Supplemental Materials: Single-Image-Based Deep Learning for Segmentation of Early Esophageal Cancer Lesions

Haipeng Li*, *Student Member, IEEE,* Dingrui Liu*, Yu Zeng, Shuaicheng Liu, *Senior Member, IEEE,* Tao Gan, Nini Rao, Jinlin Yang, and Bing Zeng, *Fellow, IEEE*

I. Rendering from Single Lesion Image

To further visualize this data augmentation process, we present 3 examples, including 2 polyp segmentation cases and one EEC segmentation case (Fig. 2). Specifically, each example includes the original lesion image with the sketched ROI and sampled circles being pasted on it and three generated training datasets. Notable, the 2nd example shows the situation that multiple lesions exist.

II. Edge-Enhanced UNet

In this section, we illustrate the architecture details for our proposed eUNet, including the kernel size, stride, number of input/output channels, and input/output resolution. The encoder architecture is shown in Table III, the decoder architecture is illustrated in Table IV, the edge detector is detailed in Table V, and the details of BE Block are shown in Table VI.

III. Loss Function

After augmentating the single input lesion image $\boldsymbol{x}$, we feed the generated training set $\boldsymbol{X}$, the segmentation mask set $\boldsymbol{S}$, and the edge map set $\boldsymbol{E}$ to the learning process to train our eUNet. Let $\boldsymbol{I}$ and $\bar{\boldsymbol{I}}$ denote the regions within and outside the ROI, respectively. During the training, pixels in $\bar{\boldsymbol{I}}$ are counted fully since they all belong to the background (i.e., non-ROI). On the other hand, pixels in $\boldsymbol{I}$ have un-certainty: some belong to the lesion area and some do not. Therefore, pixels in $\boldsymbol{I}$ are ignored during the back propagation except that they come from the selected lesion seeds.

We jointly learn the segmentation result $\hat{\boldsymbol{s}}$ and two edge maps $\hat{\boldsymbol{e}}$ and $\hat{\boldsymbol{e}}'$ through the end-to-end network. The loss function consists of 3 parts as follows.

Segmentation Loss. To learn the segmentation output, a combined loss of the binary cross entropy (BCE) and Dice loss [1] is adopted:

$$\mathcal{L}_{\text{seg}}(\hat{\boldsymbol{s}}, \boldsymbol{s}) = \mu_1 \mathcal{L}_{BCE}(\hat{\boldsymbol{s}}, \boldsymbol{s}) + \mu_2 \mathcal{L}_{\text{Dice}}(\hat{\boldsymbol{s}}, \boldsymbol{s}), \tag{1}$$

where $\hat{\boldsymbol{s}}$ is the segmentation output, $\boldsymbol{s}$ is the ground-truth segmentation mask, and μ_1 and μ_2 are weights, which are empirically set to 0.8 and 0.2, respectively.

Edge Loss. As the proportion of boundary information in an edge map is very small, we first use a weighted cross entropy (WCE) loss [2] to handle the high imbalance between boundary and non-boundary pixels. For the i-th pixel, the loss is computed as:

$$\text{WCE}(\hat{\boldsymbol{e}}_{\boldsymbol{i}}, \boldsymbol{e}_{\boldsymbol{i}}) = \begin{cases} \alpha(1-\hat{e}_i)\log(1-\hat{e}_i) & \text{if } e_i = 0 \\ \beta \hat{e}_i \log \hat{e}_i & \text{if } e_i = 1 \end{cases}, \tag{2}$$

where $e_i = 0$ indicates an edge pixel, $e_i = 1$ corresponds to a non-edge pixel, and

$$\alpha = \frac{\sigma|\boldsymbol{Y}^+|}{|\boldsymbol{Y}^+| + |\boldsymbol{Y}^-|}, \quad \boldsymbol{\beta} = \frac{|\boldsymbol{Y}^-|}{|\boldsymbol{Y}^+| + |\boldsymbol{Y}^-|}. \tag{3}$$

Here, $\boldsymbol{Y}^+$ and $\boldsymbol{Y}^-$ denote the positive boundary set and the negative boundary set, respectively, σ is set to 1.2 empirically, $\hat{e}_i$ and e_i represent the output probability of the edge decoder and the GT edge, respectively.

Given the per-pixel WCE loss, the edge loss is then computed as:

$$\mathcal{L}_{\text{edge}}(\hat{\boldsymbol{e}}, \boldsymbol{e}) = \sum_i \text{WCE}(\hat{\boldsymbol{e}}_{\boldsymbol{i}}, \boldsymbol{e}_{\boldsymbol{i}}), \tag{4}$$

where $\hat{\boldsymbol{e}}$ and $\boldsymbol{e}$ represent the output of the edge detector and the ground-truth edge map, respectively.

Consistency Loss. To further enhance the edge information, an edge-consistency loss is proposed as follows:

$$\mathcal{L}_{\text{consist}}(\hat{\boldsymbol{e}}', \hat{\boldsymbol{e}}) = \text{WCE}(\hat{\boldsymbol{e}}', \wr(\hat{\boldsymbol{e}})), \tag{5}$$

where $\hat{\boldsymbol{e}}'$ is the output produced by the BE-block and $\wr(.)$ represents the binarization operation:

$$\wr(x) = \begin{cases} 0 & \text{if } 0 < x < 0.5 \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

* Equal contribution.

Manuscript submitted on August 19, 2023. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61720106004.

H. Li, D. Liu, S. Liu, and B. Zeng are with School of Information and Communication Engineering, Y. Zeng is with School of Glasgow College, N. Rao is with School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, Sichuan, China.

T. Gan and J. Yang are with West China Hospital, Sichuan University, Chengdu, Sichuan, China.

Corresponding author: Bing Zeng (eezeng@uestc.edu.cn)

Total Loss. Finally, the total loss function is constructed by a weighted sum of the three losses defined above:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{seg}}\left(\hat{s}, s\right) + \lambda_2 \mathcal{L}_{\text{edge}}\left(\hat{e}, e\right) + \lambda_3 \mathcal{L}_{\text{consist}}\left(\hat{e}', \hat{e}\right), \quad (7)$$

where λ_1, λ_2, and λ_3 are the weighting coefficients and are set to 1.0, 1.0, and 0.2, respectively.

IV. User Study

In this section, we illustrate the qualitative comparisons of the user study in Figs. 3, 4, and 5. As mentioned in our paper, this blind evaluation contains 30 samples from EEC-2022 and involves 4 participants who are all experienced endoscopists. In each figure shown here, the first row represents the original lesion images, the second row contains the ground-truth labels, and the last row shows our segmentation results. Note that this arrangement is different from what is described in the paper (original images in middle, GT labels and our results on the left and right randomly). We arrange them in this order just for the simplicity reason. Specifically, we use the check ✓ and the cross X to represent the voting results. For example, "✓✓XX" shown in one image indicates that the first two doctors prefer this result, but the last two doctors vote against it.

It is important to note that, when making decisions, doctors generally found the segmentation results to be correct, as evidenced in TABLE.III in the manuscript, where the number of evaluations favoring either one result as good or both as good totaled 117, in contrast to a mere 3 votes for both results being unsatisfactory. Nevertheless, they often relied on their own intuition to choose between the results presented to them. This intuition is difficult to quantify and express, which presents a challenge in our study. Consequently, our focus is on providing a detailed description of the experimental setup and procedures.

V. Qualitative Comparisons

In this section, we present more qualitative results on 3 datasets: CVC-ClinicDB, Kvasir, and EEC-2022. For each dataset, we select 10 different samples: the original input images and their ground-truth masks are placed in the first column, and the other columns show the segmentation results of various methods (the numerical number shown in the lower-left corner of each image represents the Dice score). These results are shown in Figs. 6, 7, 8, 9. Clearly, they further validate the effectiveness of our proposed method.

VI. Ablation Studies

Two extra ablation studies are conducted in this section. The first one focuses on the relationship between processing time and performance. In clinical applications, our goal is to compare to the conventional method of spraying agents (e.g., iodine) to highlight the lesion area. Therefore, our requirement for the algorithm is to minimize the processing time while keeping good performance. Here, the key factors include the size of the training dataset, the total training epochs, and the input image resolution. The results are shown in Table. I. We find that reducing the training set size, training epochs, or image resolution does decrease the segmentation performance, and reducing the training epochs seems to have a bigger impact.

	Data	Epoch	Resolution	Time	mDice
1/2 Trainset	0.8k	40	256x256	2.4 minutes	0.915
1/2 Epoch	1.6k	20	256x256	2.3 minutes	0.896
Reduce Size	1.6k	40	224x224	3.8 minutes	0.922
Ours	1.6k	40	256x256	4.5 minutes	**0.940**

TABLE I
COMPARISON WITH DIFFERENT TOTAL PROCESSING TIME.

	mean Dice	mean IoU	F_β^w	S_α	$E_\phi^{\max}$	MAE
Dilate 5 X 5	0.917	0.852	0.911	0.929	0.981	0.011
Dilate 11 X 11	0.906	0.834	0.890	0.918	0.974	0.013
Ours	**0.940**	**0.890**	**0.949**	**0.950**	**0.988**	**0.008**

TABLE II
COMPARISON WITH DIFFERENT ROI SIZES.

The second experiment aims to study the effect of different initial ROIs. To this end, we use the dilation in OpenCV to enlarge the initial ROI. Here, 2 different dilation masks are considered. The corresponding results are shown in Table. II. It is noted that dilation does decrease the segmentation performance. In clinical perspective, a dilation over the original ROI corresponds to what would be sketched by an unexperienced physician. Our results show that the involvement of experienced physicians plays an important role. Meanwhile, this involvement is rather simple and convenient. In particular, it can be done by an experienced physician at a distant site so as to promote the quality of medical treatment in underdeveloped regions.

VII. Generalizability of YOHO

As shown in Fig. 1, we have analyzed 2 different modality lesion images (polyp and EEC) taken at different times during the same examination for a single patient. For these images of polyp, we trained a YOHO on one particular image (case-31) and evaluated its performance on the others. For these images of EEC, we also trained a YOHO on one particular image (case-40) and evaluated its performance on the others. However, the YOHO model, when trained on this single image, failed to demonstrate good results across the remaining images, obtaining a mean Dice score of 0.6345 as compared to 0.9311 for polyp segmentation, and 0.8100 as compared to 0.8549 for EEC segmentation when we trained the model on each lesion image separately. From these results, we observe a significant decrease in accuracy as indicated in Fig. 1 (blue data indicates that YOHO is trained on one specific case and red data indicates that YOHO is trained on each corresponding lesion image). Based on these findings, we conclude that training a separate instance of the YOHO model for each unique lesion image is highly necessary, particularly when high accuracy is imperative to effectively assist doctors.

Moreover, as an extension association from the previous point. Given the fact we can augment k training data from each single image, we would like to evaluate the performance of a n $\times$ K training set generated for training from n input images. More specifically, we randomly select n=5 samples from the dataset and compared their performance when trained

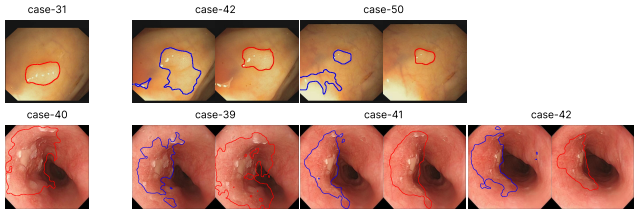

Fig. 1. The first row details the result of YOHO's polyp segmentation, where YOHO is trained using the "case-31" image and evaluated on two other images at different time (within a sequence), all pertaining to the same patient. Meanwhile, the second row illustrates the similar scenario for the EEC lesion segmentation, i.e., YOHO is trained with the "case-40" image and tested on 3 neighbouring cases. In this figure, the blue segmentation result denotes that YOHO is trained only using one specific case, whereas the red segmentation result signifies that YOHO receives training on each individual lesion image.

separately on individual sample (each sample trained an YOHO) versus a single YOHO encompassing all 5 samples. According to the results, the former one resulted in a mean Dice of 0.8318, while the latter one yielded a mean Dice of 0.7941. Based on these findings, we believe the "one image one network" strategy for YOHO is justified.

VIII. Applying RFSLI to existing methods

We believe that our YOHO framework is specifically designed to leverage the concept of "over-fitting" as an advantage, aiming to accomplish a "one-image-one-network" paradigm. However, we do recognize that when our framework is viewed as an augmentation step for existing supervised medical segmentation methods, a concern arises due to the scarcity of medical data, as it may inadvertently increase the training-testing performance gap.

To explore this further, we have conducted an experiment using PraNet [3] on the EEC dataset. In our method, the ground-truth segmentation mask identifies the region of interest (ROI), which consists solely of lesion areas, and we augment these by randomly sampling additional lesion points. Each training pair is then augmented with 80 synthetic images, effectively increasing the training set size by 80-fold. The results of this experiment reveal a significant degradation in PraNet's performance (mean Dice from 0.683 to 0.557), which aligns with our hypothesis.

However, despite these results, we posit that such augmentation methods might be more effective in domains where copious data is available. Therefore, investigating these methods' applicability across a broader data pool constitutes a key objective for our future research endeavors.

IX. Future Work

In our future work, we will implement our method on the original resolution. For example, we will maintain the original 384×288 in CVC-612 and 480×480 in EEC-2022. In this way, we believe that the segmentation performance of our method would be further increased. We are aware that the training time will increase while working on a bigger resolution. However, a clinical scenario usually consists of two stages: the diagnosis stage and the treatment/surgery stage. At the diagnosis stage, we have much more time so that the training on the original resolution (without any down-sizing) becomes feasible. The segmentation result can greatly help the formulation of a treatment/surgery plan. At the same time, this well-trained network can serve as an excellent start at the treatment/surgery stage where the same patient is involved.

References

[1] F. Milletari, N. Navab, and S.-A. Ahmadi, "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. 1

[2] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. CVPR*, pp. 3000–3009, 2017. 1

[3] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Pranet: Parallel reverse attention network for polyp segmentation," in *International conference on medical image computing and computer-assisted intervention*, pp. 263–273, Springer, 2020. 3

Name	Kernel	Str.	Ch I/O	InpRes	OutRes	Input
conv1	7×7	2	$3/64$	$H\times W$	$H\times W$	Images
maxpooling	3×3	2	$64/64$	$H\times W$	$H/2\times W/2$	conv1
conv2a	3×3	1	$64/64$	$H/2\times W/2$	$H/2\times W/2$	maxpooling
conv2b	3×3	1	$64/64$	$H/2\times W/2$	$H/2\times W/2$	conv2a
conv2c	3×3	1	$64/64$	$H/2\times W/2$	$H/2\times W/2$	conv2b
conv2d	3×3	1	$64/64$	$H/2\times W/2$	$H/2\times W/2$	conv2c
conv2e	3×3	1	$64/64$	$H/2\times W/2$	$H/2\times W/2$	conv2d
conv2f	3×3	1	$64/64$	$H/2\times W/2$	$H/2\times W/2$	conv2e
conv3a	3×3	2	$64/128$	$H/2\times W/2$	$H/4\times W/4$	conv2f
conv3b	3×3	1	$128/128$	$H/4\times W/4$	$H/4\times W/4$	conv3a
conv3c	3×3	1	$128/128$	$H/4\times W/4$	$H/4\times W/4$	conv3b
conv3d	3×3	1	$128/128$	$H/4\times W/4$	$H/4\times W/4$	conv3c
conv3e	3×3	1	$64/128$	$H/2\times W/2$	$H/4\times W/4$	conv3d
conv3f	3×3	1	$128/128$	$H/4\times W/4$	$H/4\times W/4$	conv3e
conv3g	3×3	1	$128/128$	$H/4\times W/4$	$H/4\times W/4$	conv3f
conv3h	3×3	1	$128/128$	$H/4\times W/4$	$H/4\times W/4$	conv3g
conv4a	3×3	2	$128/256$	$H/4\times W/4$	$H/8\times W/8$	conv3h
conv4b	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4a
conv4c	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4b
conv4d	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4c
conv4e	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4d
conv4f	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4e
conv4g	3×3	1	$128/256$	$H/4\times W/4$	$H/8\times W/8$	conv4f
conv4h	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4g
conv4i	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4h
conv4j	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4i
conv4k	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4j
conv4l	3×3	1	$256/256$	$H/8\times W/8$	$H/8\times W/8$	conv4k
conv5a	3×3	2	$256/512$	$H/8\times W/8$	$H/16\times W/16$	conv4l
conv5b	3×3	1	$512/512$	$H/16\times W/16$	$H/16\times W/16$	conv5a
conv5c	3×3	1	$512/512$	$H/16\times W/16$	$H/16\times W/16$	conv5b
conv5d	3×3	1	$256/512$	$H/8\times W/8$	$H/16\times W/16$	conv5c
conv5e	3×3	1	$512/512$	$H/16\times W/16$	$H/16\times W/16$	conv5d
conv5f	3×3	1	$512/512$	$H/16\times W/16$	$H/16\times W/16$	conv5e

TABLE III
DETAIL ARCHITECTURE OF ENCODER.

Name	Kernel	Str.	Ch I/O	InpRes	OutRes	Input
upconv4a	3×3	1	$768/256$	$H/16\times W/16$	$H/16\times W/16$	conv5f+conv4l
upconv4b	3×3	1	$256/256$	$H/16\times W/16$	$H/16\times W/16$	upconv4a
upconv3a	3×3	1	$384/128$	$H/16\times W/16$	$H/8\times W/8$	upconv4b+conv3h
upconv3b	3×3	1	$128/128$	$H/8\times W/8$	$H/8\times W/8$	upconv3a
upconv2a	3×3	1	$192/64$	$H/8\times W/8$	$H/4\times W/4$	upconv3b+conv2f
upconv2b	3×3	1	$64/64$	$H/4\times W/4$	$H/4\times W/4$	upconv2a
upconv1a	3×3	1	$64/32$	$H/4\times W/4$	$H\times W$	upconv2b+conv1
upconv1b	3×3	1	$32/1$	$H/2\times W/2$	$H\times W$	upconv1a

TABLE IV
DETAIL ARCHITECTURE OF DECODER.

Name	Kernel	Str.	Ch I/O	InpRes	OutRes	Input
ED-conv1	1×1	1	$64/1$	$H/2\times W/2$	$H\times W$	conv1
ED-conv2	1×1	1	$64/1$	$H/4\times W/4$	$H\times W$	conv2f
ED-conv3	1×1	1	$128/1$	$H/8\times W/8$	$H\times W$	conv3h
ED-conv4	1×1	1	$256/1$	$H/16\times W/16$	$H\times W$	conv4l
ED-conv5	1×1	1	$512/1$	$H/16\times W/16$	$H\times W$	conv5f
EA-conv1a	3×3	1	$5/64$	$H\times W$	$H\times W$	ED-conv1+ED-conv2+ED-conv3+ED-conv4+ED-conv5
EA-conv1b	3×3	1	$64/64$	$H\times W$	$H\times W$	EA-conv1a
EA-conv1c	3×3	1	$64/1$	$H\times W$	$H\times W$	EA-conv1b

TABLE V
DETAIL ARCHITECTURE OF EDGE DETECTOR.

Name	Kernel	Str.	Ch I/O	InpRes	OutRes	Input
Laplace $Conv*$	3×3	1	$1/1$	$H\times W$	$H\times W$	upconv1b
BE-upconv3	1×1	1	$1/16$	$H\times W$	$H\times W$	Laplace Conv
BE-upconv2	3×3	1	$16/16$	$H\times W$	$H\times W$	BE-upconv3
BE-upconv1	1×1	1	$16/1$	$H\times W$	$H\times W$	BE-upconv2

TABLE VI
DETAIL ARCHITECTURE OF BE BLOCK. NOTABLE, THE LAPLACE CONV:[-1,-1,-1,-1,-8,-1,-1,-1,-1], DOES NOT ENGAGE IN LEARNING.

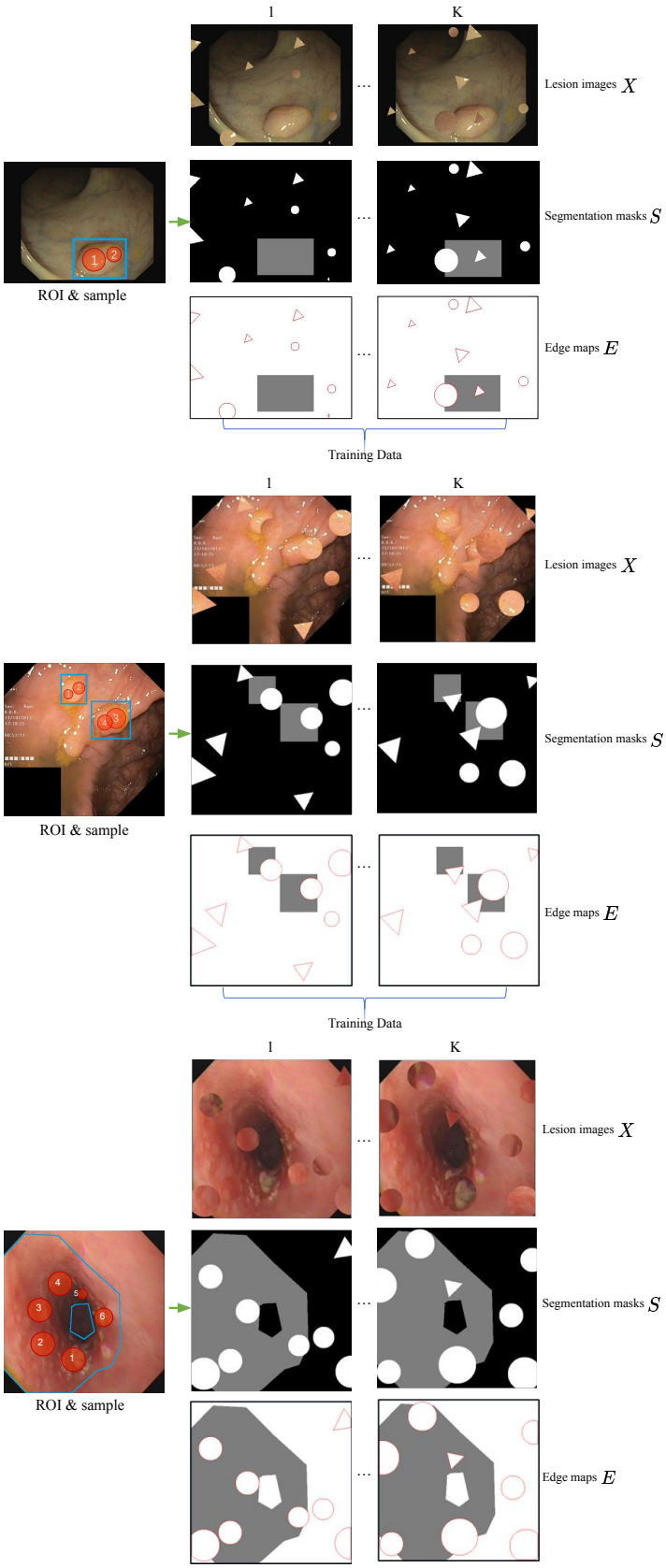

Fig. 2. Data augmentation results on CVC-ClinicDB, Kvasir, and EEC-2022.

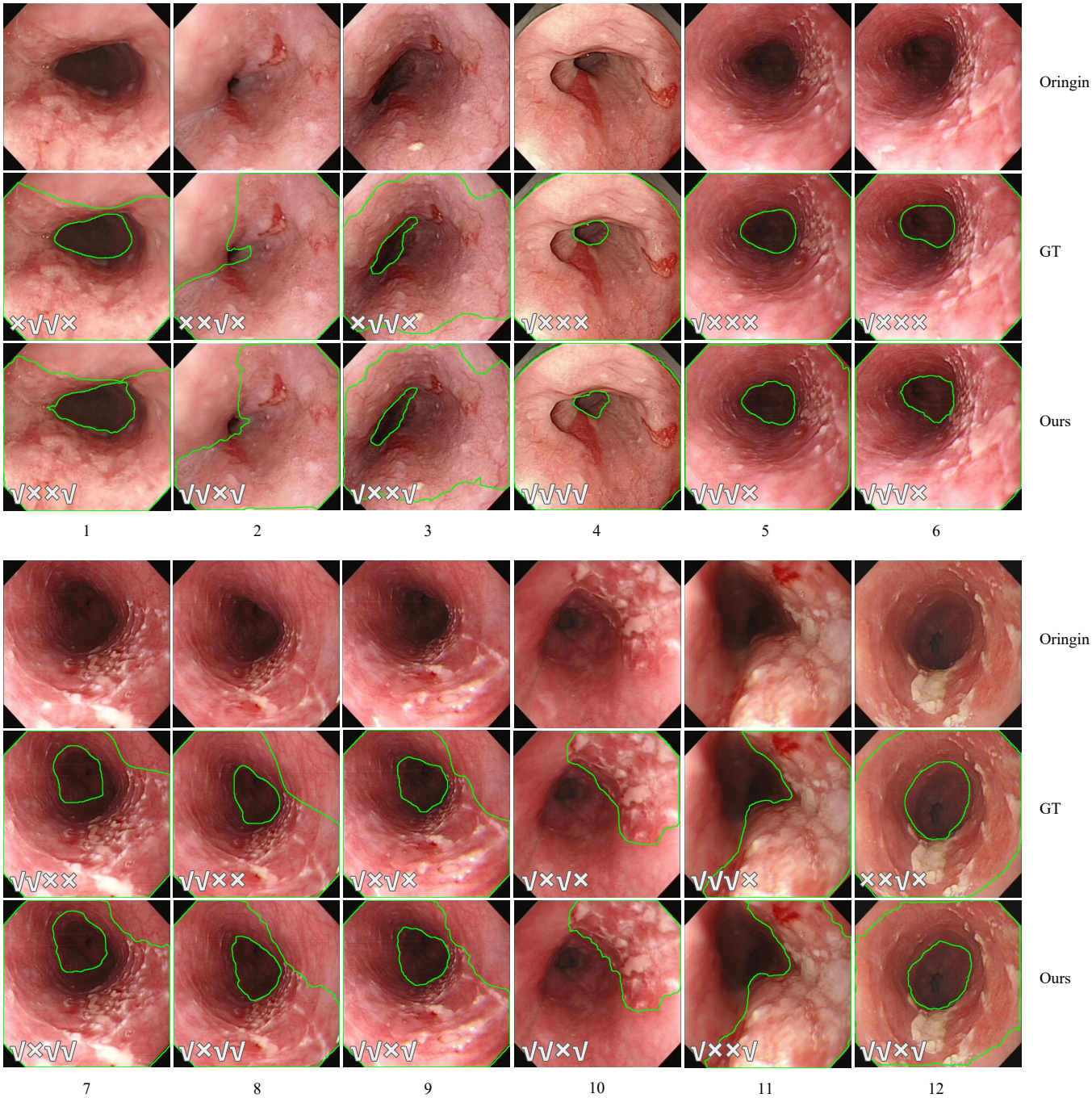

Fig. 3. Blind evaluations (1st - 12th case) by Doctors.

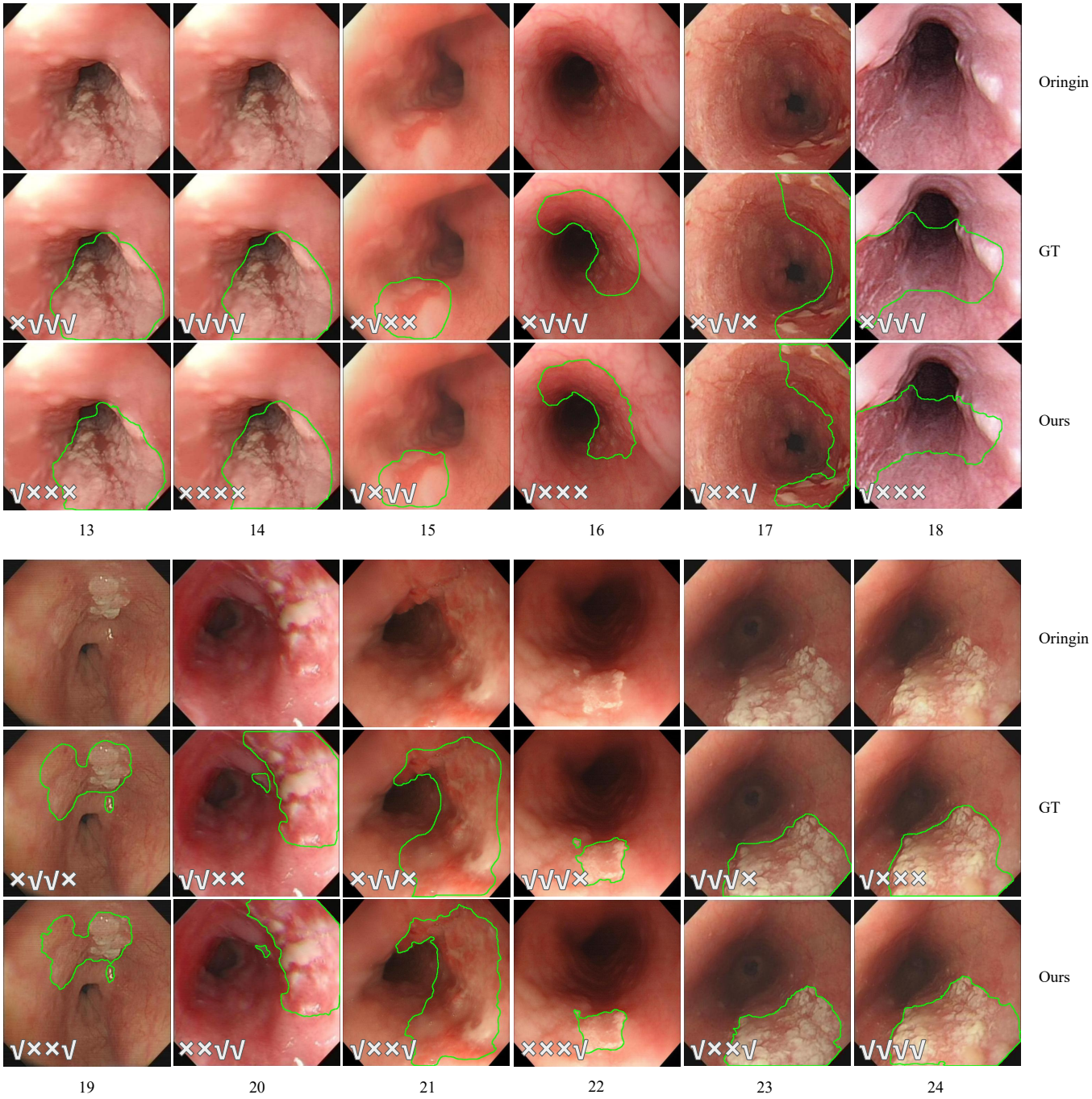

Fig. 4. Blind evaluations (13th - 25th case) by Doctors.

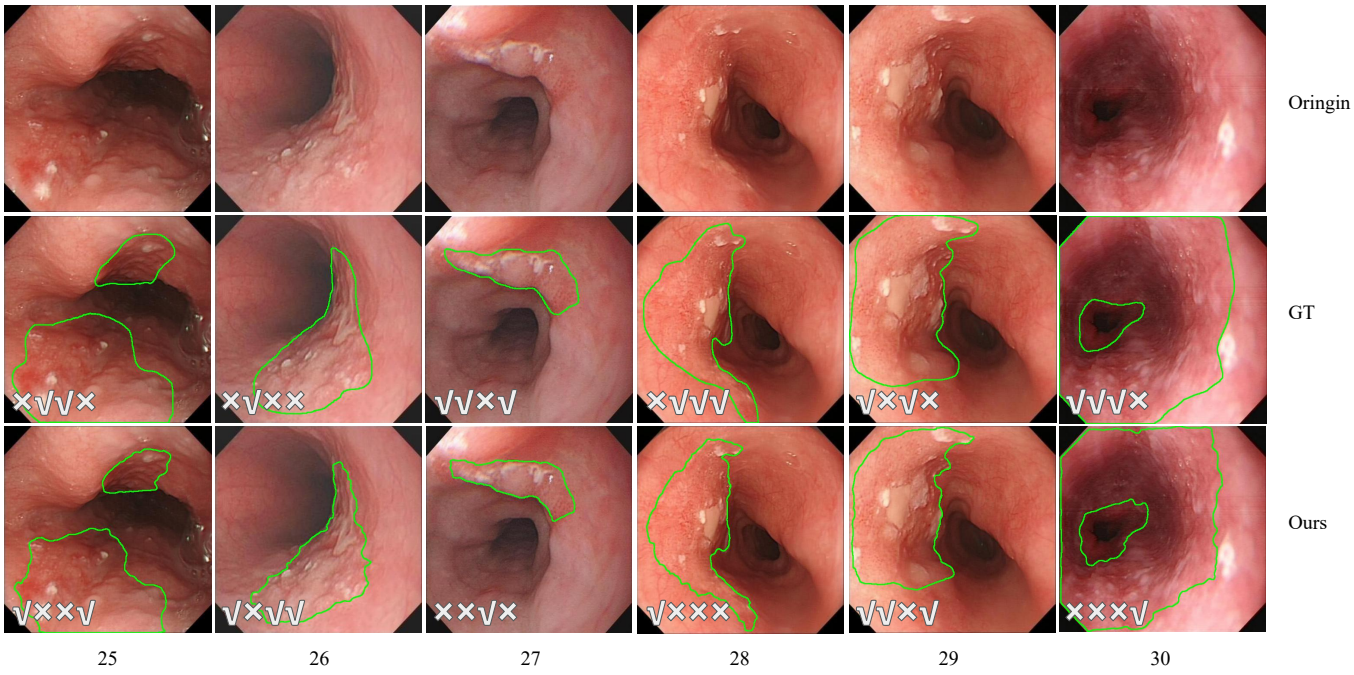

Fig. 5. Blind evaluations (25th - 30th case) by Doctors.

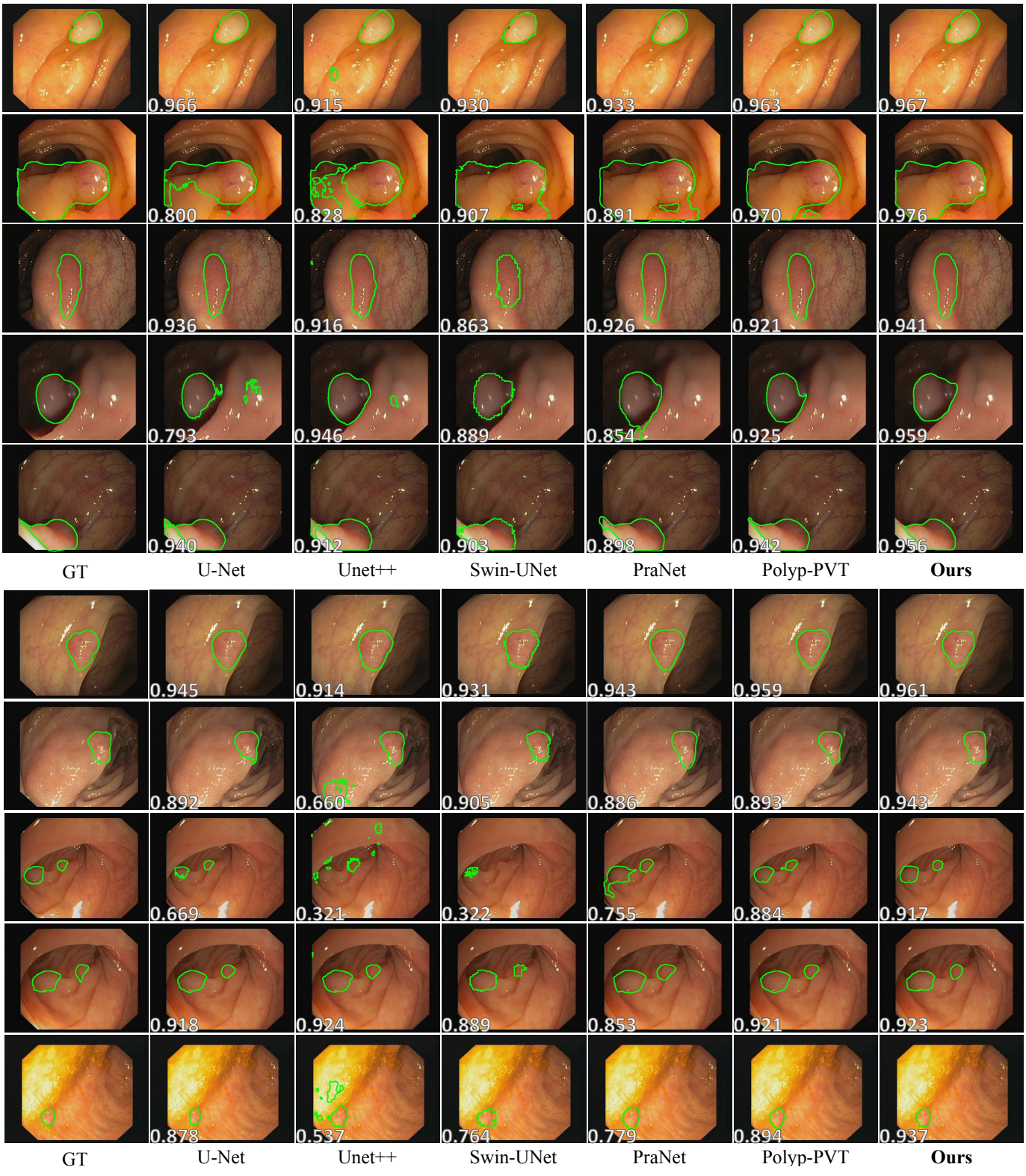

Fig. 6. Some segmentation results of different methods on CVC-612.

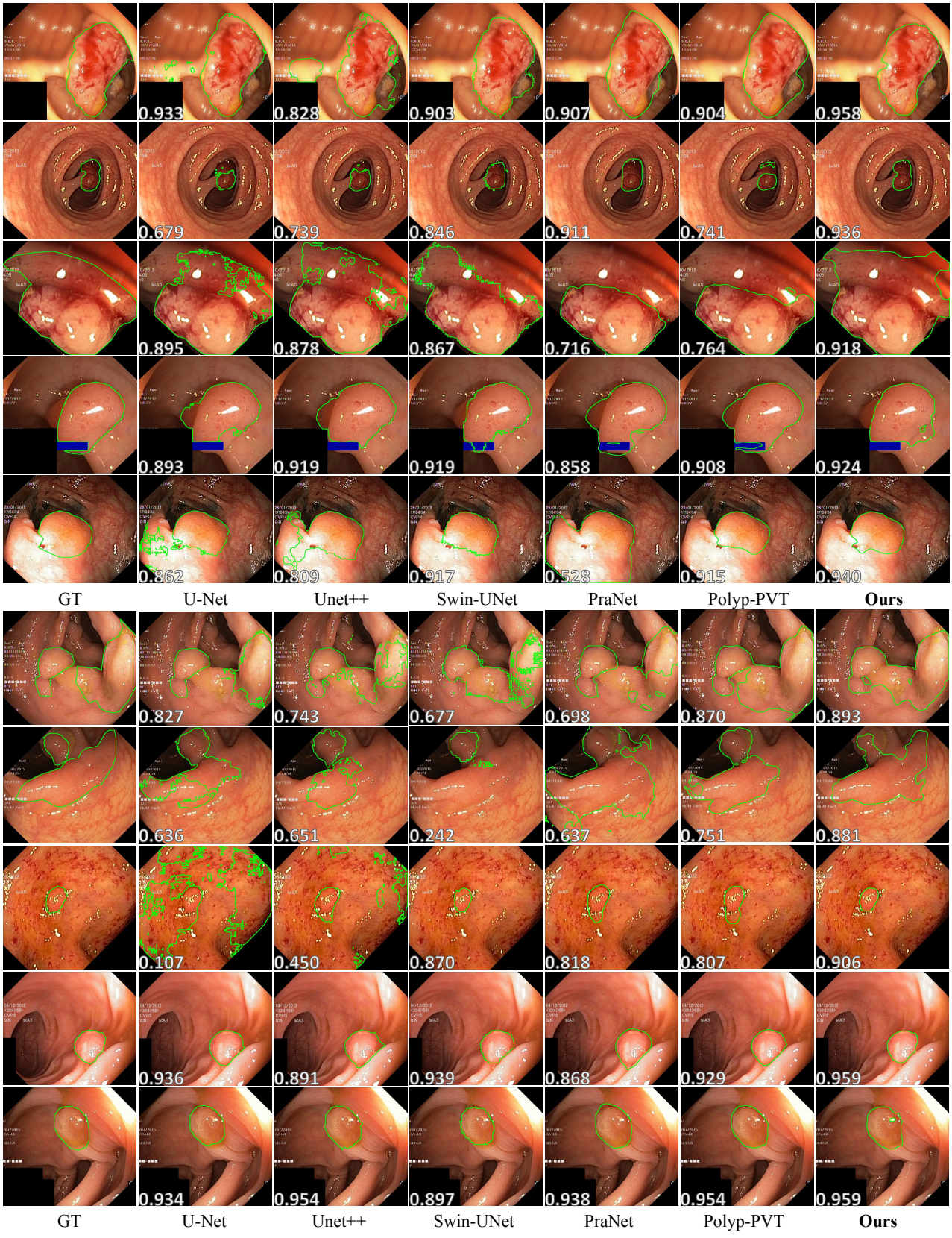

Fig. 7. Some segmentation results of different methods on Kvasir.

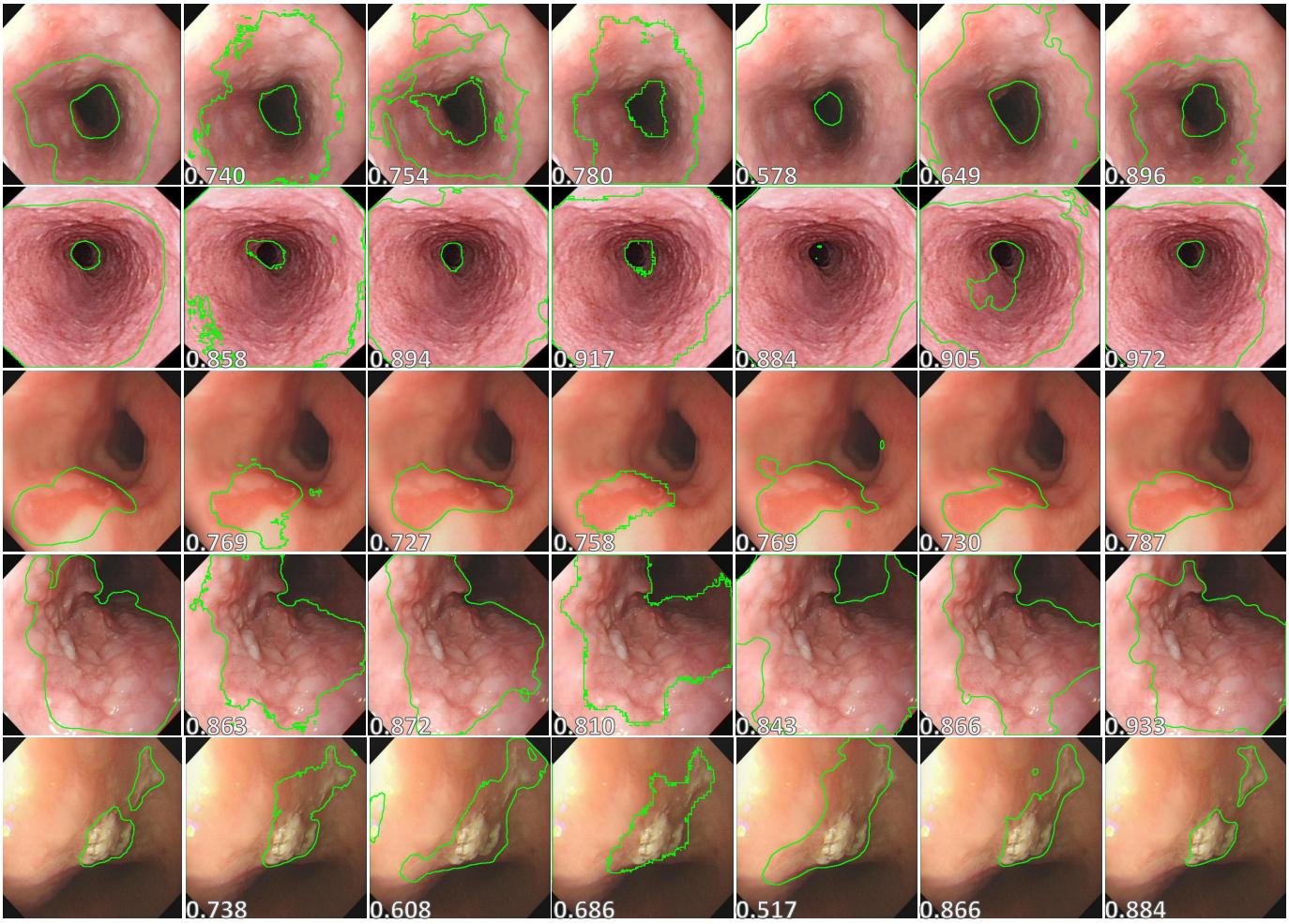

Fig. 8. Some segmentation results of different methods on EEC-2022.

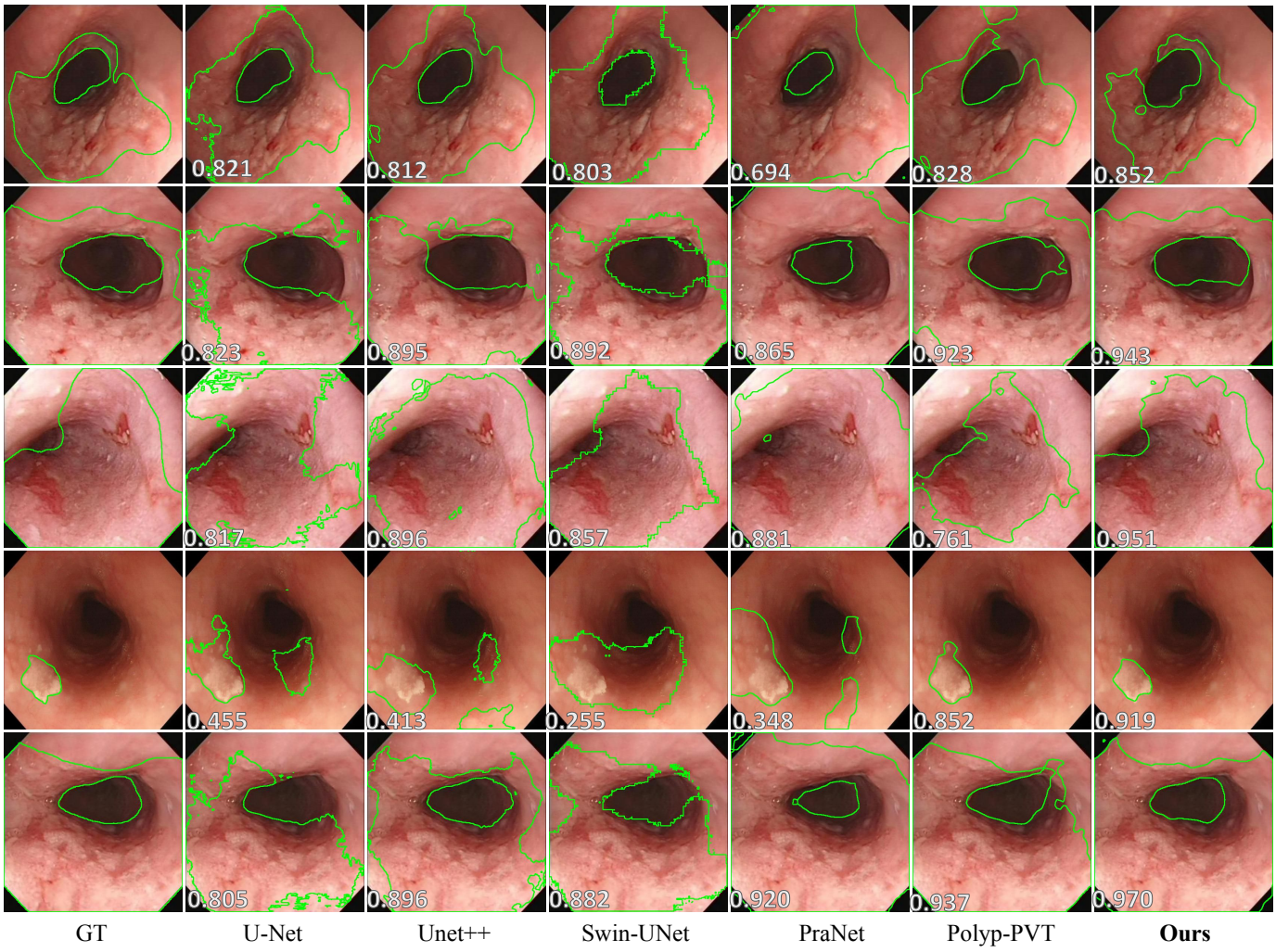

Fig. 9. Some segmentation results of different methods on EEC-2022.