RFC-Net: Learning High Resolution Global Features for Medical Image Segmentation on a Computational Budget (Student Abstract)

Sourajit Saha[1], Shaswati Saha[2], Md Osman Gani[3], Tim Oates[4], David Chapman[5]

[1,4] Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, MD 21250, USA
[2, 3] Department of Information Systems, University of Maryland Baltimore County, MD 21250, USA
[5] Department of Computer Science, University of Miami, FL 33124, USA
{[1]ssaha2, [2]ssaha3, [3]mogani, [4]oates}@umbc.edu, [5]dchapman@cs.miami.edu

Abstract

Learning High-Resolution representations is essential for semantic segmentation. Convolutional neural network (CNN) architectures with downstream and upstream propagation flow are popular for segmentation in medical diagnosis. However, due to performing spatial downsampling and upsampling in multiple stages, information loss is inexorable. On the contrary, connecting layers densely on high spatial resolution is computationally expensive. In this work, we devise a Loose Dense Connection Strategy to connect neurons in subsequent layers with reduced parameters. On top of that, using a m-way Tree structure for feature propagation we propose Receptive Field Chain Network (RFC-Net) that learns high-resolution global features on a compressed computational space. Our experiments demonstrates that RFC-Net achieves *state-of-the-art* performance on *Kvasir* and *CVC-ClinicDB* benchmarks for Polyp segmentation. Our code is publicly available at ***github.com/sourajitcs/RFC-NetAAAI23***.

Introduction

Decreasing spatial resolution in CNN's forward propagation engenders difficulty in learning high-resolution global features which affects pixel-wise image segmentation quality. Preserving spatial resolution therefore, has been a consistent design choice among a number of high-precision CNN models (Wang et al. 2020) proposed in recent times. However, such design choices (Han, Yoo, and Oh 2022) lead to computational overhead due to having multiple ResNet and transformer blocks running in parallel at high-resolution (spatial). Despite their high precision, training such models are often challenging in the presence of constraints in computational budget and n-dimensional imagery where $n \in \{\mathcal{Z}, n > 2\}$. To mitigate this bottleneck, we propose **R**eceptive **F**ield **C**hain **Net**work **(RFC-Net)** by means of devising the following:

$$\|\Theta_{SDCS}\|_l^{l+1} = d_{l+1}([\mathbf{k^2}]d_l + \mathbf{d_{l+1}}) \quad (1)$$

- Introduce **L**oose **D**ense **C**onnection **S**trategy **(LDCS)** for reducing the number of parameters.
- Design a **m-way Tree structure** to learn features through a chain constituting all possible combination of different receptive fields from a selected set of kernels.

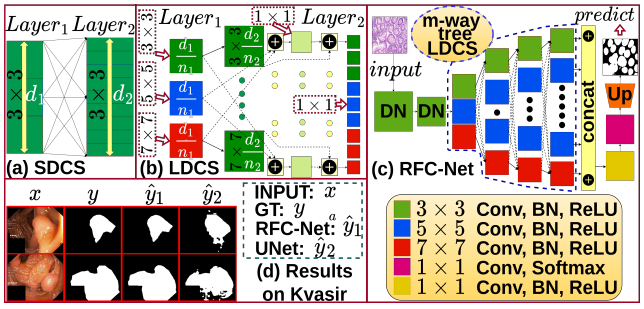

Figure 1: Enhanced view recommended. **(a):** Conventional Strong Dense Connection Strategy (SDCS). **(b):** Proposed Loose Dense Connection Strategy (LDCS). **(c):** Proposed RFC-Net Architecture: Expanding with m-way tree on uncompressed resolution with LDCS (DN: Downsample). **(d):** Comparison between the quality of RFC-Net and UNet's prediction $(\hat{y})$ mask on Kvasir dataset (GT: Ground Truth).

Methodology

As constituent of our proposed RFC-Net we first introduce Loose Dense Connection Strategy (LDCS) which in comparison with conventional Strong Dense Connection Strategy (SDCS) aids in reducing the number of parameters as illustrated in Figure 1(a-b). We define strong connection between two neurons as being filtered through a $k \times k$ convolutional kernel $(\forall k > 1)$ since such connections enable learning spatial correlations whereas, we define loose connection as using a $k \times k$ kernel $(\forall k = 1)$ and this operation is equivalent to that of a linear layer with no propagation of spatial information about neighbouring pixels. Using $k \times k$ kernels, conventional SDCS (Ronneberger, Fischer, and Brox 2015) strongly $(k > 1)$ connects every neurons from l^{th} layer to accrue each neuron in the $(l+1)^{th}$ layer and Equation 1 denotes the number of parameters $(\|\Theta\|)$ to constitute the entire $(l+1)^{th}$ layer where d_l is the total number of neurons in layer l, $l = \{1, 2, .., L\}$ and L is the total number of layers. However, in LDCS we split the number of neurons in l^{th} layer to n_l groups. Using our proposed LDCS, we first of all strongly $(k > 1)$ connect one of those n_l groups in l^{th} layer to one of those n_{l+1} groups in $(l+1)^{th}$ layer. As illustrated in Figure 1(b), we then concatenate the other $(d_l - n_l)$ groups in l^{th} layer and loosely $(k = 1)$ connect

them to the previously chosen n_{l+1}^{th} group in $(l+1)^{th}$ layer where we performed the strong connection with n_l^{th} group. Finally, we concatenate the strongly and loosely connected neurons to the aforementioned n_{l+1}^{th} group and pass through one more (1×1) kernel to construct the final n_{l+1}^{th} group in $(l+1)^{th}$ layer. Equation 2 depicts the number of parameters $(\|\Theta\|)$ required to construct the entire $(l+1)^{th}$ layer using LDCS. The computation reduction factors driven by LDCS are **highlighted** in Equation 1 and 2.

$$\|\Theta_{LDCS}\|_l^{l+1} = d_{l+1}(\left[\frac{\mathbf{k^2}}{\mathbf{n_l}} + \frac{n_l - 1}{n_l}\right] d_l + \frac{\mathbf{d_{l+1}}}{\mathbf{n_{l+1}}}) \quad (2)$$

$$x_i^{l+1} = \mathcal{F}_{1\times1}(\mathcal{F}_{k\times k}(x^l_{\lceil i/m \rceil}) \oplus \mathcal{F}_{1\times1}(\oplus_{j\in n_l} x^l_{j\neq\lceil i/m\rceil})) \quad (3)$$

Secondly, we introduce a m-way Tree structure into the network and the number of groups at layer l, $n_l = m^l$. We repeatedly deploy m kernels $(k > 1)$ of different receptive fields throughout each layers across the network as shown in Figure 1(c). Equation 3 shows how connecting these groups across layers with LDCS using a m-way Tree structure thus facilitates the propagation of features through a chain constituting all possible combination of different receptive fields with different values of k, where $\mathcal{F}_{k\times k}$ is a convolution kernel of size $k \times k$ and x_i^l denotes i^{th} intermediate feature representation at layer l. However, connecting only one group from a layer to that of the next one (considering only the $\mathcal{F}_{k\times k}$ portion in Equation 3) leads to amassing m^L number of week-performing segregated networks due to lack of information exchange. On the contrary, connecting all of the groups using the conventional SDCS strategy will result in a combinatorial explosion in computation. We observe that, replacing n_l in Equation 2 with m^l exponentially reduces the computation constraints. Therefore, juxtaposing m-way Tree structure with different receptive fields and LDCS enables Receptive Field Chain Network termed as RFC-Net (Figure 1(c)) to learn high-resolution aware features within computational budget. Finally, learning features on a hierarchical space of cascaded receptive fields without severe spatial downsampling (we only downsample twice before passing through the m-way tree, as illustrated in Figure 1(c)) helps RFC-Net learn more robust global features.

Experiments, Results And Observations

We evaluate RFC-Net on three binary segmentation benchmarks- (1) Kvasir SEG Polyp Dataset (Gastrointestinal Disease Detection, train:test=850:150), (2) GlaS Dataset (Gland Segmentation in Colon Histology, train:test=132:33), (3) CVC-ClinicDB (Polyp segmentation from colonoscopy video frames, train:test=521:91). We use SGD optimizer with momentum=0.9, weight decay=0.0005. Furthermore, we train (without data augmentation) all our models on the benchmarks with Online Hard Example Mining Cross Entropy (OHEM CE) loss (threshold=0.7). Firstly, we train Kvasir SEG Polyp Dataset (resized to 200×200, batch size of 4) for 160 epochs (base learning rate=0.01, step size=45, converges in 77 epochs). Secondly, we train GlaS Dataset (resized to 300×300, batch size of 2) for 500 epochs (base learning rate=0.01, step size=60, converges in 259 epochs). Thirdly, we train CVC-ClinicDB Dataset (resized to 200×300, batch size of 3) for 160 epochs (base learning rate=0.01, step size=45, converges in 85 epochs). We ran all of our experiments on PyTorch using one Nvidia RTX 3090 and one Nvidia RTX 3070 GPU.

	Computation		*Performance (mIOU %)*		
Model	*Params*	*GFLOPs*	*Kvasir*	*GlaS*	*CVC-DB*
U-Net	07.76M	30.75B	74.39	83.51	75.91
U-Net++	09.04M	34.91B	75.95	84.02	79.27
ResUNet	13.04M	43.56B	77.77	**85.67**	81.33
ESPNet-C	00.41M	02.49B	75.95	65.43	72.53
RFC-Net^a	**05.76M**	**18.13B**	**81.31**	77.88	**85.90**
RFC-Net^b	04.49M	14.03B	79.15	75.34	83.34
RFC-Net^c	00.39M	01.27B	76.41	75.49	79.51
RFC-Net^d	00.28M	00.91B	73.24	66.17	77.68

Table 1: Results on Kvasir, GlaS, CVC-ClinicDB Datasets. GFLOPs are calculated on 224×224 RGB image. RFC-Net^a:[m=3,k=3,5,7], RFC-Net^b:[m=3,k=3,3,3], RFC-Net^c:[m=2,k=3,5], RFC-Net^d:[m=2,k=3,3].

In Table 1, we narrate how RFC-Net performs in comparison to existing heavier and lighter models. We observe that while reducing parameters RFC-Net^a outperforms the existing models in Kvasir and CVC-ClinicDB benchmarks and exhibits comparable performance on GlaS dataset. We further perform an ablation study (Table 1), to inspect the efficacy of our model. We notice that RFC-Net^b with even lesser parameters outperforms the existing models on kvasir and CVC-Clinic DB benchmark. We further observe that, it is the exploitation of the m-way tree structure by RFC-Net^a for using enhanced receptive field that leads to a sharp performance gain (RFC-Net^b $\mapsto$RFC-Net^a, RFC-Net^d $\mapsto$RFC-Net^c). Additionally, the performance of RFC-Net's lightweight versions (RFC-Net^c, RFC-Net^d) is substantially higher than that of ESPNet-C which is another existing lightweight CNN for segmentation. Figure 1(d) further depicts the higher quality (in comparison to UNet) of prediction masks produced by RFC-Net. Being computationally inexpensive, RFC-Net can be further applied to 3D and 4D image segmentation for *Computer Aided Diagnosis*.

References

Han, D.; Yoo, J.; and Oh, D. 2022. SeeThroughNet: Resurrection of Auxiliary Loss by Preserving Class Probability Information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4463–4472.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3349–3364.