Progressively Normalized Self-Attention Network for Video Polyp Segmentation

Ge-Peng Ji[1,2], Yu-Cheng Chou[2], Deng-Ping Fan[1,✉]
Geng Chen[1], Huazhu Fu[1], Debesh Jha[3], and Ling Shao[1]

[1] Inception Institute of AI (IIAI) [2] Wuhan University [3] SimulaMet
dengpfan@gmail.com

Abstract. Existing video polyp segmentation (VPS) models typically employ convolutional neural networks (CNNs) to extract features. However, due to their limited receptive fields, CNNs cannot fully exploit the global temporal and spatial information in successive video frames, resulting in false positive segmentation results. In this paper, we propose the novel **PNS-Net** (Progressively Normalized Self-attention Network), which can efficiently learn representations from polyp videos with real-time speed ($\sim$**140fps**) on a single RTX 2080 GPU and no post-processing. Our *PNS-Net* is based solely on a basic normalized self-attention block, equipping with recurrence and CNNs entirely. Experiments on challenging VPS datasets demonstrate that the proposed *PNS-Net* achieves state-of-the-art performance. We also conduct extensive experiments to study the effectiveness of the channel split, soft-attention, and progressive learning strategy. We find that our *PNS-Net* works well under different settings, making it a promising solution to the VPS task.

Keywords: Normalized self-attention · Polyp segmentation · Colonoscopy

1 Introduction

Early diagnosis of colorectal cancer (CRC) plays a vital role in improving the survival rate of CRC patients. In fact, the survival rate in the first stage of CRC is over 95%, decreasing to below 35% in the fourth and fifth stages [4]. Currently, colonoscopy is widely adopted in clinical practice and has become a standard method for screening CRC. During the colonoscopy, physicians visually inspect the bowel with an endoscope to identify polyps, which can develop into CRC if left untreated. In practice, colonoscopy is highly dependent on the physicians' level of experience and suffers from a high polyp miss rate [18]. These limitations can be resolved with automatic polyp segmentation techniques, which segment polyps from colonoscopy images/videos without intervention from physicians. However, accurate and real-time polyp segmentation is a challenging task due to the low boundary contrast between a polyp and its surroundings and the large shape variation of polyps [8].

G.-P. Ji and Y.-C. Chou contributed equally. Code: http://dpfan.net/pranet/

Significant efforts have been dedicated to overcoming these challenges. In early studies, learning-based methods turned to handcrafted features [16,20], such as color, shape, texture, appearance, or some combination. These methods train a classifier to separate the polyps from the background. However, they usually suffer from low accuracy due to the limited representation capability of handcrafted features in depicting heterogeneous polyps, as well as the close resemblance between polyps and hard mimics [24]. In more recent studies, deep learning methods have been used for polyp segmentation [24,26]. Although these methods have made some progress, they only use bounding boxes to detect polyps, and therefore cannot accurately locate the boundaries. To solve this, Brandao *et al.* [5] adopted a fully convolutional networks (FCN) with a pre-trained model to recognize and segment polyps. Later, Akbari *et al.* [1] introduced a modified FCN to increase the accuracy of polyp segmentation. Inspired by the success of UNet [19] in biomedical image segmentation, UNet++ [28] and ResUNet [13] are employed for polyp segmentation and achieved good results. Some methods also focus on area-boundary constraints. For instance, Psi-Net [17] makes use of polyp boundary and area information simultaneously. Fang *et al.* [9] introduced a three-step selective feature aggregation network. ACSNet [25] utilized an adaptive context selection based encoder-decoder framework. Zhong *et al.* [27] propose a context-aware network based on adaptive scale and global semantic context. Introduced more recently, the current golden standard for image polyp segmentation, PraNet [8], applies area and boundary cues in a reverse attention module, achieving the cutting-edge performance. However, these methods have only been trained and evaluated on still images and focus on static information, ignoring the temporal information in endoscopic videos which can be exploited for better results. To this end, Puyal *et al.* [18] propose a hybrid 2D/3D CNN architecture. Their model aggregates spatial and temporal correlations and achieves better segmentation results. However, the spatial correlation between frames is restricted by the size of the kernel, preventing the accurate segmentation of fast videos.

Recently, the self-attention network [22] has shown superior performance in computer vision tasks such as video object segmentation [10], image super-resolution [23], and others. Inspired by this, in this paper, we propose a novel self-attention framework, called the **P**rogressively **N**ormalized **S**elf-attention **Net**work (***PNS-Net***), for the video polyp segmentation (VPS) task. Our contributions are as follows:

- Different from existing CNN-based models, the proposed *PNS-Net* framework is a self-attention model for VPS, introducing a new perspective for addressing this task.
- To fully utilize the temporal and spatial cues, we propose a simple normalized self-attention (NS) block. The NS block is flexible (backbone-free) and efficient, enabling it to easily be embedded into current CNN-based encoder-decoder architectures for better performance.
- We evaluate the proposed *PNS-Net* on challenging VPS datasets and compare it with two classical methods (*i.e.*, UNet [19] and UNet++ [28]) and
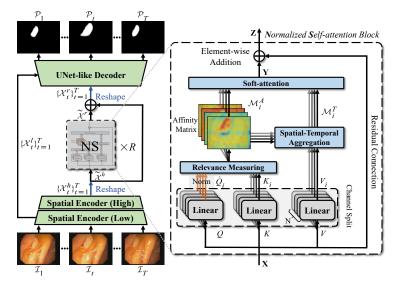
Fig. 1: Pipeline of the proposed *PNS-Net*, including the normalized self-attention block (see § 2.1) with a stacked (×R) learning strategy (see § 2.2).

three cutting-edge models (*i.e.*, ResUNet [13], ACSNet [25], and PraNet [8]). Experimental results show that *PNS-Net* achieves state-of-the-art performance with real-time speed. All the training data, models, results, and evaluation tools will be released to advance the development of this field.

2   Method

2.1   Normalized Self-attention (NS)

Motivation. Recently, the self-attention mechanism [22] has been widely exploited in many popular computer vision tasks. However, in our initial studies, we found that introducing the original self-attention mechanism to the VPS task does not achieve satisfactory results (*i.e.*, high accuracy and speed).

Analysis. For the VPS task, multi-scale polyps move at various speeds. Thus, dynamically updating the receptive field of the network is important. Further, the self-attention, such as the non-local network [22], incurs a high computational and memory cost, which limits the inference speed for our fast and dense prediction task. Motivated by the recent video salient object detection model [10], we utilize the **channel split**, **query-dependent**, and **normalization** rules to reduce the computational cost and improve the accuracy, respectively.

Channel Split Rule. Specifically, given an input feature (*i.e.*, $\mathbf{X} \in \mathbb{R}^{T \times H \times W \times C}$) extracted from T video frames with a size of $H \times W$ and C channels, we first utilize three linear embedding functions $\theta(\cdot)$, $\phi(\cdot)$, and $g(\cdot)$ to generate the corresponding attention features, which are implemented by a $1 \times 1 \times 1$ convolutional

layer [22]. This can be expressed as:

$$Q = \theta(\mathbf{X}), K = \phi(\mathbf{X}), V = g(\mathbf{X}). \tag{1}$$

Then we split each attention feature $\{Q, K, V\} \in \mathbb{R}^{T \times H \times W \times C}$ into N groups along the channel dimension and generate query, key, and value features, *i.e.*, $\{Q_i, K_i, V_i\} \in \mathbb{R}^{T \times H \times W \times \frac{C}{N}}$, where $i = \{1, 2, \cdots, N\}$.

Query-Dependent Rule. To extract the spatial-temporal relationship between successive video frames, we need to measure the similarity between query features Q_i and key features K_i. Inspired by [10], we introduce N relevance measuring (*i.e.*, query-dependent rule) blocks to compute the spatial-temporal affinity matrix for the *constrained neighborhood* of the target pixel. Rather than computing the response between a query position and the feature at all positions, as done in [22], the relevance measuring block can capture more relevance regarding the target object within T frames. More specifically, given a sliding window with fixed kernel size k and dilation rate $d_i = 2i - 1$, we get the corresponding constrained neighborhood in K_i for query pixel $\mathbf{X}^q$ of Q_i in position (x, y, z), which can be obtained by a sampling function $\mathcal{F}^S$. This is computed by:

$$\mathcal{F}^S \langle \mathbf{X}^q, K_i \rangle \in \mathbb{R}^{T(2k+1)^2 \times \frac{C}{N}} = \Sigma_{m=x-kd_i}^{x+kd_i} \Sigma_{n=y-kd_i}^{y+kd_i} \Sigma_{t=1}^{T} K_i(m, n, t), \tag{2}$$

where $1 \le x \le H$, $1 \le y \le W$, and $1 \le z \le T$. Thus, the size of the constrained neighborhood depends on the various spatial-temporal receptive fields with different kernel size k, dilation rate d_i, and frame number T, respectively.

Normalization Rule. However, the internal covariate shift problem [11] exists in the feed-forward of input Q_i, incurring that the layer parameters cannot dynamically adapt the next mini-batch. Therefore, we maintain a fixed distribution for Q_i via:

$$\hat{Q}_i = \text{Norm}(Q_i), \tag{3}$$

where Norm is implemented by layer normalization [2] *along temporal dimension.*
Relevance Measuring. Finally, the affinity matrix is computed as:

$$\mathcal{M}_i^A \in \mathbb{R}^{THW \times T(2k+1)^2} = \text{Softmax}(\frac{\hat{Q}_i \mathcal{F}^S \langle \hat{\mathbf{X}}^q, K_i \rangle^{\mathbf{T}}}{\sqrt{C/N}}), \text{ when } \hat{\mathbf{X}}^q \in \hat{Q}_i, \tag{4}$$

where $\sqrt{C/N}$ is a scaling factor to balance the multi-head attention [21].
Spatial-Temporal Aggregation. Similar to relevance measuring, we also compute the spatial-temporally aggregated features $\mathcal{M}_i^T$ within the constrained neighborhood during temporal aggregation. This can be formulated as:

$$\mathcal{M}_i^T \in \mathbb{R}^{THW \times \frac{C}{N}} = \mathcal{M}_i^A \mathcal{F}^S \langle \mathbf{X}^a, V_i \rangle, \text{ when } \mathbf{X}^a \in \mathcal{M}_i^A, \tag{5}$$

Soft-Attention. We use a soft-attention block to synthesize features from the group of affinity matrices $\mathcal{M}_i^A$ and aggregated features $\mathcal{M}_i^T$. During the synthesis process, relevant spatial-temporal patterns should be enhanced while less relevant ones should be suppressed. We first concatenate a group of affinity

matrices $\mathcal{M}_i^A$ along the channel dimension to generate $\mathcal{M}^A$. Thus, the soft-attention map $\mathcal{M}^S$ is computed by:

$$\mathcal{M}^S \in \mathbb{R}^{THW \times 1} \leftarrow \mathtt{max}\mathcal{M}^A, \text{ when } \mathcal{M}^A \in \mathbb{R}^{THW \times T(2k+1)^2 N}, \qquad (6)$$

where the $\mathtt{max}$ function computes the channel-wise maximum value. Then we concatenate a group of the spatial-temporally aggregated features $\mathcal{M}_i^T$ along the channel dimension to generate $\mathcal{M}^T$.

Normalized Self-attention. Finally, our NS block can be computed as:

$$\mathbf{Z} \in \mathbb{R}^{T \times H \times W \times C} = \mathbf{X} + \mathbf{Y} = \mathbf{X} + (\mathcal{M}^T \mathbf{W}_T) \circledast \mathcal{M}^S, \qquad (7)$$

where $\mathbf{W}_T$ is the learnable weight and $\circledast$ is the channel-wise Hadamard product.

2.2 Progressive Learning Strategy

Encoder. For fair comparison, we use the same backbone (*i.e.*, Res2Net-50) as in PraNet [8]. Given a polyp video clip with T frames as input (*i.e.*, $\{\mathcal{I}\}_{t=1}^T \in \mathbb{R}^{H' \times W' \times 3}$), we first feed it into a spatial encoder to extract two spatial features from the conv3_4 and conv4_6 layers, respectively. To alleviate the computational burden, we adopt an RFB-like [15] module to reduce the feature channel. Thus, we generate two spatial features, including low-level (*i.e.*, $\{\mathcal{X}_t^l\}_{t=1}^T \in \mathbb{R}^{H^l \times W^l \times C^l}$) and high-level (*i.e.*, $\{\mathcal{X}_t^h\}_{t=1}^T \in \mathbb{R}^{H^h \times W^h \times C^h}$)[1].

Progressively Normalized Self-attention (PNS). Most attention strategies aim to refine candidate features, such as first-order [8] and second-order [22,21] functions. As such, the strong semantic information in high-level features might be diffused gradually during the forward pass of the network. To alleviate this, we introduce a progressive residual learning strategy in our NS block. Specifically, we first reshape the corresponding high-level features $\{\mathcal{X}_t^h\}_{t=1}^T$ of consecutive input frames into a temporal feature, which can be viewed as a four-dimensional tensor (*i.e.*, $\widetilde{\mathcal{X}}^h \in \mathbb{R}^{T \times H^h \times W^h \times C^h}$). Then we refine $\widetilde{\mathcal{X}}^h$ via stacked normalized self-attention in a progressive manner:

$$\widetilde{\mathcal{X}}^r \in \mathbb{R}^{T \times H^h \times W^h \times C^h} = \mathtt{NS}^{\times R}(\widetilde{\mathcal{X}}^h) = \mathtt{NS}^{\times R}(\mathcal{F}^R(\{\mathcal{X}_t^h\}_{t=1}^T)), \qquad (8)$$

where $\mathtt{NS}^{\times R}$ means that R normalized self-attention blocks are stacked in the refinement process. $\mathcal{F}^R$ is the reshaping function for the temporal dimension. To allow this block to easily be plugged into pre-trained networks, the commonly adopted solution is to add a residual learning process. Finally, the refined spatial-temporal feature is generated by:

$$\{\mathcal{X}_t^r\}_{t=1}^T \in \mathbb{R}^{H^h \times W^h \times C^h} = \mathcal{F}^R(\widetilde{\mathcal{X}}^h + \widetilde{\mathcal{X}}^r). \qquad (9)$$

Decoder and Learning Strategy. We combine the low-level feature $\{\mathcal{X}_t^l\}_{t=1}^T$ from the spatial decoder and the spatial-temporal feature $\{\mathcal{X}_t^r\}_{t=1}^T$ from the PNS block via a two-stage UNet-like decoder $\mathcal{F}^D$. Thus, the output of our method is computed by $\{\mathcal{P}_t\}_{t=1}^T = \mathcal{F}^D(\{\mathcal{X}_t^l\}_{t=1}^T, \{\mathcal{X}_t^r\}_{t=1}^T)$. We adopt the standard *cross-entropy* loss function in the learning process.

---

[1] We set $H^l = \frac{H'}{4}$, $W^l = \frac{W'}{4}$, $C^l = 24$, $H^h = \frac{H'}{8}$, $W^h = \frac{W'}{8}$, and $C^h = 32$.

3   Experiments

3.1   Implementation Details

Datasets. We adopt four widely used polyp datasets in our experiments, including image-based (*i.e.*, Kvasir [12]) and video-based (*i.e.*, CVC-300 [4], CVC-612 [3], and ASU-Mayo [20]) ones. Kvasir is a large-scale and challenging dataset, which consists of 1,000 polyp images with fully annotated pixel-level ground truths (GTs). The whole Kvasir is used for training. ASU-Mayo contains 10 negative video samples from normal subjects and 10 positive samples from patients. We only adopt the positive part for training. Following the same protocol as [4,3], we split the videos from CVC-300 (12 clips) and CVC-612 (29 clips) into 60% for training, 20% for validation, and 20% for testing.

Training. Due to the limited video training data, we try to fully utilize large-scale image data to capture more appearances of the polyp and scene. Thus, we train our model in two steps: *i) Pre-training phase.* We remove the normalized self-attention (NS) block from *PNS-Net* and pre-train the static backbone using an image-based polyp dataset (*i.e.*, Kvasir [12]) and the training set of video-based polyp datasets (*i.e.*, CVC-300 [4], CVC-612 [3], and ASU-Mayo [20]). The initial learning rate of the Adam algorithm and the weight decay are both 1e-4. The static part of our *PNS-Net* convergences after 100 epochs. *ii) Fine-tuning phase.* We plug the NS block into our *PNS-Net* and fine-tune the whole network using the video polyp datasets, including the ASU-Mayo and the training sets of CVC-300 and CVC-612. We set the number of attention groups $N = 4$ and the number of stacked normalized self-attention blocks $R = 2$, along with a kernel size of $k = 3$. The initial learning is set to 1e-4, and the whole model is fine-tuned over one epoch. In this way, although the densely labeled VPS data is scarce, our *PNS-Net* still achieves good generalization performance.

Testing and Runtime. To test the performance of our *PNS-Net*, we validate it on challenging datasets, including the test set of CVC-612 (*i.e.*, CVC-612-T), the validation set of CVC-612 (*i.e.*, CVC-612-V), and the test/validation set of CVC-300 (*i.e.*, CVC-300-TV). During inference, we sample $T=5$ frames from a polyp clip and resize them to 256×448 as the input. For final prediction, we use the output $\mathcal{P}_t$ of the network followed by a *sigmoid* function. Our *PNS-Net* achieves a speed of ∼140fps on a single RTX 2080 GPU without any post-processing (*e.g.*, CRF [14]). The speeds of the compared methods are listed in Tab. 1.

3.2   Evaluation on Video Polyp Segmentation

Baselines. We re-train five cutting-edge polyp segmentation baselines (*i.e.*, UNet [19], UNet++ [28], ResUNet [13], ACSNet [25], and PraNet [8]) with the same data used by our *PNS-Net*, under their default settings, for fair comparison.

Metrics. The metrics used included: (1) maximum Dice (maxDice), which measures the similarity between two sets of data; (2) maximum specificity (maxSpe), which refers to the percentage of the samples that are negative and are judged as such; (3) maximum IoU (maxIoU), which measures the overlap between two

Table 1: Quantitative results on different datasets.

	Metrics	2018~2019			2020		2021
		UNet MICCAI [19]	UNet++ TMI [28]	ResUNet ISM [13]	ACSNet MICCAI [25]	PraNet MICCAI [8]	*PNS-Net* (OUR)
	Speed	108fps	45fps	20fps	35fps	97fps	**140fps**
CVC-300-TV	maxDice↑	0.639	0.649	0.535	0.738	0.739	**0.840**
	maxSpe↑	0.963	0.944	0.852	0.987	0.993	**0.996**
	maxIoU↑	0.525	0.539	0.412	0.632	0.645	**0.745**
	S_α ↑	0.793	0.796	0.703	0.837	0.833	**0.909**
	E_ϕ ↑	0.826	0.831	0.718	0.871	0.852	**0.921**
	M ↓	0.027	0.024	0.052	0.016	0.016	**0.013**
CVC-612-V	maxDice↑	0.725	0.684	0.752	0.804	0.869	**0.873**
	maxSpe↑	0.971	0.952	0.939	0.929	0.983	**0.991**
	maxIoU↑	0.610	0.570	0.648	0.712	0.799	**0.800**
	S_α ↑	0.826	0.805	0.829	0.847	0.915	**0.923**
	E_ϕ ↑	0.855	0.830	0.877	0.887	0.936	**0.944**
	M ↓	0.023	0.025	0.023	0.054	0.013	**0.012**
CVC-612-T	maxDice↑	0.729	0.740	0.617	0.782	0.852	**0.860**
	maxSpe↑	0.971	0.975	0.950	0.975	0.986	**0.992**
	maxIoU↑	0.635	0.635	0.514	0.700	0.786	**0.795**
	S_α ↑	0.810	0.800	0.727	0.838	0.886	**0.903**
	E_ϕ ↑	0.836	0.817	0.758	0.864	**0.904**	0.903
	M ↓	0.058	0.059	0.084	0.053	0.038	**0.038**

(a) Frame   (b) GT   (c) OUR   (d) PraNet   (e) ACSNet   (f) ResUNet   (g) UNet++   (h) UNet
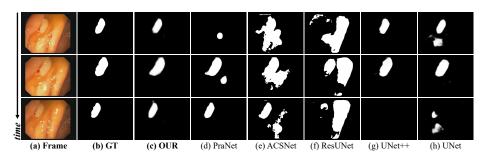
Fig. 2: Qualitative results on CVC-612-T [3]. For more visualization results please refer to the supplementary material (*i.e.*, PDF file and videos).

masks; (4) S-measure [6] (S_α), which evaluates region- and object-aware structural similarity; (5) enhanced-alignment measure [7] (E_ϕ), which measures pixel-level matching and image-level statistics; and (6) mean absolute error (M), which measures the pixel-level error between the prediction and GT.

Qualitative Comparison. In Fig. 2, We provide the polyp segmentation results of our *PNS-Net* on CVC-612-T. Our model can accurately locate and segment polyps in many difficult situations, such as different sizes, homogeneous areas, different textures, *etc*.

Quantitative Comparison. Quantitative comparison results are summarized in Tab. 1. We conduct three experiments on test datasets to verify the model's performance. CVC-300-TV consists of both validation set and test set, which include six videos in total. CVC-612-V and CVC-612-T each contain five videos. On CVC-300, where all the baseline methods perform poorly, our *PNS-Net* achieves remarkable performance in all metrics and outperforms all SOTA methods by

Table 2: Ablation studies. See § 3.3 for more details.

No.	Variants					CVC-300-TV				CVC-612-T			
	Base	N	Soft	Norm	R	maxDice↑	maxIoU↑	S_α ↑	E_ϕ ↑	maxDice↑	maxIoU↑	S_α ↑	E_ϕ ↑
#1	✓					0.778	0.665	0.850	0.858	0.850	0.778	0.896	0.885
#2	✓	1			1	0.755	0.650	0.865	0.844	0.850	0.779	0.896	0.891
#3	✓	2			1	0.790	0.679	0.876	0.872	0.825	0.746	0.870	0.856
#4	✓	4			1	0.809	0.709	0.893	0.884	0.834	0.760	0.881	0.867
#5	✓	8			1	0.763	0.663	0.867	0.842	0.787	0.702	0.841	0.829
#6	✓	4	✓			0.829	0.729	0.896	0.903	0.852	0.784	0.895	0.897
#7	✓	4	✓	✓	1	0.827	0.732	0.897	0.898	0.856	0.792	0.898	0.896
#8	✓	4	✓	✓	2	**0.840**	**0.745**	**0.909**	**0.921**	**0.860**	**0.795**	**0.903**	**0.903**
#9	✓	4	✓	✓	3	0.737	0.609	0.793	0.751	0.732	0.613	0.776	0.728

a large margin (max Dice: ~10%). On CVC-612-V and CVC-612-T, our *PNS-Net* consistently outperforms other SOTAs.

3.3   Ablation Study

Effectiveness of Channel Split. We investigate the contribution of channel split rule under different scales. The results are listed in rows #2 to #5 in Tab. 2. We observe that #4 (N=4) outperforms other settings (*i.e.*, #2, #3, and #5) on CVC-300-TV, in all metrics. This improvement shows that an improper receptive field (RF) harms the ability to excavate temporal information, since a large RF will pay more attention to the global environment rather than local motion information. On the other hand, when the split number is too small, the model fails to capture multi-scale polyps moving at various speeds.

Effectiveness of Soft-attention. We further investigate the contribution of the soft-attention mechanism. As shown in Tab. 2, #6 is generally better than #4 with the soft-attention block on CVC-612-T. This improvement suggests that introducing the soft-attention block to synthesize the aggregation feature and affinity matrix is necessary for increasing performance.

Effectiveness of the Number of NS Blocks. To access the number of normalized self-attention blocks under different settings, we derive three variants as #7, #8, and #9. We observe that #8 (*PNS-Net* setting) is significantly better than #7 and #9, with $R = 2$, in all metrics on CVC-300-TV and CVC-612-T. This improvement illustrates that too many iterations of NS blocks may cause overfitting on small datasets (#9). In contrast, the model fails to alleviate the diffusion issue of high-level features with a single residual block. Empirically, we recommend increasing the number of NS blocks when training on larger datasets.

4   Conclusion

We have proposed a self-attention based framework, *PNS-Net*, to accurately segment polyps from colonoscopy videos with super high speed (~140fps). Our basic normalized self-attention blocks can be easily plugged into existing CNN-based architectures. We experimentally show that our *PNS-Net* achieves the best performance on all existing publicly available datasets under six metrics. Further, extensive ablation studies demonstrate that the core components in our

PNS-Net are all effective. We hope that the proposed *PNS-Net* can serve as a catalyst for progressing both in VPS as well as other closely related video-based medical segmentation tasks. Exploring the performance of *PNS-Net* on a larger VPS dataset will be left to our future work.

References

1. Akbari, M., Mohrekesh, M., Nasr-Esfahani, E., Soroushmehr, S.R., Karimi, N., Samavi, S., Najarian, K.: Polyp segmentation in colonoscopy images using fully convolutional network. In: IEEE EMBC. pp. 69–72 (2018)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., Vilariño, F.: Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. CMIG **43**, 99–111 (2015)
4. Bernal, J., Sánchez, J., Vilarino, F.: Towards automatic polyp detection with a polyp appearance model. PR **45**(9), 3166–3182 (2012)
5. Brandao, P., Mazomenos, E., Ciuti, G., Caliò, R., Bianchi, F., Menciassi, A., Dario, P., Koulaouzidis, A., Arezzo, A., Stoyanov, D.: Fully convolutional neural networks for polyp segmentation in colonoscopy. In: MICAD. vol. 10134, p. 101340F (2017)
6. Fan, D.P., Cheng, M.M., Liu, Y., Li, T., Borji, A.: Structure-measure: A new way to evaluate foreground maps. In: IEEE ICCV. pp. 4548–4557 (2017)
7. Fan, D.P., Ji, G.P., Qin, X., Cheng, M.M.: Cognitive vision inspired object segmentation metric and loss function. SSI (2020)
8. Fan, D.P., Ji, G.P., Zhou, T., Chen, G., Fu, H., Shen, J., Shao, L.: Pranet: Parallel reverse attention network for polyp segmentation. In: MICCAI. pp. 263—-273 (2020)
9. Fang, Y., Chen, C., Yuan, Y., Tong, K.y.: Selective feature aggregation network with area-boundary constraints for polyp segmentation. In: MICCAI. pp. 302–310. Springer (2019)
10. Gu, Y., Wang, L., Wang, Z., Liu, Y., Cheng, M.M., Lu, S.P.: Pyramid constrained self-attention network for fast video salient object detection. In: AAAI. vol. 34, pp. 10869–10876 (2020)
11. Guo, L., Liu, J., Zhu, X., Yao, P., Lu, S., Lu, H.: Normalized and geometry-aware self-attention network for image captioning. In: IEEE CVPR. pp. 10327–10336 (2020)
12. Jha, D., Smedsrud, P.H., Riegler, M.A., Halvorsen, P., de Lange, T., Johansen, D., Johansen, H.D.: Kvasir-seg: A segmented polyp dataset. In: MMM. pp. 451–462. Springer (2020)
13. Jha, D., Smedsrud, P.H., Riegler, M.A., Johansen, D., De Lange, T., Halvorsen, P., Johansen, H.D.: Resunet++: An advanced architecture for medical image segmentation. In: IEEE ISM. pp. 225–2255 (2019)
14. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. NIPS **24**, 109–117 (2011)
15. Liu, S., Huang, D., et al.: Receptive field block net for accurate and fast object detection. In: ECCV. pp. 385–400 (2018)
16. Mamonov, A.V., Figueiredo, I.N., Figueiredo, P.N., Tsai, Y.H.R.: Automated polyp detection in colon capsule endoscopy. IEEE TMI **33**(7), 1488–1502 (2014)

17. Murugesan, B., Sarveswaran, K., Shankaranarayana, S.M., Ram, K., Joseph, J., Sivaprakasam, M.: Psi-Net: Shape and boundary aware joint multi-task deep network for medical image segmentation. In: IEEE EMBC. pp. 7223–7226 (2019)
18. Puyal, J.G.B., Bhatia, K.K., Brandao, P., Ahmad, O.F., Toth, D., Kader, R., Lovat, L., Mountney, P., Stoyanov, D.: Endoscopic polyp segmentation using a hybrid 2d/3d cnn. In: MICCAI. pp. 295–305. Springer (2020)
19. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)
20. Tajbakhsh, N., Gurudu, S.R., Liang, J.: Automated polyp detection in colonoscopy videos using shape and context information. IEEE TMI **35**(2), 630–644 (2015)
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS (2017)
22. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE CVPR. pp. 7794–7803 (2018)
23. Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: IEEE CVPR. pp. 5791–5800 (2020)
24. Yu, L., Chen, H., Dou, Q., Qin, J., Heng, P.A.: Integrating online and offline three-dimensional deep learning for automated polyp detection in colonoscopy videos. IEEE JBHI **21**(1), 65–75 (2016)
25. Zhang, R., Li, G., Li, Z., Cui, S., Qian, D., Yu, Y.: Adaptive context selection for polyp segmentation. In: MICCAI. pp. 253–262. Springer (2020)
26. Zhang, R., Zheng, Y., Poon, C.C., Shen, D., Lau, J.Y.: Polyp detection during colonoscopy using a regression-based convolutional neural network with a tracker. PR **83**, 209–219 (2018)
27. Zhong, J., Wang, W., Wu, H., Wen, Z., Qin, J.: Polypseg: An efficient context-aware network for polyp segmentation from colonoscopy videos. In: MICCAI. pp. 285–294. Springer (2020)
28. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. IEEE TMI pp. 3–11 (2019)