Selective Feature Aggregation Network with Area-Boundary Constraints for Polyp Segmentation

Yuqi Fang[1], Cheng Chen[1], Yixuan Yuan[2(✉)], and Kai-yu Tong[1(✉)]

[1] Department of Biomedical Engineering, The Chinese University of Hong Kong, Sha Tin, Hong Kong
kytong@cuhk.edu.hk

[2] Department of Electrical Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong
yxyuan.ee@cityu.edu.hk

Abstract. Automatic polyp segmentation is considered indispensable in modern polyp screening systems. It can help the clinicians accurately locate polyp areas for further diagnosis or surgeries. Benefit from the advancement of deep learning techniques, various neural networks are developed for handling the polyp segmentation problem. However, most of these methods neither aggregate multi-scale or multi-receptive-field features nor consider the area-boundary constraints. To address these issues, we propose a novel selective feature aggregation network with the area and boundary constraints. The network contains a shared encoder and two mutually constrained decoders for predicting polyp areas and boundaries, respectively. Feature aggregation is achieved by (1) introducing three up-concatenations between encoder and decoders and (2) embedding Selective Kernel Modules into convolutional layers which can adaptively extract features from different size of kernels. We call these two operations the Selective Feature Aggregation. Furthermore, a new boundary-sensitive loss function is proposed to take into account the dependency between the area and boundary branch, thus two branches can be reciprocally influenced and enable more accurate area predictions. We evaluate our method on the EndoScene dataset and achieve the state-of-the-art results with a Dice of 83.08% and a Accuracy of 96.68%.

1 Introduction

Colorectal cancer is the third leading cause of cancer-related deaths. It is estimated that the number of new cases of colorectal cancer in the US will reach 150 thousand in 2019 [1]. Colorectal polyps are believed one of the early symptoms of colorectal cancer. Therefore, regular screening of colorectal polyps is crucial, during which automatic polyp segmentation is considered an indispensable component. It can help clinicians locate the polyp areas for further diagnosis.

In recent years, numerous deep learning based methods are developed for handling polyp segmentation problem [2,3,10,12,13]. The Fully Convolutional

D. Shen et al. (Eds.): MICCAI 2019, LNCS 11764, pp. 302–310, 2019.
https://doi.org/10.1007/978-3-030-32239-7_34

Network (FCN) [5] is a commonly used architecture, which replaces the fully connected layers of traditional Convolutional Neural Networks (CNNs) with convolutional layers, thus preserving the spatial information for segmentation. Brandao *et al.* [3] adopted the FCN with a pre-trained VGG model to identify and segment polyps from colonoscopy images. Akbari *et al.* [2] adopted a modified version of FCN, i.e., FCN8s, to further improve the polyp segmentation accuracy. Inspired by FCN, the UNet [10], a more powerful and concise network, is then proposed. It also adopts an encoder-decoder architecture but additionally adds some parallel skip concatenations between the encoder and decoder. Based on UNet, SegNet [12] applied the pooling indices from the encoder to the decoder and UNet++ [13] developed a densely connected encoder-decoder network with deep supervision to further enhance the polyp segmentation performance.

Although these networks achieve high performance, there still exist some defects in their architectures. One problem is that compared with UNet++ [13], the FCN [5], UNet [10], and SegNet [12] do not consider multi-scale features. For instance, skip concatenations in UNet connect encoder layers and decoder layers in a parallel manner, which can only aggregate image features of the same scale. In contrast, UNet++ builds up dense connections for integrating more features at different scales but significantly decrease the training and inference efficiency. Besides, all these networks adopt a fixed size of kernel at each layer. This restricts the network from exploring and leveraging representational features obtained from different receptive fields. As a matter of fact, the fusion of features from different scales and multiple receptive fields can remarkably enlarge the perception dimensions, thus helping the network learn more discriminative representations of the input images. In this work, one of our contributions is to optimize the skip concatenations and enrich the diversity of receptive fields at each layer, namely the selective feature aggregation.

Another problem is that the existing methods neglect the area-boundary constraints, a key factor in improving the segmentation performance. Murugesan *et al.* [7,8] talked about this issue and proposed to utilize area and boundary information simultaneously in polyp segmentation, but the relationship between the area and boundary is not further explored. Actually, the dependency between areas and boundaries is quite crucial, which is capable of enhancing the area prediction. Specifically, if the predicted area is larger than ground truth, boundary information can constrain the extended areas, while if the predicted area is much smaller than ground truth, boundary information can enlarge the predicted areas. We leverage such relationship to improve the performance of polyp segmentation via a boundary-sensitive loss function in our method.

In this paper, we propose a novel selective feature aggregation network with a boundary-sensitive loss. Three contributions are claimed as follows. (1) We develop a new encoder-decoder network in which multi-scale features of the encoder are selected and concatenated with two decoders, i.e., the area branch and the boundary branch. Moreover, a Selective Kernel Module (SKM) is embedded into convolutional layers to dynamically learn features from different size of kernels, i.e., 3×3, 5×5, 7×7. (2) A boundary-sensitive loss is proposed to lever-

age the area-boundary constraints, with which the two branches can be mutually improved to produce more accurate predictions. (3) We validate the effectiveness of our method in EndoScene dataset and achieve the state-of-the-art results.

2 Method

2.1 Network Architecture

In this section, we first present the network architecture, as shown in Fig. 1, then introduce two strategies to select and aggregate the polyp features at different scales or receptive fields. Finally we propose a new boundary-sensitive loss via which the area and boundary branch can be reciprocally affected, thus generating more accurate predictions. Our network is composed of a shared encoder, an area branch, and a boundary branch. Each branch contains four convolutional modules. Each module contains three layers integrated with the SKMs. On top of the area branch, a light-weight UNet is adopted to help detect boundaries of the predicted areas. These boundaries and the contours derived from the boundary branch are used to formalize the boundary-sensitive loss.

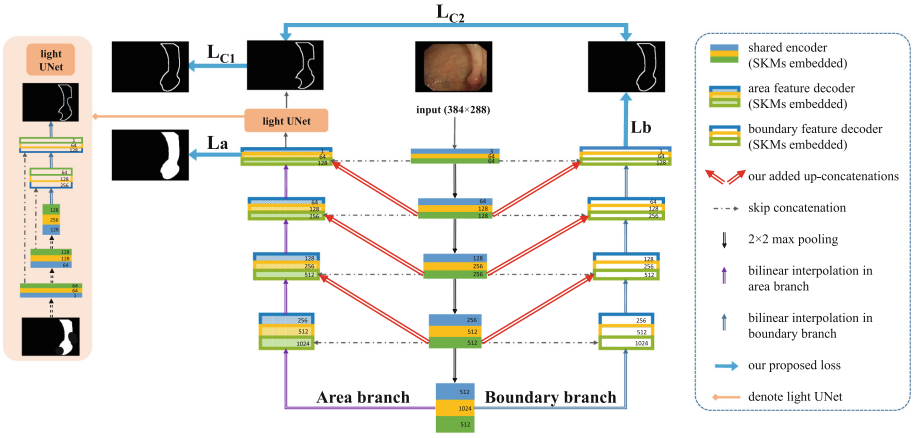

Fig. 1. Illustration of the selective feature aggregation network with area-boundary constraints. Numbers in each block represents the number of feature channels. (Color figure online)

2.2 Selective Feature Aggregation

Up-Concatenations. Inspired by UNet [10] and BESNet [9], we adopt a shared encoder and two decoder branches to assist polyp segmentation, shown in Fig. 1. The shared encoder (middle solid blocks) utilizes five convolutional modules to learn feature representations of the input images. The area branch (left blocks

with stripe) is built for area localization while the boundary branch (right hollow blocks) is built for boundary recovery. Different from the architectures in [9,11,12], where skip concatenations exist in a parallel manner (black dash lines), three extra up-concatenations are added to both the area and boundary branch (red arrow lines) in our method. In this case, multi-scale feature maps derived from deep layers of the shared encoder are copied to shallow layers of the decoders, which enriches the feature representations and keeps the training and inference process more efficient compared with [13].

SKM. Traditionally, most of the convolutional kernels in CNN are with a fixed size, i.e., 3×3, which cannot simultaneously capture features from other receptive fields. To tackle this problem, we adopt the SKM [4] in our method, which can dynamically aggregate features obtained from different size of kernels. Our work is the first study to embed SKMs into the encoder-decoder networks. As shown in Fig. 2, an input feature map $\mathbf{X} \in \mathbb{R}^{C\times W\times H}$ is first filtered by three respective kernels simultaneously, then followed by a Batch Normalization and a ReLU activation, and outputs three distinct feature maps X_3, X_5, X_7. To regress the weight vectors, we first perform element-wise summation of the three feature maps, $\widetilde{X} = \sum X_k (k \in \{3,5,7\})$, then process $\widetilde{X}$ according to the following procedure.

$$f = \mathcal{F}_{fc}\left(\frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H}\widetilde{X}(i,j)\right) \tag{1}$$

$$m_k = e^{W_k f} / \sum e^{W_k f} \quad (k \in \{3,5,7\}) \tag{2}$$

where $\mathcal{F}_{fc}$ is a fully connected operation, $f \in \mathbb{R}^C$ denotes the adaptive features, W_k are learnable parameters in m_k, and m_k is the weight vector corresponding to X_k. With the weight vectors, we are able to adaptively aggregate the feature maps, $\widehat{X} = \sum m_k X_k, k \in \{3,5,7\}$.

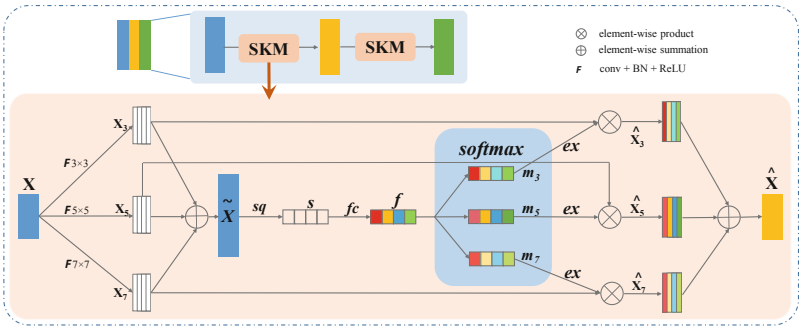

Fig. 2. Selective Kernel Module (SKM). sq: squeeze; fc: fully connected; ex: excitation.

2.3 Boundary-Sensitive Loss

To make full use of the area-boundary constraints, we propose a joint loss function that can mutually propagate the information between the area branch and boundary branch. As shown in Fig. 1, the loss function is composed of three parts: an area loss L_a, a boundary loss L_b, and the area-boundary constraint loss, i.e., L_{C1} and L_{C2}.

Area Loss. L_a consists of a *binary cross-entropy loss* and a *dice loss* [6], which can be represented by the following function.

$$L_a = -\sum_i z_i log(m_i) + (1 - \frac{2\sum_i m_i z_i + \varepsilon}{\sum_i m_i + \sum_i z_i + \varepsilon}) \tag{3}$$

where m_i indicates the probability of pixel i being categorized into polyp class, $z_i \in \{0, 1\}$ is the corresponding area ground truth label, and ε is a small positive number used for increasing numerical stabilities. The cross-entropy loss function penalizes pixel classification errors while the dice loss function measures the overlap between the predicted polyp areas and the area ground truth, which can handle the foreground-background imbalance problem.

Boundary Loss. L_b measures the difference between outputs of boundary branch and boundary ground truth labels, which can be represented as

$$L_b = -\sum_i y_i log(p_i) \tag{4}$$

where p_i denotes the probability of pixel i being the polyp boundary and $y_i \in \{0, 1\}$ is the corresponding boundary ground truth label.

Area-Boundary Constraint Loss. The constraint loss aims to model the dependency between areas and boundaries. To achieve that, we utilize a two-layer UNet to extract boundaries of the predicted polyp areas. Here, the light-weight UNet acts as a differentiable edge detector. The area-boundary constraint loss is composed of two parts, the first part L_{C1} is to minimize the difference between edge detector results and boundary ground truth and the second part L_{C2} aims to minimize the difference between edge detector results and outputs of boundary branch. These two constraint loss functions are represented as

$$L_{C1} = -\sum_i y_i log(q_i) \tag{5}$$

$$L_{C2} = D_{KL}(P||Q) + D_{KL}(Q||P) = -\sum_i p_i log(\frac{q_i}{p_i}) - \sum_i q_i log(\frac{p_i}{q_i}) \tag{6}$$

where q_i is the results predicted by the edge detector, i.e., the light-weight UNet, y_i denotes boundary ground truth, and p_i indicates outputs of boundary branch. D_{KL} denotes *Kullback-Leibler divergence* which can measure the

distance between two distributions. As a result, minimizing D_{KL} is equivalent to making the final outputs of area and boundary branch closer. Intuitively speaking, L_{C2} tries to make the area and boundary branch internally consistent, thus preventing the two branches deviating from each other too much. Compared with L_{C1} that uses boundary supervision explicitly, L_{C2} can implicitly impose the area-boundary constraints. Due to the consistent training goal of the two branches, the shared encoder can learn more discriminative features and help increase the final segmentation performance. The final loss function is as follows.

$$L_{total} = w_a L_a + w_b L_b + w_{C1} L_{C1} + w_{C2} L_{C2} \tag{7}$$

where w_a, w_b, and w_{C1} are set to 1, w_{C2} is set to 0.5 by empirical studies.

3 Experimental Results

3.1 Dataset and Evaluation Metrics

In this work, we adopt the EndoScene dataset [11] which contains 912 images with at least one polyp in each image, acquired from 44 video sequences obtained from 36 subjects. The dataset is divided into a train set, a validation set, and a test set and each sequence is uniquely included in one of the three sets. The area ground truth is provided, while the boundary ground truth is derived by ourselves. The area ground truth is filtered with a 5*5 kernel, where kernel elements are initialized with 0. The element turns to 1 if it overlaps with the area ground truth, otherwise is 0. If kernel elements are not identical, i.e. 0 and 1 exist simultaneously, these elements are regarded as boundary points.

The evaluation metrics we adopt are *Recall, Specificity, Precision, Dice Score, Intersection-over-Union for Polyp* (IoU_P), *IoU for Background* (IoU_B), *Mean IoU* ($mIoU$), and *Accuracy*. In all experiments, the batch size is set to 4. The initial learning rate is 0.01 and decreases by 10 times after 20 and 100 epochs, respectively. The SGD optimizer is used with a weight decay of 0.0005 and a momentum of 0.9. The models are implemented based on the PyTorch framework and trained on a workstation with Intel Core(TM) i7-9700K@3.60 GHz processors and a NVIDIA GeForce RTX 2080 Ti (11 GB) installed.

3.2 Comparative Experiments

The detailed experiment results are presented in Table 1, in which we list the performance of four state-of-the-arts (*row* 1–4) and our proposed method (*row* 5). To further analyze the influence of each component of our method, we conduct several comparative experiments and the results are reported in *row* 6–10. All the results are evaluated on test set, with the checkpoint achieving the highest *Dice Score* on validation set.

As shown in Fig. 3, our method outperforms the four state-of-the-art methods on all the evaluation metrics by a significant increment. Segmentation performance in SegNet [12] and UNet++ [13] is superior to that in FCN8 [3] and

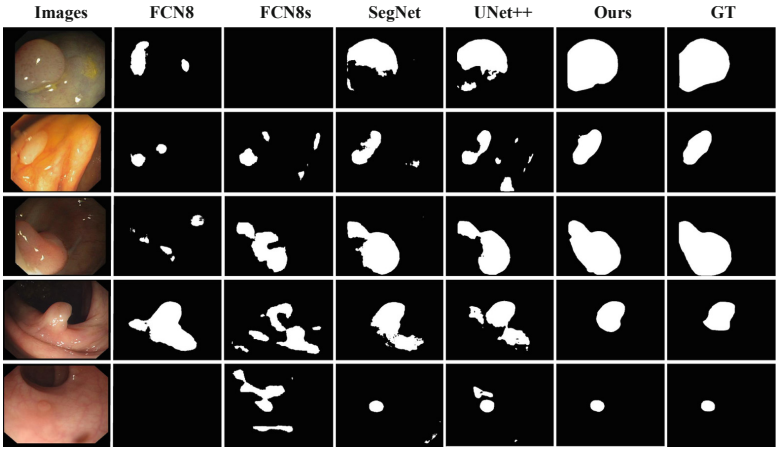

Fig. 3. Polyp segmentation results of different methods.

FCN8s [2], which indicates the significance of skip concatenations on accuracy improvement. However, increasing the density of concatenations does not guarantee more improvement on the segmentation performance. We have attempted numerous combinations of concatenations and find that models with complicated concatenations are hard to train and usually cannot give a competitive result. Compared with our up-concatenation model, UNet++ [13] that adopts even more complicated concatenations shows a significant drop in the performance.

Then, we evaluate the performance of the up-concatenations and the SKM components, which are denoted as UNet+Up and UNet+Up+SKM in Table 1. The results show that the UNet with up-concatenations achieves better performance than UNet alone on all the evaluation criteria. This indicates that up-concatenations can effectively transmit multi-scale features from the encoder to the decoder. The SKM component is also verified to be effective in improving the segmentation performance, especially the $Precision$ and IoU_p which increase by more than 1.5%. This improvement is achieved by allowing the network to dynamically aggregate features derived from different size of kernels, which is a more appropriate strategy than manually specifying the kernel size.

Further, we evaluate the effectiveness of the boundary-sensitive loss. By comparing the 'UNet+Up+SKM+bd' with 'UNet+Up+SKM' in Table 1, it is apparent that the integration of boundary branch helps improve segmentation accuracy a lot. In particular, the $Recall$, $Precison$, $Dice$, and IoU_P show an increment of more than 1% compared with the model that only contains area branch. This indicates that the boundary information can help improve the representation capability of the shared encoder network via the loss backward propagation mechanism. According to last two rows and *Ours* in Table 1, we can see the area-boundary constraint loss functions also play important roles in improving the segmentation performance. L_{C1} aims to minimize the difference between edge detector results generated by the light-weight UNet and boundary ground truth.

Table 1. Comparison with different baselines and other state-of-the-art methods. 'UNet': the typical UNet with area branch; 'Up': up-concatenations; 'SKM': selective kernel module; 'bd': two-branch model with L_a and L_b; 'L_{C1}': the first constraint loss; 'Ours': the two-branch model with total loss, i.e. L_a, L_b, L_{C1} and L_{C2}.

Methods	*Rec*	*Spec*	*Prec*	*Dice*	IoU_P	IoU_B	*mIoU*	*Acc*
FCN8 [3]	53.38	98.83	78.11	55.70	45.48	93.69	69.59	93.97
FCN8s [2]	62.81	98.08	72.60	58.52	45.99	93.22	69.61	93.56
SegNet [12]	81.70	99.03	85.12	79.29	70.33	95.71	83.02	95.99
UNet++ [13]	80.68	99.24	85.31	78.55	69.83	95.71	82.77	95.97
Ours	**83.84**	99.43	90.19	**83.08**	**76.23**	**96.44**	**86.33**	**96.68**
UNet	82.29	99.08	86.13	80.45	72.10	95.84	83.97	96.12
UNet+Up	82.53	99.18	87.13	80.85	72.74	95.88	84.31	96.16
UNet+Up+SKM	82.27	99.37	88.71	81.51	74.41	96.18	85.30	96.41
UNet+Up+SKM+bd	83.30	99.44	90.13	82.61	75.84	96.39	86.12	96.62
UNet+Up+SKM+bd+L_{C1}	83.51	**99.44**	**90.20**	82.97	76.16	96.44	86.30	96.68

Since the results of the edge detectors are built upon the area branch, L_{C1} can explicitly backpropagate the boundary supervision information, thus the boundary constraints can be introduced to the area branch. The purpose of L_{C2} is to minimize the difference between edge detector results and outputs of boundary branch. In this way, edge detector results and the outputs of boundary branch can be mutually constrained, thus making area and boundary branch internally consistent. It is worth noting that introduction of either L_{C1} or L_{C2} improves the segmentation accuracy compared with models without any constraint.

4 Conclusion

We propose a novel selective feature aggregation network with the area and boundary constraints for polyp segmentation. Up-concatenations and SKMs are used to select multi-scale and multi-receptive-field representations of polyp images. Furthermore, a new loss is proposed to take into account the dependency between the area and boundary branch, thus two branches can be reciprocally influenced and enable more accurate predictions. Experimental results demonstrate that our method shows a superior performance over existing state-of-the-arts and it is capable of enabling more accurate polyp localization. The source code is available at https://github.com/Yuqi-cuhk/Polyp-Seg.

References

1. American cancer society: Key statistics for colorectal cancer. http://www.cancer.org/cancer/colonandrectumcancer/detailedguide/colorectal-cancer-keystatistics/. Accessed 18 Mar 2019
2. Akbari, M., et al.: Polyp segmentation in colonoscopy images using fully convolutional network. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 69–72. IEEE (2018)
3. Brandao, P., et al.: Fully convolutional neural networks for polyp segmentation in colonoscopy. In: Medical Imaging 2017: Computer-Aided Diagnosis, vol. 10134, p. 101340F. International Society for Optics and Photonics (2017)
4. Li, X., et al.: Selective kernel networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–519 (2019)
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
6. 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), pp. 565–571. IEEE (2016)
7. Murugesan, B., et al.: Joint shape learning and segmentation for medical images using a minimalistic deep network. arXiv preprint arXiv:1901.08824 (2019)
8. Murugesan, B., et al.: Psi-Net: shape and boundary aware joint multi-task deep network for medical image segmentation. arXiv preprint arXiv:1902.04099 (2019)
9. Oda, H., et al.: BESNet: boundary-enhanced segmentation of cells in histopathological images. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11071, pp. 228–236. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00934-2_26
10. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
11. Vázquez, et al.: A benchmark for endoluminal scene segmentation of colonoscopy images. J. Healthc. Eng. **2017** (2017)
12. Wickstrøm, K., Kampffmeyer, M., Jenssen, R.: Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps. arXiv preprint arXiv:1807.10584 (2018)
13. Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J.: UNet++: a nested U-Net architecture for medical image segmentation. In: Stoyanov, D., et al. (eds.) DLMIA/ML-CDS -2018. LNCS, vol. 11045, pp. 3–11. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00889-5_1