

Generative Adversarial Networks for Automatic Polyp Segmentation

Awadelrahman M. A. Ahmed
University of Oslo, Norway; aahmed@ifi.uio.no

ABSTRACT

This paper aims to contribute in bench-marking the automatic polyp segmentation problem using generative adversarial networks framework. Perceiving the problem as an image-to-image translation task, conditional generative adversarial networks are utilized to generate masks conditioned by the images as inputs. Both generator and discriminator are convolution neural networks based. The model achieved 0.4382 on Jaccard index and 0.611 as F2 score.

1 INTRODUCTION

Developing an automated computer-aided diagnosis system for polyp segmentation is indeed one of the potential solutions that can assist colonoscopy, the current gold-standard medical procedure for examining the colon, in the sense that can lessen the percentage of the overlooked polyps. Motivated by that, this paper contributes by examining a model based on generative adversarial networks (GANs) to represent one of the benchmark methods that other cutting edge new methods can be compared to. This paper discusses a submission to *Medico automatic polyp segmentation challenge* [3] for task 1 which asks the participants to develop algorithms for segmenting polyps on a comprehensive data set. In this work we perceive the polyp segmentation problem as an image-to-image translation problem, as we are given a gastrointestinal polyp image and the task is to translate it to the corresponding mask that locates the polyps. GANs framework have been successfully implemented to solve image-to-image translation problem in many application fields. Authors in [2] investigated conditional GANs as a general-purpose solution to image-to-image translation problems and evaluated in different application fields such as translating aerial images to maps and reconstructing objects from edge maps or templates. The challenge task can be seen in the same way, hence we adapted the model architecture suggested in [2] to fit the polyp segmentation problem. The following sections will illustrate the model details, evaluation and results followed by the conclusion and future work.

2 GANS FOR POLYP SEGMENTATION

The GANs framework is a generative model scheme based on a game-theoretic formulation for training data-synthesis models [1]. A GAN consists of two models (e.g., neural networks) trained in opposition to each other. A generator network G takes a random noise vector as an input and maps it to an output which represents fake data. The other network is the discriminator D , which receives that fake data and classifies it as fake data (i.e., generated by G) or as real data from a real dataset. Nevertheless, our particular task is to

map each gastrointestinal polyp image to the corresponding mask that defines the polyps area, this description does not fit within the standard GAN setting described above where the input is a noise and the output is an arbitrary (but realistic-looking) image. Therefore, for that we utilize a prominent variation of GANs which produces an output that is *conditioned* by its input and hence called *conditional GAN*.

The polyp segmentation GAN-based model consists of two networks. A generator takes the images as input and tries to produce realistic-looking masks conditioned by this input, and a discriminator which is basically a classifier which has the access to the ground truth masks and tries to classify whether the generated masks are real or not. To stabilize the training, the images are concatenated with the masks (generated or real) before being fed to the discriminator. The ultimate goal is to find the generator parameter set θ_G^* that satisfies the minimax optimization problem is shown in (1). We particularly draw a batch of images x and their corresponding masks y from their distributions X and Y respectively. We feed x to the generator G to produce a fake mask set $G(x)$. We then concatenate both masks (real and fake) with the input images, this is represented by C . These concatenated pairs are then fed to the discriminator D . To learn these models, both networks are updated in an adversarial fashion based on the discriminator output; meaning that the discriminator aims to maximize the exact function that the generator aims to minimize. Through training time, the generator will be able to generate realistic-looking masks that are good enough to deceive the discriminator, i.e. it can classify as real.

$$\theta_G^* = \min_{\theta_G} \max_{\theta_D} \mathbb{E}_{x,y \sim X,Y} [\log D(C(x,y))] + \mathbb{E}_{x \sim X} [\log(1 - D(C(x,G(x))))] \quad (1)$$

The model block diagram is shown in Figure 1 and the model details are shown in Table 1. Both networks are based on convolution neural networks (CNNs). The generator has two segments, one is based on convolution operations followed by deconvolution layers with using skip connections.

3 EXPERIMENTAL EVALUATION

As required by the challenge, the data set used is the Kvaris-SEG [4] training data set which consists of 1000 image-mask pairs. We firstly split the dataset to 800 training set and 200 validation set to optimize the parameters. After we got satisfied with the model, we re-trained the model on the whole 1000 data pairs and used to produce the masks of a separate 160 test set that we submitted. The model is implemented using Python[6] and PyTorch[5] framework on a 2×14-core Intel/128Gib machine.

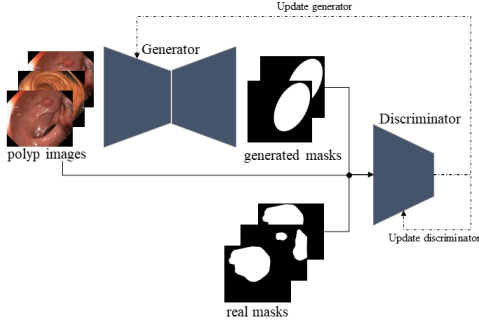


Figure 1: Model block diagram

Generator Network		
Conv1	f^*1	LeakyReLU
Conv2	f^*2	LeakyReLU
Conv3	f^*4	LeakyReLU
Conv4, 5, 6, 7, 8	f^*8	LeakyReLU
DeConv1, 2, 3, 4	f^*8	ReLU, Dropout(0.5)
DeConv5	f^*4	ReLU, Dropout(0.5)
DeConv6	f^*2	ReLU, Dropout(0.5)
DeConv7	f^*1	ReLU, Dropout(0.5)
DeConv8	1	Tanh
Discriminator Network		
Conv1	f^*1	LeakyReLU
Conv2	f^*2	LeakyReLU
Conv3	f^*4	LeakyReLU
Conv4	1	Sigmoid

f : number of filters=64

Table 1: Model details

3.1 Data Preparation and Training

As the images have different dimensions, we fit the images to the mean width and height by cropping the larger images and padding the smaller images with zeros. The pixel values are normalized between -1 and 1. No data augmentation were used. The networks in Table 1 were trained by optimizing the loss function in equation 1. The discriminator minimizes the negative log likelihood and the generator minimizes the negative of that. We tried other options like using feature matching, i.e. minimizing the loss between the features of the discriminator pre-last layer and the original masks instead of using the last classification layer output, but this seems to hurt the performance. We used Adam optimizer for both networks with learning rates of 0.002, for 12 epochs and batch size of 4. Some samples of the generated masks for some epochs during training is shown in Figure 2.



Figure 2: Generated masks in different epochs

3.2 Results

To quantify the overlap percentage between the ground truth mask and our generated masks, we report both Jaccard index the Dice similarity coefficient (DSC). We also report the per pixel recall, precision, accuracy and F2 (giving more weight to recall). Results on the test set are in Table 2. However it is difficult to comment on those results objectively because of the uniqueness of the dataset [4], i.e. no previous publications to compare with, but in general the higher recall than precision reflects the model accountability for false-negatives which is desirable for this application. The test throughput is 16 frames/sec on a 2×14-core Intel/128Gib machine.

We report the model outputs for some samples in Figure 3. Even though we do not have the access to the ground truth of the test set, we may observe that the model incorrectly identified the polyp location of the bottom two samples; whereas it did far better in locating the polyps area of other samples (top and middle rows). We do not have a clear explanation for that, but we speculate that the small receptive field of the convolution layers could be a reason. In other words, the convolution layers pay attention to the close by area to each pixel and if this area is rich enough with features (e.g. has sharp edges) the polyp can be more distinguishable. This why incorporating attention mechanism [7] might help the model to attend to fine details and larger ranges; hence we suggest studying it as an extended model.

Jaccard	DSC	Recall	Precision	Accuracy	F2
0.4382	0.562	0.697	0.556	0.881	0.611

Table 2: Test results as reported by the challenge organizers

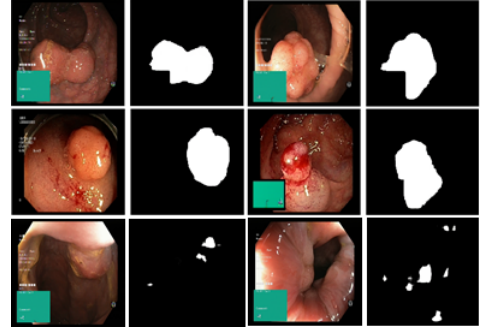


Figure 3: Model outputs from the test set

4 CONCLUSION

This paper aimed to benchmark the polyp segmentation problem using GANs. The problem has been perceived as an image-to-image translation task and we utilized conditional GANs architecture. The model was able to learn the masks however higher performance can be achieved by trying some improvements such as adding reconstruction loss and increasing the dataset with data augmentation. An interesting extension for this model is to try incorporating an attention layer [7] which can help the convolution layers in both generator and discriminator to attend to fine details and expand the receptive field.

REFERENCES

- [1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
- [2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.
- [3] Debesh Jha, Steven A. Hicks, Krister Emanuelsen, Håvard Johansen, Dag Johansen, Thomas de Lange, Michael A. Riegler, and Pål Halvorsen. 14-15 December 2020. Medico Multimedia Task at MediaEval 2020: Automatic Polyp Segmentation. In *Proc. of the MediaEval 2020 Workshop, Online*.
- [4] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D Johansen. 2020. Kvasir-SEG: A segmented polyp dataset. In *Proc. of International Conference on Multimedia Modeling*. 451–462.
- [5] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).
- [6] Guido Van Rossum and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
- [7] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019. Self-attention generative adversarial networks. In *International conference on machine learning*. PMLR, 7354–7363.