SinGAN-Seg: Synthetic Training Data Generation for Medical Image Segmentation

Vajira Thambawita[1,2], Pegah Salehi[1], Sajad Amouei Sheshkal[1], Steven A. Hicks[1,2], Hugo L. Hammer[2], Sravanthi Parasa[4], Thomas de Lange[3], Pål Halvorsen[1], and Michael A. Riegler[1]

[1] *SimulaMet, Oslo, Norway*
[2] *Oslo Metropolitan University, Oslo, Norway*
[3] *Department of Medical Research, Bærum Hospital, Gjettum, Norway*
[4] *Department of Gastroenterology, Swedish Medical Group, Seattle, WA, USA*

Abstract

Processing medical data to find abnormalities is a time-consuming and costly task, requiring tremendous efforts from medical experts. Therefore, artificial intelligence (AI) has become a popular tool for the automatic processing of medical data, acting as a supportive tool for doctors. AI tools highly depend on data for training the models. However, there are several constraints to access to large amounts of medical data to train machine learning algorithms in the medical domain, e.g., due to privacy concerns and the costly, time-consuming medical data annotation process.

To address this, in this paper we present a novel synthetic data generation pipeline called *SinGAN-Seg* to produce synthetic medical data with the corresponding annotated ground truth masks. We show that these synthetic data generation pipelines can be used as an alternative to bypass privacy concerns and as an alternative way to produce artificial segmentation datasets with corresponding ground truth masks to avoid the tedious medical data annotation process. As a proof of concept, we used an open polyp segmentation dataset. By training UNet++ using both real polyp segmentation dataset and the corresponding synthetic dataset generated from the SinGAN-Seg pipeline, we show that the synthetic data can achieve a very close performance to the real data when the real segmentation datasets are large enough. In addition, we show that synthetic data generated from the SinGAN-Seg pipeline improving the performance of segmentation algorithms when the training dataset is very small. Since our SinGAN-Seg pipeline is applicable for any medical dataset, this pipeline can be used with any other segmentation datasets.

1 Introduction

AI has become a popular tool in medicine and has been vastly discussed in recent decades to augment performance of clinicians [1, 2, 3, 4]. According to the statistics discussed by Jiang et al. [1], artificial neural networks (ANNs) [5] and support vector machines (SVMs) [6] are the most popular machine learning (ML) algorithms used with medical data. These ML models learn from data; thus the medical data have a direct influence on the success of ML solutions in real applications. While the SVM algorithms are popular within regression [7, 8] and classification [9] tasks, ANNs or deep neural networks (DNNs) are used widely for all the types; regression, classification, detection and segmentation.

A segmentation model makes more advanced predictions than regression, classification, and detection as it performs pixel-wise classification of the input images. Therefore, medical image segmentation is

a popular application of AI in medicine, so it is used more widely with different kinds of medical image data [10, 11, 12]. Polyp segmentation is one of popular segmentation tasks that uses ML techniques to detect and segment polyps in images/videos collected from gastrointestinal tract (GI) screenings. Early identification of polyps in GI tract is critical to prevent colorectal cancers [13]. Therefore, many ML models have been investigated to segment polyps automatically in GI tract videos recorded from endoscopy [14, 15, 16] or PilCams examinations [17, 18, 19] to augment performance of doctors by detecting polyps missed by experts, thereby both decreasing the miss rates and reducing the observer variations.

Most of polyp segmentation models are based on convolutional neural networks (CNNs) and are trained using publicly available polyp segmentation datasets [20, 21, 22, 23, 24]. However, these datasets have a limited number of images with corresponding expert annotated masks. For examples, the CVC-VideoClinicDB [21] dataset has $11,954$ images from 10 polyp videos and 10 non-polyp videos, the PICCOLO dataset [24] has $3,433$ manually annotated images ($2,131$ white-light images and $1,302$ narrow-band images), and the Hyper-Kvasir [20] dataset has only $1,000$ segmented images, but also contains of $100,000$ unlabeled images.

We identified two main reasons for having small datasets in medical domain compared to other domains. The first reason is privacy concerns attached with medical data, and the second is the costly and time-consuming medical data annotation processes that the medical domain experts must perform.

The privacy concerns can vary from country to country and region to region according to data protection regulations introduced in the specific areas. For example, Norway should follow the rules given by the Norwegian data protection authority (NDPA) [25] and enforce the personal data act [26] in addition to following the general data protection regulation (GDPR) [27] guidelines being the same for all European countries. While there is no central level privacy protection guideline in the US like GDPR in Europe, US rules and regulations are enforced through other US privacy laws, such as Health Insurance Portability and Accountability Act (HIPAA) [28] and California Consumer Privacy Act (CCPA) [29]. In Asian counties, they follow their own sets of rules, such as Japan's Act on Protection of Personal Information [30], the South Korean Personal Information Protection Commission [31] and the Personal Data Protection Bill in India [32].

If research is performed with such privacy restrictions, the papers published are often theoretical methods only. According to the analyzed medical image segmentation studies in [33], 30% have used private datasets. As a result, the studies are not reproducible. Researchers must keep datasets private due to medical data sharing restrictions. Furthermore, universities and research institutes that use medical domain data for teaching purposes use the same medical datasets for years, which affects the quality of education. In addition to the privacy concerns, the costly and time-consuming medical data labeling and annotation process [34] is an obstacle to producing big datasets for AI algorithms. Compared to other already time-consuming medical data labeling processes, a pixel-wise data annotation are far more demanding on the valuable medical experts' time. The experts in the medical domain can perform the annotations fully trustable in terms of correctness. If the data annotations by experts are not possible, the experts should do at least a review process to make the annotations correct before using them in AI algorithms. The importance of having accurate annotations from experts for medical data is, for example, discussed by Yu et al. [35] using a mandible segmentation dataset of CT images. In this regard, researching a way to produce synthetic segmentation datasets is important to overcome the timely and costly medical data annotation process. Therefore, researching an alternative way for medical data sharing, bypassing both the privacy and time-consuming dataset generation challenges, is the main objective of this study.

In this regard, the contributions of this paper are as follows.

- This study introduces the novel SynGAN-Seg pipeline to generate synthetic medical image and its corresponding segmentation mask using a modified version of the state-of-the-art SinGAN

architecture with a fine-tuning step using a style-transfer method. We use polyp segmentation as a case study, the SinGAN-Seg can be applied for all types of segmentation tasks.

- We have published the biggest synthetic polyp dataset and the corresponding masks at `https://osf.io/xrgz8/`. Moreover, we have published our generators as a python package at Python package index (PyPI) (`https://pypi.org/project/singan-seg-polyp/`) to generate an unlimited number of polyps and corresponding mask images as needed. To the best of our knowledge, this is the first publicly available synthetic polyp dataset and the corresponding generative functions as a PyPI package.
- We show that synthetic images and corresponding mask images can improve the segmentation performance when the size of a training dataset is limited.

2 Method

In the pipeline of SinGAN-Seg, there are as depicted in Figure 1 two main steps: (1) training novel SinGAN-Seg generative models and (2) style transferring. The first step generates synthetic polyp images and corresponding binary segmentation masks representing the polyp area. The novel four channels SinGAN-Seg, based on the vanilla SinGAN architecture [36], is introduced in this first step. The novel training process of four channels SinGAN-Seg models is presented in this step. Using a single SinGAN-Seg model, we can generate multiple synthetic images and masks from a single real image and the corresponding masks. Therefore this generation process can be identified as $1 : N$ generations, and it is denoted using $[img]_N$, where N represents the number of samples generated in the figure. The second step focuses on transferring styles such as features of polyps' texture from real images into the corresponding generated synthetic images. This second step is depicted in the Step 2 in Figure 1.

SinGAN-Seg is a modified version of SinGAN [36] which was designed to generate synthetic data from a generative adversarial network (GAN) trained only using a single image. The original SinGAN is trained using different scales of the same input image, the so-called image pyramid. This image pyramid is a set of images of different resolutions of a single image from low resolution to high resolution. SinGAN consists of a GAN pyramid, which takes the corresponding image pyramid. In this study, we build on the implementation and the training process used in SinGAN, except for the number of input and output channels. The original SinGAN implementation [36] uses a three-channel RGB image as the input and produces a three-channel RGB image as the output. However, our SinGAN-Seg uses four-channels images as the input and the output. The four-channels image consist of the input RGB image and the single channel ground truth mask by stacking them together as depicted in the SinGAN-Seg model in Figure 1. The main purpose of this modification is to generate four-channels synthetic output, which consists of a synthetic image and the corresponding ground truth mask.

In the second step of the SinGAN-Seg pipeline, we fine-tune the output of the four channels SinGAN-Seg model using the style-transfer method introduced by Leon et al. [37]. This step aims to improve the quality of the generated synthetic data by transferring realistic styles from real images to synthetic images. As depicted in Step 2 in Figure 1, every generated image G_M is enhanced by transferring style form the corresponding real image im_M. Then, the style transferred output image is presented using ST_M where $M = [0, 1, 2...999]$ in this study, representing the 1000 images in thr training dataset. In this process, a suitable $content : style$ ratio should be found, and it is a hyper-parameter in this second stage. However, this step is a separate training step from the training step of the SinGAN-Seg generative models. Therefore, this step is optional to follow, but we strongly recommend this style-transferring step to enhance the quality of the output data from the first step.

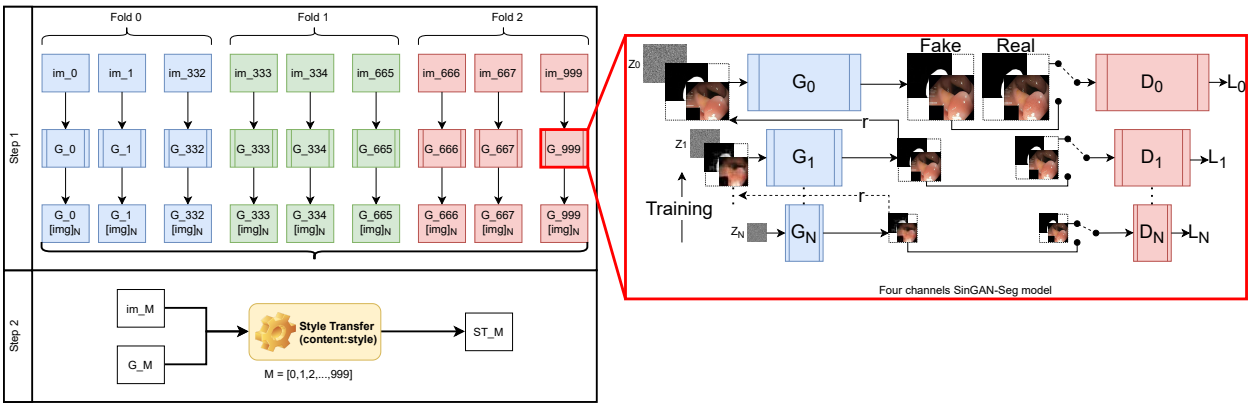

Figure 1: The complete pipeline of SinGAN-Seg to generate synthetic segmentation datasets. *Step 1*: represents the training of four channels SinGAN models. *Step 2*: represents fine tuning step using the neural style transfer [37]. *Four channels SinGAN*: Single training step of our four-channels SinGAN. Note the stacked input and output compared to the original SinGAN implementation [36] which input only single image with a noise vector and output only an image. In our SinGAN implementation, all the generators (from G_0 to G_{N-1}), except G_N, get four channels image (a polyp image and a ground truth) as the input in addition to the input noise vector. The first generator, G_N get only the noise vector as the input. The discriminators also get four channels images which consist of a RGB polyp image and a binary mask as input. The inputs to the discriminators can be either real or fake.

3 Experiments and results

This section demonstrates all the experiments and results collected using a polyp dataset as a case study. For all the experiments discussed in the following sections, we have used Pytorch deep learning framework [38].

3.1 Data

We have used a polyp dataset published with HyperKvasir dataset [20] which consists of polyp findings extracted from endoscopy examinations. This polyp dataset has 1000 polyp findings and a corresponding segmentation mask annotated by experts. We use only the polyp dataset as a case study because of the time and resource-consuming training process of the SinGAN-Seg pipeline. Furthermore, we use three-fold cross-validation, which is another time-consuming technique, for the experiments performed to find the validity of using synthetic data instead of real data.

A few sample images and the corresponding masks of the polyp dataset of HyperKvasir are depicted in Figure 2. The polyp images of the dataset are RGB images. The masks of the polyp images are single-channel images with white (255) for true pixels, which represent polyp regions, and black (0) for false pixels, which represent clean colon or background regions. In this dataset, there are different sizes of polyps. The distribution of polyp sizes as a percentage of the full image size is presented in the histogram plot in Figure 3. In this dataset, there are more relatively small polyps compared to larger polyps according to the plot presented in Figure 3. Additionally, this dataset was used to prove that the performance of segmentation models trained with small datasets can be improved using our SinGAN-Seg pipeline.

This dataset was used for two purposes.

1. To train SinGAN-Seg models to generate synthetic data.

2. To compare performance of real and synthetic

data for training segmentation ML models.

3.2 Training Generators

To use SinGAN-Seg to generate synthetic segmentation datasets to represent real segmentation datasets, we first trained SinGAN-Seg models one by one for each image in the training dataset. In our case study, there were 1000 polyp images and corresponding ground truth masks. Therefore, 1000 SinGAN-Seg models were trained. To train these SinGAN-Seg models, we have followed the same SinGAN settings used in the vanilla SynGAN paper [36]. Despite using the original training process, the input and output of SinGAN-Seg are four channels. After training each SinGAN-Seg by iterating 2000 epochs per scale of pyramidal GAN structure (see four channels SinGAN-Seg architecture in Figure 1 to understand this pyramidal GAN structure), we stored final checkpoints to generate synthetic data in the later stages from the each scale. The resolution of the training image of the SinGAN-Seg model is arbitrary because it depends on the size of the real polyp image. This input image is resized according to the pyramidal re-scaling structure introduced in the original implementation of SinGAN [36]. This rescaling pattern is depicted in the four channels SinGAN architecture in Figure 1. The re-scaling pattern used to train SinGAN-Seg models is used to change the randomness of synthetic data when pre-trained models are used to generate synthetic data. The models were trained on multiple computing nodes such as Google Colab with Tesla P100 16GB GPUs and a DGX-2 GPU server with 16 V100 GPUs because training 1000 GAN architectures one by one is a tremendous task. The average training time per SinGAN-Seg model was around 65 minutes.

After training SinGAN-Seg models, we have generated 10 random samples per real image using the input scale 0, which is the lowest scale that use a random noise input instead of a rescaled input image. For more details about these scaling numbers and corresponding output behaviors, please refer to the vanilla SinGAN paper [36]. Randomly selected three training images and the corresponding first 5 synthetic images generated using scale 0 are depicted in Figure 4. The first column of the figure represents the real images and the ground truth mask annotated from experts. The rest of the columns represents randomly generated synthetic images, and the corresponding generated mask.

In total, we have generated $10,000$ synthetic polyp images and the corresponding masks. SinGAN-Seg generates random samples with high variations when the input scale is 0. This variation can be easily recognized using the standard deviation (std) and the mean mask images presented in Figure 5. The mean and std images were calculated by stacking the 10 generated mask images corresponding to the 10 synthetic images related to a real image and calculating pixel-wise std and mean. Bright color in std images and dark color in mean images mean low variance of pixels. In contrast, dark color in std and bright color in mean images reflect high variance in pixel values. By investigating Figure 5, we can notice that small polyp masks have high variance compared to the large polyp mask as presented in the figure.

To understand the difference between the mask distribution of real images and synthetic images, we plotted pixel distribution of masks of synthetic $10,000$ images in Figure 6. This plot is comparable to the pixel distribution presented in Figure 3. The randomness of generations made differences in the distribution of true pixel percentages compared to the true pixel distribution of real masks of real images. However, the overall shape of synthetic data mask distribution shows a more or less similar distribution pattern to the real true pixel percentage distribution.

3.3 Style Transferring

After finishing the training of 1000 SinGAN-Seg models, the style transfer algorithm [37] was applied to every synthetic sample generated from SinGAN-Seg. In the style-transferring algorithm, we can change several parameters such as the number of epochs to transfer style from an image to another and the *content* : *style* weight ratio. This paper used a 1000 epoch to transfer style from a style image (real polyp image) to a content image (generated syn-

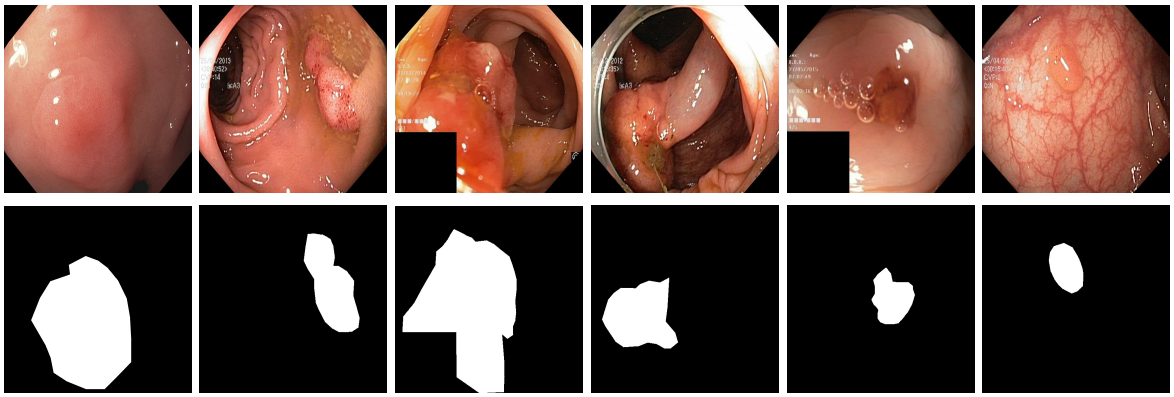

Figure 2: Sample images and corresponding masks from HyperKvasir [20] segmentation 1000 images.

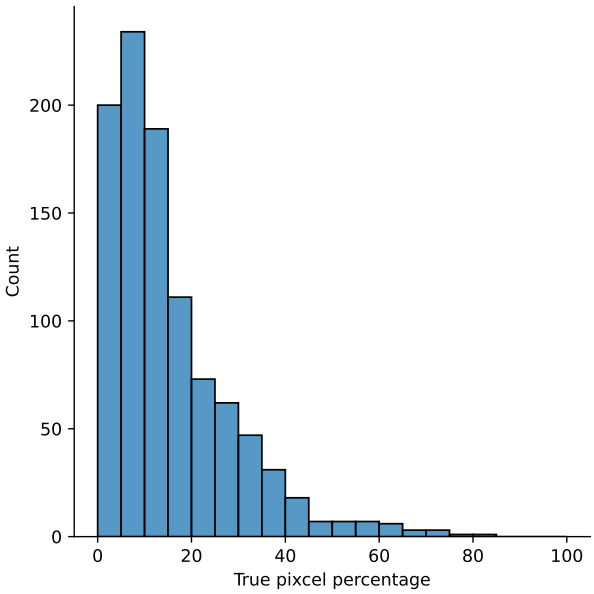

Figure 3: Distribution of true pixel percentages of the 1000 polyp masks of HyperKvasir [20] dataset.

thetic polyp). For performance comparisons, two *content* : *style* ratios, 1 : 1 and 1 : 1000 were used. An NVIDIA GeForce RTX 3080 GPU took around 20 seconds to transfer style for a single image.

We have depicted visual comparison between pure generated synthetic images and style transferred images (*content* : *style* = 1 : 1000) in Figure 7. Samples with the style transfer ratio 1 : 1 are not depicted here because it is difficult to see the differences between 1 : 1 and 1 : 1000 visually. The first column of Figure 7 shows the real images used as content images to transfer styles. The rest of the images in the first row of each image shows synthetic images generated from SinGAN-Seg before applying the style transferring algorithm. Then, the second row of each image shows the style transferred synthetic images. Differences of the synthetic images before and after applying the style transfer method can be easily recognized from images of the second reference image (using 3^{rd} and 4^{th} rows in Figure 7).

3.4 Python package and synthetic data

Using all the pre-trained SinGAN-Seg checkpoints, we have published a PyPI package and the corresponding GitHub repository to make all the experiments reproducible. Additionally, we have published the first synthetic polyp dataset to demonstrate how to share synthetic data instead of a real dataset that may have privacy concerns. The synthetic dataset is available at `https://osf.io/xrgz8/`. Moreover, this is an example synthetic dataset generated using the SinGAN-Seg pipeline. Furthermore, this dataset is an example showing how to increase a segmentation dataset size without using the time-consuming and

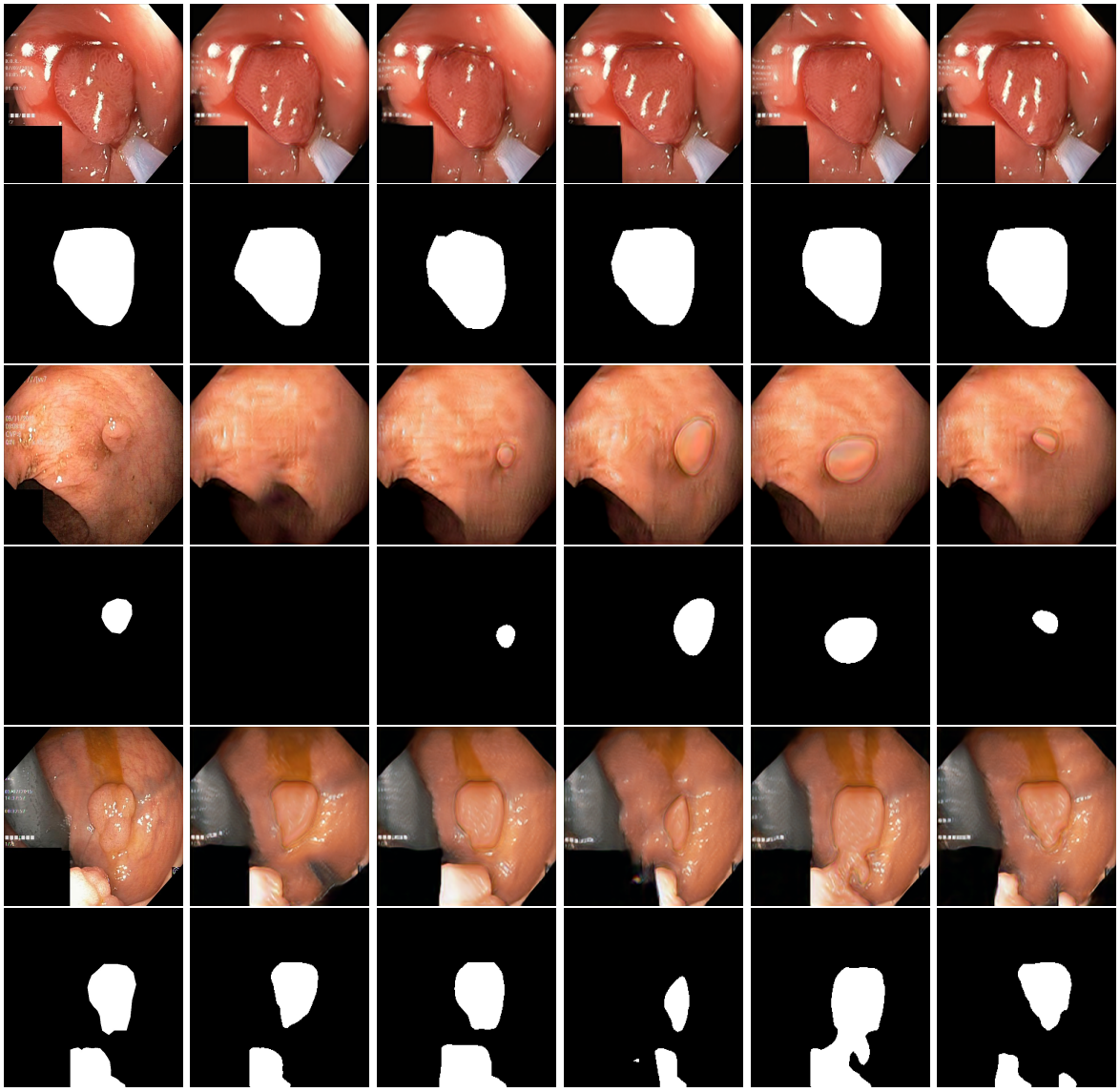

Figure 4: Sample real images and corresponding SinGAN generated synthetic GI-tract images with corresponding masks. The first column is illustrated with real images and masks. All other columns represent randomly generated synthetic data from SinGANs which were trained from the image on the first column.

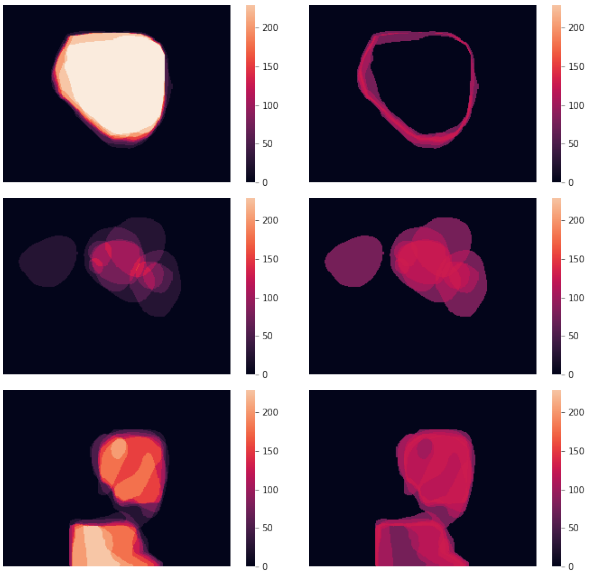

Figure 5: Mean and standard deviation calculated from 10 random mask generated from SinGAN-Seg. The corresponding real mask annotated from expertes can be seen in Figure 4.

costly medical data annotation process that needs experts' knowledge.

We named this PyPI package as `singan-seg-polyp (pip install singan-seg-polyp)` and it can be found here: `https://pypi.org/project/singan-seg-polyp/`. To the best of our knowledge, this is the only PyPI package to generate an unlimited number of synthetic polyps and corresponding masks. The corresponding GitHub repository is available at `https://github.com/vlbthambawita/singan-seg-polyp`. A set of functionalities were introduced in this package for end-users. Generative functions can generate random synthetic polyp data with their corresponding mask for a given image id from 1 to 1000 or for the given checkpoint directory, which is downloaded automatically when the generative functions are called. The style transfer function is in this package to transfer style from the real polyp images to the corresponding synthetic polyp images. In both functionalities, the relevant hyper-parameters can be

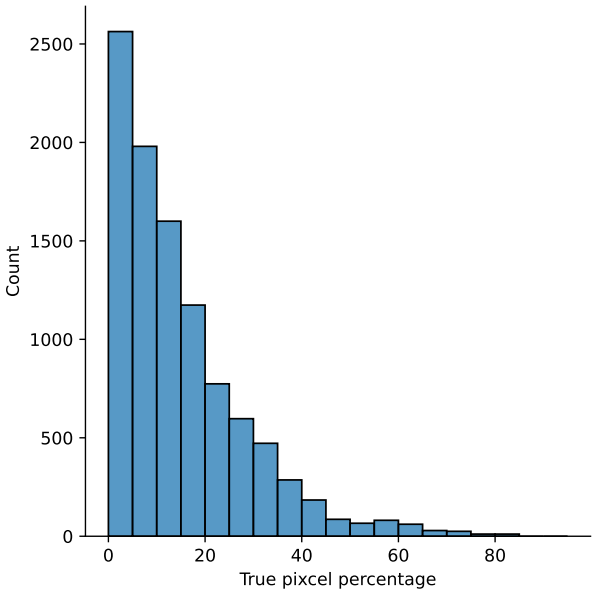

Figure 6: Distribution of $10,000$ masks of the synthetic generations. This $10,000$ represent the 1000 real polyp images. From each real image, 10 synthetic samples were generated. The synthetic $10,000$ dataset can be downloaded from `https://osf.io/xrgz8/`.

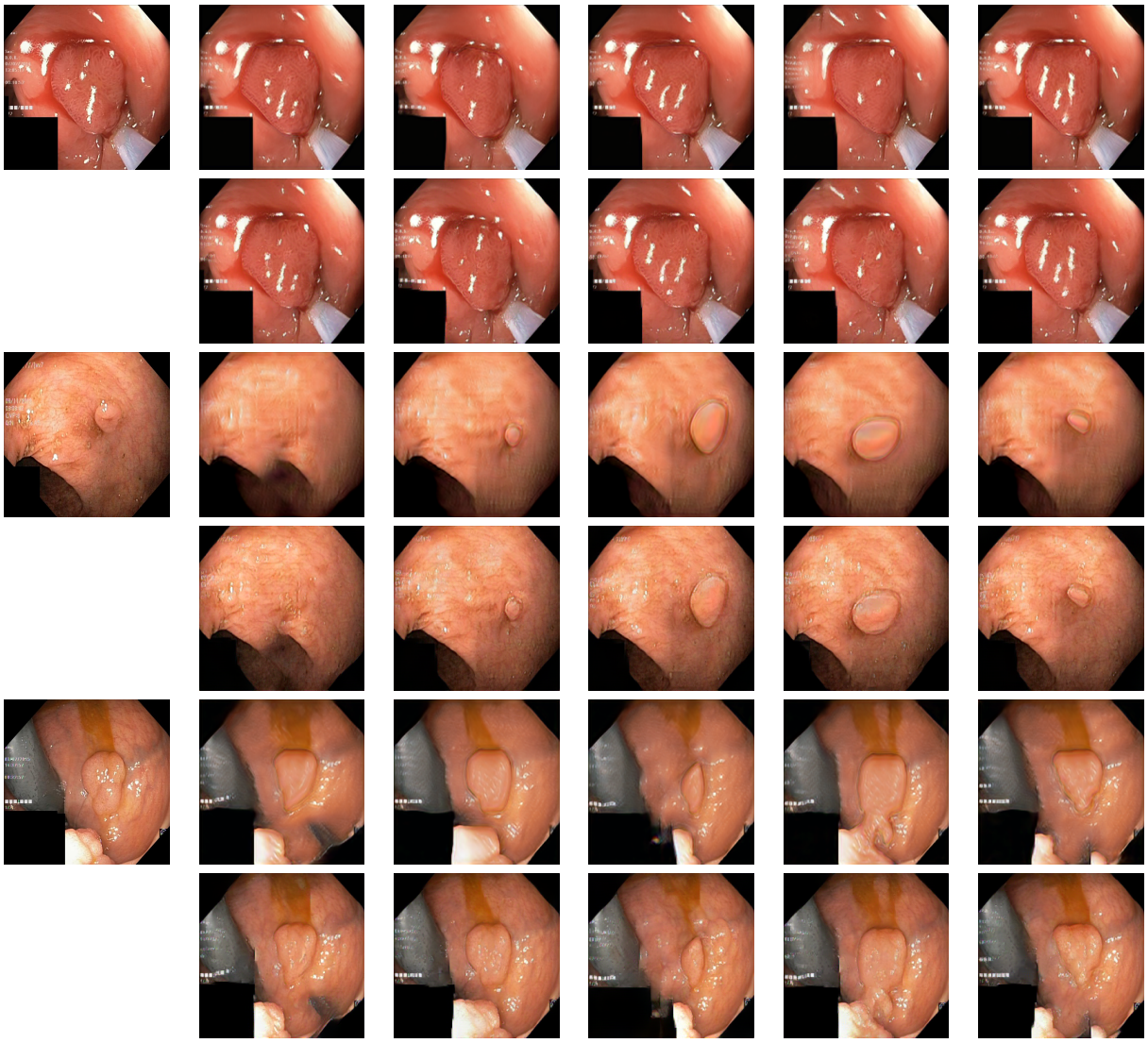

Figure 7: Direct generations of SinGAN-Seg versus Style transferred samples. The style transferring was performed using $1:1000$ content to style ratio.

changed as needed to end-users of this PyPI package.

3.5 Baseline experiments

Two different sets of baseline experiments were performed for two different objectives. The first objective was to compare the quality of generated synthetic data over the real data. Using these baseline experiments, we can identify the capability of sharing SinGAN-Seg synthetic data instead of the real datasets for omitting privacy concerns. The second objective was to test how to use SinGAN-Seg pipeline to improve the segmentation performance when the size of training dataset of real images and masks are small. For all the baseline experiments, we selected Unet++ [39] as the main segmentation model according to the performance comparison done by the winning team at EndoCV 2021 [16]. The single-channel dice loss function used in the same study was used to train Unet++ polyp segmentation models. The `se_resnext50_32x4d` network as the encoder of the UNet++ model and `softmax2d` as the activation function of the last layer were used according to the result of the winning team at EndoCV 2021 [16].

Pytorch deep learning library was used as the main development framework for the baseline experiments also. Training data stream was handled using PYRA [14] data loader with Albumentations augmentation library [40]. The real images and the synthetic images were resized into 128×128 using this data handler for all the baseline experiments to save training time because we had to train multiple models for fair comparisons. We have used an initial learning rate of 0.0001 for 50 epochs and then change it to 0.00001 for the rest of the training epochs for all the training processes of UNet++. The UNet++ models used to compare real versus synthetic data were trained 300 epochs in total. On the other hand, the UNet++ models used to measure the effect of using SinGAN-Seg synthetic data for small segmentation datasets were trained only 100 epochs because the size of the data splits used to train the models are getting bigger when increasing the training data. In all the experiments, we have selected the best checkpoint using the best validation IOU score. Finally, dice loss, IOU score, F-score, accuracy, recall, and precision were calculated for comparisons using validation folds.

3.5.1 Synthetic data vs real data for segmentation

We have performed three-folds cross-validation to compare polyp segmentation performance using UNet++ when using real and synthetic data. First, we divided the real dataset (1000 polyp images and the corresponding segmentation masks) into three folds. Then, the trained SynGAN-Seg generative models and the corresponding generated synthetic data were also divided into the same three folds. These three folds are presented using three colors in Step I of Figure 1. In any of the experiments, training data folds and corresponding synthetic data folds were not mixed with the validation data folds. If mixed, it leads to a data leakage problem.

Then, the baseline performance of the UNet++ model was evaluated using the three folds of the real data. In this experiment, the UNet++ model was trained using two folds and validated using the remaining fold of the real data. In total, three UNet++ models were trained and calculated the average performance using dice loss, IOU score, F-score, accuracy, recall, and precision only for the polyp class because the most important class of this dataset is the polyp class. This three-fold baseline experiment setup is depicted on the left side of Figure 8.

The usability of synthetic images and corresponding masks generated from SinGAN-Seg was investigated using three-fold experiments as organized in the right side of Figure 8. In this case, UNet++ models were trained only using synthetic data generated from pre-trained generative models and tested using the real data folds, which were not used to train the generative models used to generate the synthetic data. Five different $N(N = [1, 2, 3, 4, 5])$ amount of synthetic data per image were used to train UNet++ models. This data organization process can be identified easily using the color scheme of the figure. To test the quality of pure generations, first, we used the direct output from SinGAN-Seg to train UNet++ models. Then, the style transfer method was applied with $1 : 1$ content to style ratio for all the synthetic

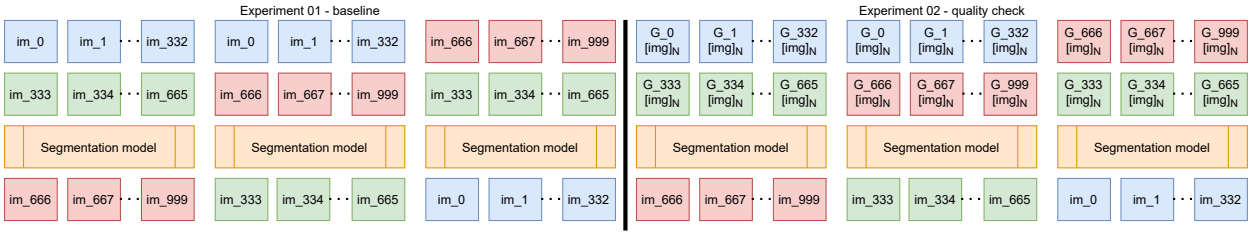

Figure 8: Three step experiment setup to analyze the quality of SinGAN output.

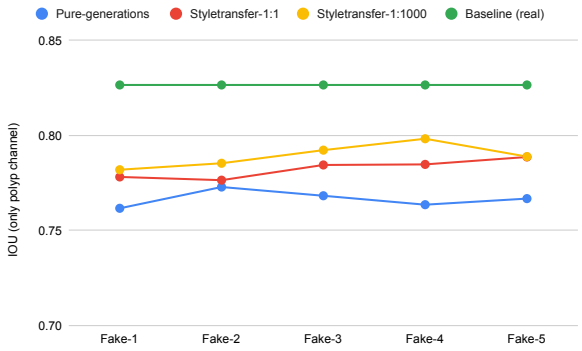

Figure 9: Real versus synthetic data performance comparison with UNet++ and the effect of applying the style-transferring post processing.

data. These style transferred images were used as training data and tested using the real dataset. In addition to the 1 : 1 ratio, 1 : 1000 was tested as a style transfer ratio for the same set of experiments.

Table 1 shows the results collected from the UNet++ segmentation experiments for the baseline experiment and the experiments conducted with synthetic data, which contains pure generated synthetic data and style transferred data using 1 : 1 and 1 : 1000. Differences in IOU scores of these three experiments are plotted in Figure 9 for easy comparison.

3.5.2 Synthetic segmentation data for real small datasets

The main purpose of these experiments are to find the effect of using synthetic data generated from the SinGAN-Seg pipeline instead of small real datasets because the SinGAN-Seg pipeline can generate an unlimited number of synthetic samples per real image. A synthetic sample consists of a synthetic image and the corresponding ground truth mask. Therefore, experts' knowledge is not required to annotate the ground truth mask. For these experiments, we have selected the best parameters of the SinGAN-Seg pipeline from the experiments performed under Section 3.5.1. First, we created small 10 real polyp datasets from the fold one such that each dataset contains R number of images and R can be one of the values of $[5, 10, 15, ..., 50]$. The corresponding synthetic dataset was created by generating 10 synthetic images and corresponding masks per real image. Then, our synthetic datasets consist of S number of images such that $S = [50, 100, 150, ..., 500]$. Then we have compared true pixel percentages of real masks and synthetic masks generated from SynGAN-Seg pipeline using histograms of bin size of 5. The histograms are depicted in Figure 10. The first row represents the histograms of real small detests, and the second row represents the histograms of corresponding synthetic datasets. Compare pairs (one from the top row and the corresponding one from the bottom) to get a clear idea of how the generated synthetic data improved the distribution of masks.

UNet++ segmentation models were trained using these real and synthetic datasets separately. Then we have compared the performance differences using

Table 1: Three-fold average of basic metrics to compare real vs synthetic performance with UNet++ and the effect of style-transfers performance

Train data	ST (cw:sw)	dice_loss	iou_score	fscore	accuracy	recall	precision
REAL	NA	0.1123	0.8266	0.8882	0.9671	0.8982	0.9161
FAKE-1	No ST	0.1645	0.7617	0.8357	0.9531	0.863	0.8793
	1:1	0.1504	0.7782	0.85	0.9572	0.8672	0.8917
	1:1000	0.1473	0.782	0.853	0.9591	0.8624	0.9005
FAKE-2	No ST	0.1549	0.7729	0.8453	0.9561	0.8692	0.8895
	1:1	0.155	0.7765	0.8453	0.9575	0.8729	0.8852
	1:1000	0.1477	0.7854	0.8525	0.9609	0.8647	0.9038
FAKE-3	No ST	0.161	0.7683	0.8391	0.9556	0.8568	0.8945
	1:1	0.1475	0.7845	0.8525	0.9585	0.8723	0.8936
	1:1000	0.1408	0.7923	0.8593	0.9629	0.8693	0.9078
FAKE-4	No ST	0.1649	0.7638	0.8352	0.9525	0.8669	0.878
	1:1	0.1464	0.7848	0.8537	0.9594	0.8713	0.8921
	1:1000	0.137	0.7983	0.863	0.9636	0.8653	0.9185
FAKE-5	No ST	0.1654	0.7668	0.8345	0.9563	0.8565	0.8919
	1:1	0.1453	0.7887	0.8547	0.961	0.8703	0.9
	1:1000	0.1458	0.7889	0.8543	0.962	0.8527	0.9211

validation folds. In this experiments, the training datasets were prepared using the fold one. The remaining two folds were used as the validation dataset. The collected results from UNet++ models trained with the real datasets and the synthetic datasets are tabulated in Table 2. A comparison of the corresponding IOU scores are plotted in Figure 11.

4 Discussion

The SinGAN-Seg pipeline has two steps. The first one is generating synthetic polyp images and the corresponding ground truth masks. The second is transferring style from real polyp images to synthetic polyp images to make them more realistic than the pure generations from the first step. We have developed this pipeline to achieve the main two goals. The first one is for sharing medical data when privacy concerns are to share real data. The second one uses is to improve the polyp segmentation performance when the size of training datasets are small.

4.1 SinGAN-Seg as data sharing technique

The SinGAN-Seg can generate unlimited synthetic data with the corresponding ground truth mask, representing real datasets. This SinGAN-Seg pipeline is applicable for any dataset with segmentation masks, particularly when the dataset is not sharable due to privacy concerns. However, in this study, we applied this pipeline to a public polyp dataset with segmentation masks as a case study. Assuming that the polyp dataset is private, we used this polyp dataset as a proof of concept medical dataset. In this case, we published PyPI package, `singan-seg-polyp` which can generate an unlimited number of polyp images and corresponding ground truth masks. If the real polyp dataset is restricted for public use, then this type of pip package can be published as an alternative dataset to represent the real dataset. Alternatively, we can publish a pre-generated synthetic dataset using the SinGAN-Seg pipeline, such as the

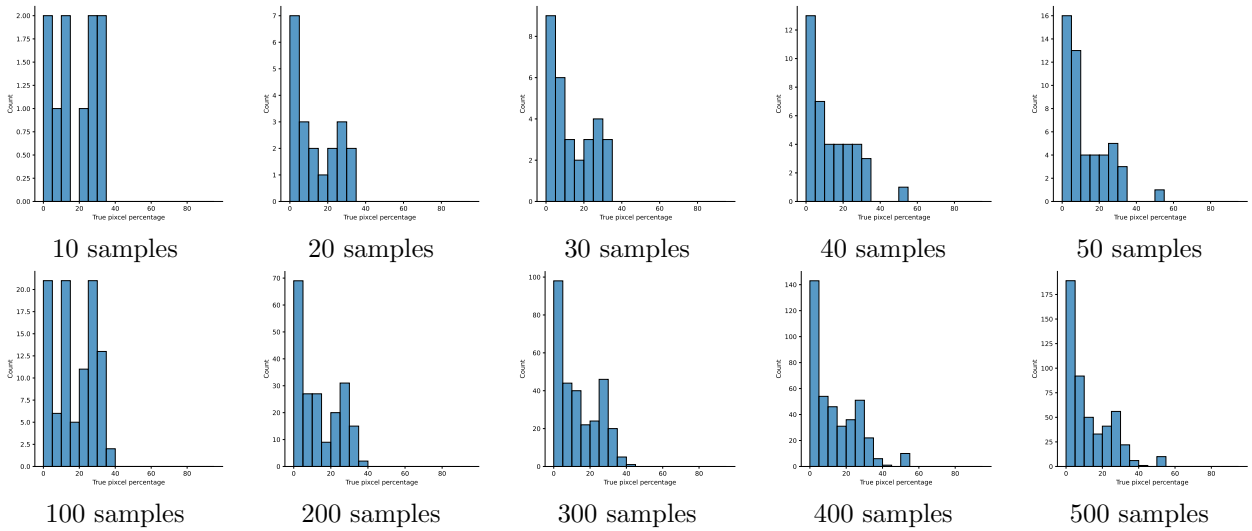

Figure 10: Distribution comparison between real and synthetic mask. Synthetic mask were generated using the SinGAN-Seg.

synthetic polyp dataset published as a case study at `https://osf.io/xrgz8/`.

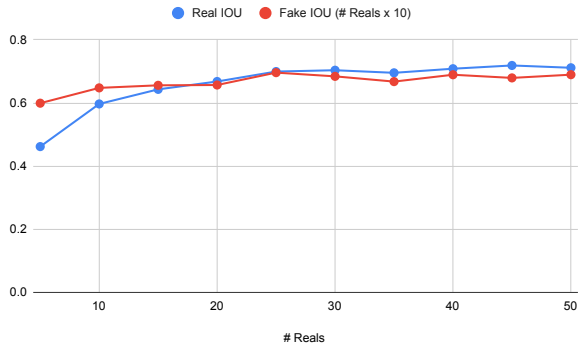

Figure 11: Real versus Fake performance comparison with small training datasets

According to the results presented in Table 1, the UNet++ segmentation network perform better when the real data is used as training data compared to using synthetic data as training data. However, the small performance gap between real and synthetic data as training data implies that the synthetic data generated from the SinGAN-Seg can use as an alternative to sharing segmentation data instead of real datasets, which are restricted to share. The style-transferring step of the SinGAN-Seg pipeline could reduce the performance gap between real and synthetic data as training data for UNet++. The performance gap between real data and the synthetic data as training data for segmentation models is negotiable because the primary purpose of producing the synthetic data is not to improve the performance of segmentation models but to introduce an alternative data sharing which are practically applicable when datasets have privacy concerns to share.

Table 2: Real vs Fake comparisons for small datasets. The fake images were generated using style tranfer ration 1 : 1000.

		dice_loss	iou_score	fscore	accuracy	recall	precision
Real	5	0.4662	0.4618	0.5944	0.8751	0.7239	0.6305
Fake	50	**0.3063**	**0.5993**	**0.7048**	**0.9211**	**0.7090**	**0.8133**
Real	10	0.3932	0.5969	0.7079	0.9164	0.7785	0.7516
Fake	100	**0.2565**	**0.6478**	**0.7457**	**0.9259**	**0.7911**	**0.7970**
Real	15	0.2992	0.6431	0.7402	0.9322	0.7388	**0.8602**
Fake	150	**0.2852**	**0.6559**	**0.7624**	**0.9329**	**0.8172**	0.7833
Real	20	0.3070	**0.6680**	**0.7668**	0.9328	**0.7771**	0.8566
Fake	200	**0.2532**	0.6569	0.7544	**0.9342**	0.7317	**0.8827**
Real	25	**0.2166**	**0.6995**	**0.7929**	0.9405	**0.7955**	0.8804
Fake	250	0.2182	0.6961	0.7860	**0.9418**	0.7690	**0.8957**
Real	30	**0.2100**	**0.7037**	**0.7971**	**0.9417**	**0.8005**	0.8758
Fake	300	0.2228	0.6843	0.7797	0.9388	0.7683	**0.8810**
Real	35	**0.2164**	**0.6955**	**0.7889**	**0.9398**	**0.8157**	0.8456
Fake	350	0.2465	0.6677	0.7543	0.9346	0.7385	**0.8933**
Real	40	**0.2065**	**0.7085**	**0.7974**	**0.9417**	0.7881	**0.8947**
Fake	400	0.2194	0.6894	0.7816	0.9305	**0.8276**	0.8219
Real	45	**0.1982**	**0.7188**	**0.8062**	**0.9441**	**0.8120**	**0.8839**
Fake	450	0.2319	0.6794	0.7697	0.9341	0.7859	0.8633
Real	50	**0.2091**	**0.7115**	**0.7948**	**0.9418**	0.7898	**0.8932**
Fake	500	0.2255	0.6896	0.7756	0.9380	**0.7961**	0.8644

4.2 SinGAN-Seg with small datasets

In addition to using the SinGAN-Seg pipeline as a data-sharing technique when the real datasets are restricted to publish, the pipeline can improve the performance of segmentation tasks when a dataset is really small. In this case, the SinGAN-Seg pipeline can generate synthetic data to overcome the problem associated with the small dataset. In other words, the SinGAN-Seg pipeline act as a data augmentation technique. The SinGAN-Seg-based data augmentation acts as an unlimited number of stochastic augmentation techniques due to the randomness of the synthetic data generated from this model. For an example, consider a manual segmentation process such as cell segmentation in any medical laboratory experiment. This type of task is really hard to perform for experts as well. As a result, the amount of data collected with manually annotated masks are limited. Our SinGAN-Seg pipeline can improve these datasets by generating an unlimited number of random samples from a single manually annotated image. This study showed that these synthetic data generated from a small real dataset can improve the performance of segmentation machine learning models. For example, when the real polyp dataset size is 5 to train our UNnet++ model, the synthetic dataset with 50 samples showed 30% improvement over the IOU score of using the real data samples.

5 Conclusions and future work

This paper presented a four-channel SinGAN-Seg model and the corresponding SinGAN-Seg pipeline with a style transfer method to generate realistic synthetic polyp images and the corresponding ground truth masks. This SinGAN-Seg pipeline can be used as an alternative data sharing method when real datasets are restricted to share. Moreover, this pipeline can be used for improving the segmentation performance when we have small segmentation real datasets. The conducted three-folds cross-validation experiments and collected results show that synthetic data can achieve very close performance for segmentation tasks when we use only synthetic images and corresponding masks compared to the segmentation performance if the real data and experts annotated data is used when the real dataset has a considerable amount of data. On the other hand, we show that SinGAN-Seg pipeline can achieve better segmentation performance when training datasets are very small.

In future studies, researchers can combine super-resolution GAN model [41] to this pipeline to improve the quality of the output after the style transfer step. When we have high-resolution images, machine learning algorithms show better performance than algorithms trained using low-resolution images [42].

6 acknowledgments

The research has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under contract 270053.

References

[1] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial intelligence in healthcare: past, present and future," *Stroke and vascular neurology*, vol. 2, no. 4, 2017.

[2] S. E. Dilsizian and E. L. Siegel, "Artificial intelligence in medicine and cardiac imaging: Harnessing big data and advanced computing to provide personalized medical diagnosis and treatment," *Current Cardiology Reports*, vol. 16, no. 1, p. 441, 2013. [Online]. Available: https://doi.org/10.1007/s11886-013-0441-8

[3] V. L. Patel, E. H. Shortliffe, M. Stefanelli, P. Szolovits, M. R. Berthold, R. Bellazzi, and A. Abu-Hanna, "The coming of age of artificial intelligence in medicine," *Artificial Intelligence in Medicine*, vol. 46, no. 1, pp. 5–17, 2009, artificial Intelligence in Medicine AIME' 07. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365708000961

[4] S. Jha and E. J. Topol, "Adapting to artificial intelligence: radiologists and pathologists as information specialists," *Jama*, vol. 316, no. 22, pp. 2353–2354, 2016.

[5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[6] M. A. Hearst, "Support vector machines," *IEEE Intelligent Systems*, vol. 13, no. 4, p. 18–28, Jul. 1998. [Online]. Available: https://doi.org/10.1109/5254.708428

[7] Haifeng Wang and Dejin Hu, "Comparison of svm and ls-svm for regression," in *2005 International Conference on Neural Networks and Brain*, vol. 1, 2005, pp. 279–283.

[8] A. Suárez Sánchez, P. García Nieto, P. Riesgo Fernández, J. del Coz Díaz, and F. Iglesias-Rodríguez, "Application of an svm-based regression model to the air quality study at local scale in the avilés urban area (spain)," *Mathematical and Computer Modelling*, vol. 54, no. 5, pp. 1453–1466, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717711002196

[9] S. Yue, P. Li, and P. Hao, "Svm classification:its contents and challenges," *Applied Mathematics-A Journal of Chinese Universities*, vol. 18, no. 3, pp. 332–342, 2003. [Online]. Available: https://doi.org/10.1007/s11766-003-0059-5

[10] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.

[11] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, p. 29, 2015. [Online]. Available: https://doi.org/10.1186/s12880-015-0068-x

[12] L. K. Lee, S. C. Liew, and W. J. Thong, "A review of image segmentation methodologies in medical image," *Advanced computer and communication engineering technology*, pp. 1069–1080, 2015.

[13] L. A. Torre, F. Bray, R. L. Siegel, J. Ferlay, J. Lortet-Tieulent, and A. Jemal, "Global cancer statistics, 2012," *CA: a cancer journal for clinicians*, vol. 65, no. 2, pp. 87–108, 2015.

[14] V. Thambawita, S. Hicks, P. Halvorsen, and M. A. Riegler, "Pyramid-focus-augmentation: Medical image segmentation with step-wise focus," *arXiv preprint arXiv:2012.07430*, 2020.

[15] D. Jha, S. Ali, N. K. Tomar, H. D. Johansen, D. Johansen, J. Rittscher, M. A. Riegler, and P. Halvorsen, "Real-time polyp detection, localization and segmentation in colonoscopy using deep learning," *Ieee Access*, vol. 9, pp. 40 496–40 510, 2021.

[16] V. Thambawita, S. A. Hicks, P. Halvorsen, and M. A. Riegler, "Divergentnets: Medical image segmentation by network ensemble." in *EndoCV@ ISBI*, 2021.

[17] V. Prasath, "Polyp detection and segmentation from video capsule endoscopy: A review," *Journal of Imaging*, vol. 3, no. 1, p. 1, 2017.

[18] D. Jha, N. K. Tomar, S. Ali, M. A. Riegler, H. D. Johansen, D. Johansen, T. de Lange, and P. Halvorsen, "Nanonet: Real-time polyp segmentation in video capsule endoscopy and colonoscopy," *arXiv preprint arXiv:2104.11138*, 2021.

[19] I. N. Figueiredo, S. Prasath, Y.-H. R. Tsai, and P. N. Figueiredo, "Automatic detection and segmentation of colonic polyps in wireless capsule images," *ICES REPORT*, pp. 10–36, 2010.

[20] H. Borgli, V. Thambawita, P. H. Smedsrud, S. Hicks, D. Jha, S. L. Eskeland, K. R. Randel, K. Pogorelov, M. Lux, D. T. D. Nguyen, D. Johansen, C. Griwodz, H. K. Stensland, E. Garcia-Ceja, P. T. Schmidt, H. L. Hammer, M. A. Riegler, P. Halvorsen, and T. de Lange, "Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Scientific Data*, vol. 7, no. 1, p. 283, 2020.

[21] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Medical Imaging and Graphics*, vol. 43, pp. 99–111, 2015.

[22] J. Silva, A. Histace, O. Romain, X. Dray, and B. Granado, "Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer," *International journal of computer assisted radiology and surgery*, vol. 9, no. 2, pp. 283–293, 2014.

[23] N. Tajbakhsh, S. R. Gurudu, and J. Liang, "Automated polyp detection in colonoscopy videos using shape and context information," *IEEE transactions on medical imaging*, vol. 35, no. 2, pp. 630–644, 2015.

[24] L. F. Sánchez-Peralta, J. B. Pagador, A. Picón, Á. J. Calderón, F. Polo, N. Andraka, R. Bilbao, B. Glover, C. L. Saratxaga, and F. M. Sánchez-Margallo, "Piccolo white-light and narrow-band imaging colonoscopic dataset: A performance comparative of models and datasets," *Applied*

Sciences, vol. 10, no. 23, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/23/8501

[25] "The norwegian data protection authority," accessed: 2021-04-25. [Online]. Available: https://www.datatilsynet.no/en/

[26] "The personal data act," accessed: 2021-04-25. [Online]. Available: https://www.forskningsetikk.no/en/resources/the-research-ethics-library/legal-statutes-and-guidelines/the-personal-data-act/

[27] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)."

[28] P. Edemekong, P. Annamaraju, and M. Haydel, "Health insurance portability and accountability act," *StatPearls*, 2020.

[29] "California consumer privacy act," 2018. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[30] "Act on the protection of personal information," 2003. [Online]. Available: https://www.cas.go.jp/jp/seisaku/hourei/data/APPI.pdf

[31] "Personal information protection commission," 2011. [Online]. Available: http://www.pipc.go.kr/cmt/main/english.do

[32] "The personal data protection bill," 2018. [Online]. Available: https://www.meity.gov.in/writereaddata/files/Personal_Data_Protection_Bill,2018.pdf

[33] F. Renard, S. Guedria, N. D. Palma, and N. Vuillerme, "Variability and reproducibility in deep learning for medical image segmentation," *Scientific Reports*, vol. 10, no. 1, p. 13724, 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-69920-0

[34] M. J. Willemink, W. A. Koszek, C. Hardell, J. Wu, D. Fleischmann, H. Harvey, L. R. Folio, R. M. Summers, D. L. Rubin, and M. P. Lungren, "Preparing medical imaging data for machine learning," *Radiology*, vol. 295, no. 1, pp. 4–15, 2020.

[35] S. Yu, M. Chen, E. Zhang, J. Wu, H. Yu, Z. Yang, L. Ma, X. Gu, and W. Lu, "Robustness study of noisy annotation in deep learning based medical image segmentation," *Physics in Medicine & Biology*, vol. 65, no. 17, p. 175007, aug 2020. [Online]. Available: https://doi.org/10.1088/1361-6560/ab99e5

[36] T. R. Shaham, T. Dekel, and T. Michaeli, "Singan: Learning a generative model from a single natural image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4570–4580.

[37] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[39] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[40] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2,

2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125

[41] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[42] V. L. Thambawita, S. Hicks, I. Strümke, M. A. Riegler, P. Halvorsen, and S. Parasa, "Fr615 impact of image resolution on convolutional neural networks performance in gastrointestinal endoscopy," *Gastroenterology*, vol. 160, no. 6, pp. S–377, 2021.