbb{F}_h \in \mathbb{R}^{2C\times H/2\times W/2}$ when $\mathbb{F}_l$ comes from stage1. The spatial resolution of $\mathbb{F}_l$ is the same as $\mathbb{F}_h$ when $\mathbb{F}_l$ comes from other stages, i.e. $\mathbb{F}_l, \mathbb{F}_h \in \mathbb{R}^{C\times H\times W}$.

Firstly, considering that low-level stages contain rich spatial information due to the larger spatial resolution, it is beneficial to improve the accuracy of object localization. Consequently, the feature map $\mathbb{F}_l$ first undergoes an 1×1 convolution to compress the channel of $\mathbb{F}_l$ from C to 1, and then passed through a sigmoid function, resulting

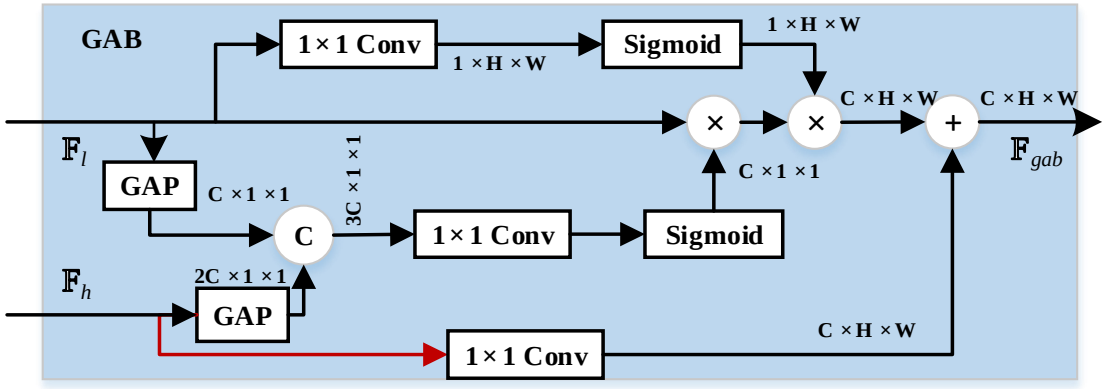

Figure 2. Our Global Attention Block (GAB). The red line represents the deconvolution with a kernel size of 3 and a stride of 2 when the low-level feature map $\mathbb{F}_l$ comes from stage1. Otherwise, the red line can not change the size of feature maps, just a path of information passing. (Best viewed in color)

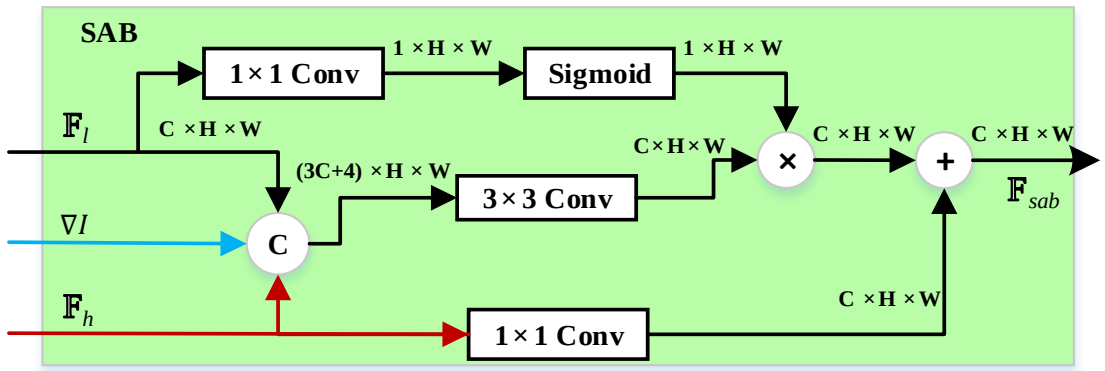

Figure 3. Our Spatial Attention Block (SAB). The blue line is detailed in the caption of Fig. 1. The red line represents the deconvolution with a kernel size of 3 and a stride of 2 when the low-level feature map $\mathbb{F}_l$ comes from stage1. Otherwise, the red line can not change the size of feature maps, just a path of information passing. (Best viewed in color)

in the spatial attention map $\boldsymbol{S} \in \mathbb{R}^{1\times H\times W}$. In addition, to utilize channel information of each stage, we perform global average pooling on $\mathbb{F}_l$ and $\mathbb{F}_h$ respectively, and then concatenate them. Thereafter, the concatenated feature map $\boldsymbol{C}' \in \mathbb{R}^{3C\times 1\times 1}$ is fed into an 1×1 convolution, which compresses the channel dimension from $3C$ to 1, and then passed through a sigmoid function, generating a channel attention map $\boldsymbol{C} \in \mathbb{R}^{C\times 1\times 1}$.

Secondly, we element-wise multiply the feature map $\mathbb{F}_l$ and $\boldsymbol{C}$ to get our weighted feature map $\mathbb{F}'_w \in \mathbb{R}^{C\times H\times W}$. Thereafter, the final weighted feature map $\mathbb{F}_w \in \mathbb{R}^{C\times H\times W}$ is obtained by multiplying $\mathbb{F}'_w$ and $\boldsymbol{S}$. In this way, the spatial attention map $\boldsymbol{S}$ obtained by low-level features encodes the importance of each pixel in $\mathbb{F}_w$, hence accurately identifies the object shapes and position. Furthermore, the channel attention generated by low-level and high-level features highlight the importance of each channel, which provides important feature information.

Finally, the $\mathbb{F}_h$ first pass through a red line (The detailed description is shown in the caption of Fig. 2 and Sec. 3.1), and then undergoes an 1×1 convolution to generate a up-sampling feature map $\mathbb{F}_{up} \in \mathbb{R}^{C\times H\times W}$. Thereafter, we add the feature map $\mathbb{F}_{up}$ and $\mathbb{F}_w$ to output the GAB's output feature map $\mathbb{F}_{gab} \in \mathbb{R}^{C\times H\times W}$.

3.3 Boundary detection branch

As illustrated in Fig. 3, our SAB has three inputs: low-level feature map $\mathbb{F}_l \in \mathbb{R}^{C\times H\times W}$, high-level feature map $\mathbb{F}_h$ (the dimension are the same as those described Sec. 3.2) and image gradient map $\nabla I \in \mathbb{R}^{4\times H\times W}$.

Still taking into account the fact that low-level features contain rich boundary information due to the larger spatial resolution. Firstly, the feature map $\mathbb{F}_l$ is fed into an 1×1 convolution to compress the channel from C

Figure 4. The boundary prediction results of three SABs on Cityscapes val set. From left to right are input images, ground truth, + The first SAB, + The second SAB and + The third SAB.

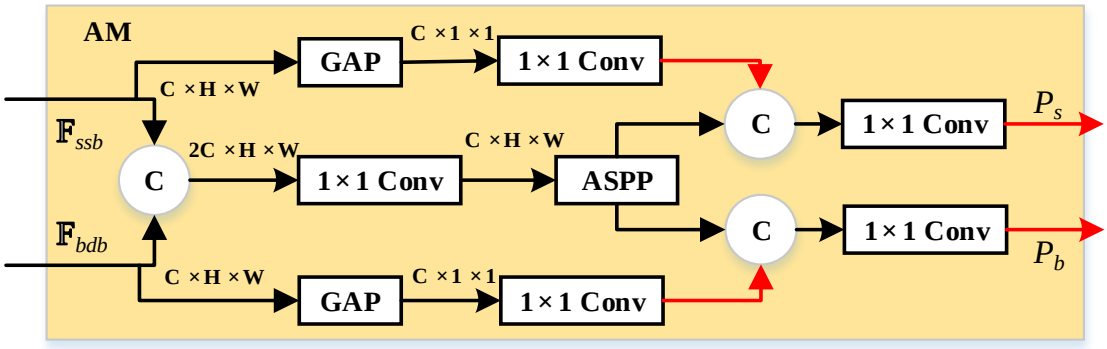

Figure 5. Our Aggregation Module (AM). The red line represents the bilinear upsampling operation. (Best viewed in color)

to 1, and then passed though a sigmoid function, generating a spatial attention map $\boldsymbol{S} \in \mathbb{R}^{1\times H\times W}$.

Secondly, we concatenate $\mathbb{F}_l$, $\mathbb{F}_h$ and ∇I together, where the feature information of each stages and image gradient information are used to assist in refining boundaries of objects. Besides, in order to perform concatenate operation, the resolution of $\mathbb{F}_h$ is up-sampled 2 times to match the resolution of $\mathbb{F}_l$. Thereafter, the concatenated feature map $\mathbb{F}''_w \in \mathbb{R}^{(3C+4)\times H\times W}$ undergoes a 3×3 convolution to output feature map $\mathbb{F}'_w \in \mathbb{R}^{C\times H\times W}$, which aggregates these three input features across channels.

Thirdly, we element-wise multiply the feature map $\mathbb{F}'_w$ and $\boldsymbol{S}$ to output a final weight feature map $\mathbb{F}_w \in \mathbb{R}^{C\times H\times W}$. On the one hand, the feature map $\boldsymbol{S}$ encodes the importance of each pixel of $\mathbb{F}'_w$ to enhance the boundary features. On the other hand, the feature map $\mathbb{F}'_w$ with image gradient information is helpful to suppress the interference of non-boundary information. Finally, we add the feature map $\mathbb{F}_h$ and $\mathbb{F}_w$ to output a SAB's output feature map $\mathbb{F}_{sab} \in \mathbb{R}^{C\times H\times W}$ (As same as GAB, see the last paragraph of Sec. 3.2 for details).

In addition, to further validate the effectiveness of our SAB, we take the following an experiment: The output feature map of the SAB first undergoes a bilinear upsampling to restore to the same resolution as the input image, and then passed through an 1×1 convolution to compress the channel to 1. Finally, we follow 6 to output a boundary prediction result. As shown in Fig. 4, whether it is the boundary of "person" and "trafficsign" in the first example or the boundary of "bus" and "sidewalk" in the second example (marked in the yellow circle), as each SAB increases, their boundaries change from foggy to clearer. The experimental results also show that our BDB can handle the interference of non-boundary parts well using three SABs and the assistance of image gradient information.

3.4 Aggregation module

In this section, we specially introduce an aggregation module (AM) to merge the region-based feature $\mathbb{F}_{ssb}$ from SSB and the boundary-based feature $\mathbb{F}_{bdb}$ to output a refined segmentation prediction result $\boldsymbol{P}_s$ and a refined boundary prediction result $\boldsymbol{P}_b$. The detailed structure of AM is illustrated in the Fig. 5.

Table 1. Comparison results of the proposed modules on Cityscapes val set without data augmentation. We use ResNet50/101 as the backbone followed by an ASPP.

Method	SSB	BDB	AM	mIoU(%)
ResNet50+ASPP				74.21
ResNet50+ASPP	✓			75.32
ResNet50+ASPP	✓	✓		75.71
ResNet50+ASPP	✓	✓	✓	77.14
ResNet101+ASPP				75.01
ResNet101+ASPP	✓			76.94
ResNet101+ASPP	✓	✓		77.73
ResNet101+ASPP	✓	✓	✓	79.20

As can be seen, our AM has two inputs: the feature map $\mathbb{F}_{ssb}$ from SSB and the feature map $\mathbb{F}_{bdb}$ from BDB. Firstly, we concatenate them and then pass through an 1×1 convolution. Thereafter, the aggregated feature map is fed into an ASPP module to output a feature map $\mathbb{F}_{cat} \in \mathbb{R}^{\frac{C}{4} \times H \times W}$ with multi-scale context information. Secondly, a global average pooling is performed on $\mathbb{F}_{ssb}$ and then undergoes an 1×1 convolution to output a feature map $\mathbb{F}'_{ssb} \in \mathbb{R}^{\frac{C}{8} \times 1 \times 1}$ with long-ranged dependencies. Thirdly, the feature map $\mathbb{F}'_{ssb}$ first pass through a bilinear upsampling to recover the spatial size, and then concatenate with $\mathbb{F}_{cat}$. Finally, the concatenated feature map is fed into 1×1 convolution to fulfills projection from feature space to semantic space, and then perform a bilinear upsampling to output a final segmentation prediction result $\boldsymbol{P}_s$. In addition, we can see from the Fig. 5 that our AM is a symmetrical architecture, where the output method of $\boldsymbol{P}_b$ and $\boldsymbol{P}_s$ are the same. Therefore, the boundary prediction result $\boldsymbol{P}_b$ will not be explained in detail.

4. EXPERIMENTS

In this section, we provide an extensive evaluation of each component of our method BASNet on the Cityscapes[22] dataset. In order to show the performance of our method, we choose GSCNN[6] , DFN,[7] and other state-of-the-art neural networks for semantic segmentation as benchmarks. We also provide quantitative results and qualitative results of our method.

4.1 Dataset

Cityscapes[22] dataset includes 30 object categories selected from 5 videos. This dataset has 5,000 high quality finely annotated images and 20,000 coarsely annotated images, where each image is shot on streets and of high-resolution (2048×1024). Following 6,7, only 19 classes are used for evaluation, and we only employ images with fine pixel-level annotations, resulting in 2,975 training, 500 validation and 1,525 testing images.

4.2 Implementation details and Evaluation metric

Motivated by 6, our training objective has two supervisions: The first one is the cross-entropy loss function L_{seg}, while the second one is the binary cross-entropy loss function L_{edge}, and they are all after the final output of the system. Therefore, our loss function is composed of two losses as:

$$L_{total} = L_{seg} + \lambda \times L_{edge} \tag{1}$$

where λ is a non-negative parameter that leverages the trade off between two losses. In our experiment, the balance parameter λ is set to 0.05, empirically.

Our BASNet is implemented in the hardware platform of the deep learning server with RTX 2080Ti GPU. The software coding is based on an open source repository for semantic segmentation using Pytorch. For the Cityscapes[22] dataset, our BASNet is trained using the stochastic gradient descent algorithm[29] with batch size of 8, where the initial learning rate is set to 1×10^{-2}, together with momentum and weight decay, which are set to 0.9 and 10^{-4}, respectively. Inspired by 15, we use the "poly" learning rate policy where the learning rate

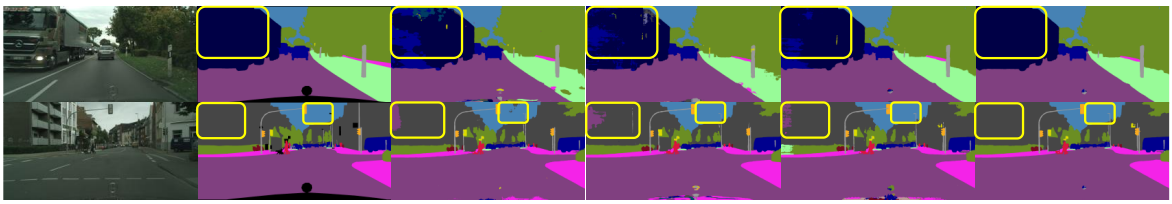

Figure 6. The segmentation prediction results of each module on Cityscapes validation set. From left to right are input images, ground truth, baseline, (a) baseline+SSB, (b) baseline+SSB+BDB and (c) baseline+SSB+BDB+AM. ResNet-101+ASPP is used as baseline.

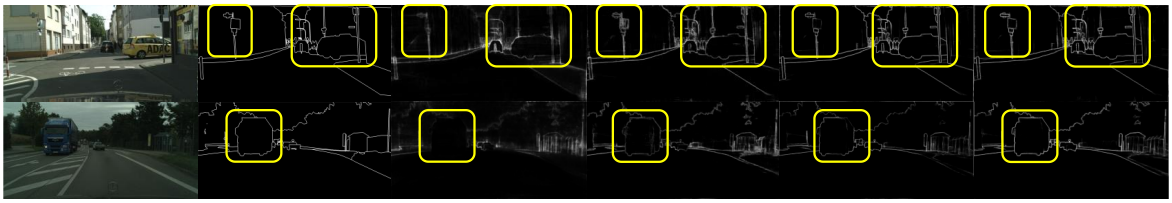

Figure 7. The boundary prediction results of each module on Cityscapes validation set. From left to right are input images, ground truth, baseline, (a) baseline+SSB, (b) baseline+SSB+BDB and (c) baseline+SSB+BDB+AM. ResNet-101+ASPP is used as baseline.

is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$ with $power = 0.9$. To augment training data, we first randomly crop out high-resolution patches with resolution of 480×480 from original images as the inputs.

For evaluation metric, we use two quantitative measures to evaluate the performance of our approach on Cityscaps[22] dataset: 1) We use the average intersection union (mIoU) in all categories to evaluate the accuracy of segmentation. 2) We also report F-score proposed in 6, 26, 30 by calculating along the boundary of the segmentation mask given a small slack in the distance to show the high-quality boundaries of segmentation mask. In our experiments, following 6, we measure the boundary F-score using thresholds 0.00088, 0.001875 and 0.00375 corresponding to 3, 5 and 9 pixels, respectively. Since boundaries are not provided for the Cityscapes testing set, we use the Cityscapes val set to compute F-scores as a metric for boundary accuracy.

4.3 Ablative studies

In Table 1, we evaluate the effectiveness of each module of our method on the Cityscapes val set. Specifically, we use ResNet[1] + ASPP[14] for the backbone architectures. We see from the table that based on the architecture of ResNet50 + ASPP, the addition of SSB and BDB improves performance by 1.1% and 0.4%. The addition of AM improves performance by 1.4%. On the basis of the architecture of ResNet-101 + ASPP, adding SSB, BDB and AM in turn has brought about 1.9%, 0.8% and 1.5% performance improvement. These experimental results well prove the effectiveness of each module in our method on the Cityscapes[22] val set.

As shown in Fig. 6 and 7, we also visualize the segmentation results and boundary results of each module to evaluate the performance of each component. In Fig. 6, we see from the first example that part of the "bus" category (marked in the yellow circle) is incorrectly classified as "car" when no modules are added. With the increase of each module, the incorrectly segmented part of the "bus" category is gradually corrected. Besides, in the second example of the Fig. 7, we see that boundary of the "bus" category (marked in the yellow circle) is very coarse when no modules are added. After SSB and BDB are added, the boundaries of this "bus" gradually become clearer, but there are still some parts that are still a bit blurred. So when the last AM is added, we see that the boundary of the bus category is completely identified.

Table 2. Evaluation results of our BASNet and other methods on Cityscapes testing set. "$*$" represents the data augmentation, which includes random scaling, random flip and multi-scale.

Method	Backbone	mIoU(%)
PSPNet[16]	Dilated-ResNet101	78.4
Deeplabv3+[14]	ResNet101	78.8
AAF[31]	ResNet101	79.1
DFN[7]	ResNet101	79.3
TKCN[32]	ResNet101	79.5
DenseASPP[33]	DenseNet161	80.6
GSCNN[6]	ResNet101	80.8
CCNet[23]	ResNet101	81.4
Ours	ResNet101	79.2
Ours*	ResNet101	**81.6**

Table 3. Comparison vs state-of-the-art baselines at different thresholds in terms of F-score on the Cityscapes val set.

Width	Method	F-score
3px	Deeplabv3+[14]	69.7
	GSCNN[6]	73.6
	Ours	**74.8**
5px	Deeplabv3+[14]	74.7
	GSCNN[6]	77.6
	Ours	**78.4**
9px	Deeplabv3+[14]	78.7
	GSCNN[6]	80.7
	Ours	**81.1**

4.4 Evaluation results

Table 2 shows the results of our method compared with other state-of-the-art methods on Cityscapes testing set. Based on ResNet101+ASPP, our method ultimately achieves 81.6% mIoU on the CityScapes testing set. In addition, Fig. 8 shows partial visual comparison of our method and some state-of-the-art methods on Cityscapes val set. We can see from the yellow circle that our method is better than other state-of-the-art methods in semantic segmentation. In Table 3, we also compare the performance of our method against other state-of-the-art methods in terms of boundary accuracy at different thresholds. Experimental results show that our method is 5% higher than Deeplabv3+[14] and 1% higher than GSCNN[6] in the strictest regime(width = 3px). In addition, Fig. 9 shows partial boundary prediction result comparison on Cityscapes val set. It can be observed from the yellow circle that our method accurately recognizes object boundary. All the results show that our method has achieved very good performance in semantic segmentation and boundary detection.

5. CONCLUSIONS

In this paper, we propose a new symmetrical network (BASNet) for semantic segmentation, which improves semantic segmentation using boundary information. The experimental results show that this is an efficient architecture that produces more accurate prediction around the object boundary and significantly improves the performance of semantic segmentation, and our method also achieves the better performance on the Cityscapes dataset. Future work includes light-weighting our network without loss of accuracy, so that our method can be applied to embedded devices in real life.

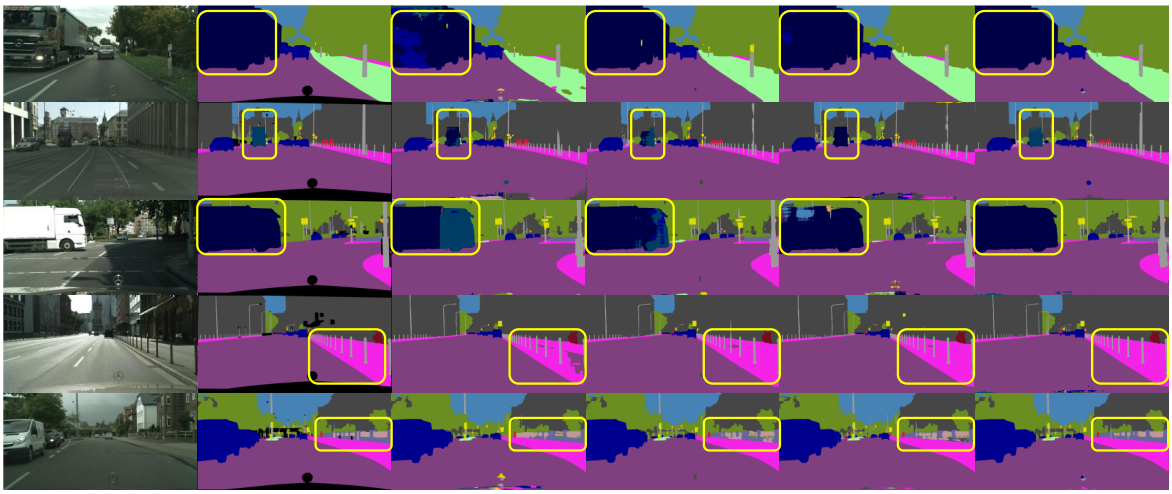

Figure 8. Partial segmentation prediction result comparison on Cityscapes val set. From left to right are (a) input images, (b) ground truth, (c) FCN, (d) CCNet, (e) GSCNN and (f) our BASNet.

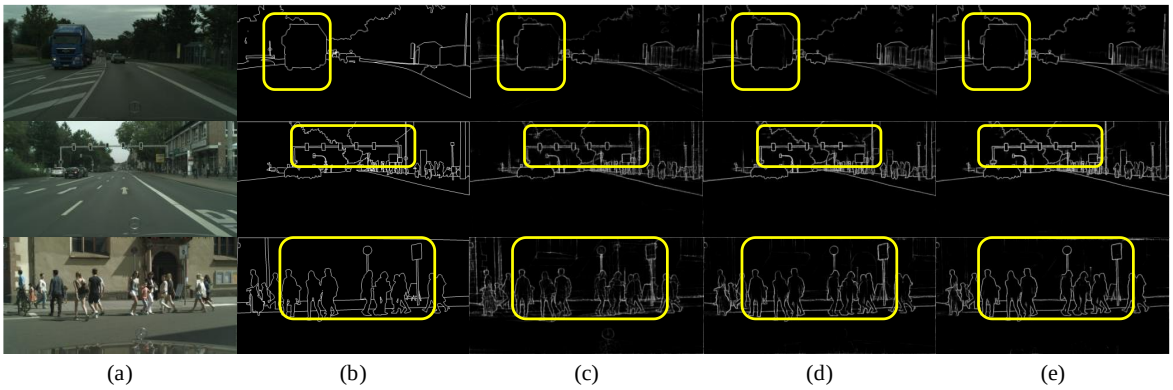

Figure 9. Partial boundary prediction result comparison on Cityscapes val set. From left to right are (a) input images, (b) ground truth, (c) Deeplabv3+, (d) GSCNN and (e) our BASNet.

REFERENCES

1. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
2. H. Lu, M. Zhang, X. Xu, Y. Li, and H. T. Shen, "Deep fuzzy hashing network for efficient image retrieval," *IEEE Transactions on Fuzzy Systems* , 2020.
3. Z. Chen, H. Lu, S. Tian, J. Qiu, T. Kamiya, S. Serikawa, and L. Xu, "Construction of a hierarchical feature enhancement network and its application in fault recognition," *IEEE Transactions on Industrial Informatics* , 2020.
4. H. Lu, R. Yang, Z. Deng, Y. Zhang, G. Gao, and R. Lan, "Chinese image captioning via fuzzy attention-based densenet-bilstm," *ACM Transactions on Multimedia Computing Communications and Applications* , 2020.
5. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934* , 2020.
6. T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5229–5238, 2019.

7. C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1857–1866, 2018.
8. M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1013–1020, IEEE, 2018.
9. K. Li, W. Tao, and L. Liu, "Online semantic object segmentation for vision robot collected video," *IEEE Access* **7**, pp. 107602–107615, 2019.
10. H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE internet of things journal* **5**(4), pp. 2315–2322, 2017.
11. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
12. H. J. Lee, J. U. Kim, S. Lee, H. G. Kim, and Y. M. Ro, "Structure boundary preserving segmentation for medical image with ambiguous boundary," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4817–4826, 2020.
13. L. Jonathan, S. Evan, and D. Trevor, "Fully convolutional networks for semantic segmentation," *IEEE TPAMI* **39**(4), pp. 640–651, 2017.
14. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.
15. L. C. Chen, P. George, S. Florian, and A. Hartwig, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587* , 2017.
16. H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.
17. V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence* **39**(12), pp. 2481–2495, 2017.
18. H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015.
19. H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6819–6829, 2019.
20. T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, and Y. Zhao, "Devil in the details: Towards accurate single and multiple human parsing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, pp. 4814–4821, 2019.
21. J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence* (6), pp. 679–698, 1986.
22. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, and et al., "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.
23. Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 603–612, 2019.
24. J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154, 2019.
25. G. Bertasius, J. Shi, and L. Torresani, "Semantic segmentation with boundary neural fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3602–3610, 2016.
26. X. Li, X. Li, L. Zhang, G. Cheng, J. Shi, Z. Lin, S. Tan, and Y. Tong, "Improving semantic segmentation via decoupled body and edge supervision," *arXiv preprint arXiv:2007.10035* , 2020.

27. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
28. Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 593–602, 2019.
29. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM* **60**(6), pp. 84–90, 2017.
30. M. Zhen, J. Wang, L. Zhou, S. Li, T. Shen, J. Shang, T. Fang, and L. Quan, "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13666–13675, 2020.
31. T.-W. Ke, J.-J. Hwang, Z. Liu, and S. X. Yu, "Adaptive affinity fields for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 587–602, 2018.
32. T. Wu, S. Tang, R. Zhang, J. Cao, and J. Li, "Tree-structured kronecker convolutional network for semantic segmentation," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 940–945, IEEE, 2019.
33. M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3684–3692, 2018.