university of groningen

faculty of science
and engineering

Generating Synthetic Training Data using Deep Generative Adversarial Networks in Medical Endoscopy Images

Mihai Popescu

university of groningen / faculty of science and engineering

University of Groningen

Generating Synthetic Training Data using Deep Generative Adversarial Networks in Medical Endoscopy Images

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Artificial Intelligence
at University of Groningen under the supervision of
dr. F. Cnossen (Director of Education Ba/Ma Artificial Intelligence/Human-Machine Communication — Associate Professor Cognitive Engineering, University of Groningen)
and
dr. ir. P.M.A. van Ooijen (Scientific Researcher / Associate Professor, University of Groningen & University Medical Center Groningen)

Mihai Popescu (s3192423)

November 12, 2021

Contents

Page

Acknowledgements **5**

Abstract **6**

1 Introduction **7**
1.1 Research Question 8

2 Background Literature **9**
2.1 Theoretical Framework 9
2.1.1 Detection 9
2.1.2 Types of Learning 10
2.1.3 Supervised learning 10
2.2 Generative Methods 10
2.2.1 Generative Adversarial Networks 10
2.2.2 Performance Criteria 11
2.2.3 Latent Space 12
2.2.4 Conditional GAN 12
2.2.5 Common Issues with GANs 13
2.3 State of the Art 13
2.3.1 PGGAN 14
2.3.2 StyleGAN2 15
2.3.3 StyleGAN2-Ada 17

3 Methods **18**
3.1 Technical Implementational Challenges 18
3.2 Model Comparison 18
3.3 Data 19
3.3.1 Data Collection 19
3.3.2 Data Preprocessing 20
3.3.3 Data Labelling 20
3.4 Models 20
3.4.1 Baseline GAN 21
3.4.2 PGGAN 21
3.4.3 StyleGAN2-Ada 21
3.4.4 Conditional SyleGAN2-Ada 21
3.4.5 YOLO5 Detection Model 21

4 Experimental Setup **26**
4.1 Tools and Technologies 26
4.2 Performance Criteria 26

5 Results **29**
- 5.1 Images . . . 29
  - 5.1.1 Baseline GAN . . . 29
  - 5.1.2 PGGAN . . . 30
  - 5.1.3 StyleGAN2-Ada . . . 31
- 5.2 FID Scores . . . 31
- 5.3 Conditional StyleGAN2-Ada . . . 31
- 5.4 YOLOv5 Detector . . . 31

6 Discussion **36**
- 6.1 Research Question 1: Image Realism . . . 36
- 6.2 Research Question 2: Detection Increase Through Synthesis . . . 37
  - 6.2.1 Automated Data Labeling . . . 37
  - 6.2.2 Latent Space Interpolation . . . 37
- 6.3 Further Extensions . . . 38

7 Conclusion **40**
- 7.1 Image Realism . . . 40
- 7.2 Detection System Performance . . . 40

8 Bibliography **41**

Appendices **43**
- A Literature . . . 43
- B Generated images . . . 44
  - B.1 Baseline Candidate Model . . . 44
  - B.2 PGGAN Candidate Model . . . 55
  - B.3 StyleGAN2-Ada Candidate Model . . . 56
  - B.4 Latent space interpolation . . . 59
- C ZiuZ detector on fakes . . . 60

Acknowledgments

The following paper was done in collaboration with the University Medical Centre of Groningen (UMCG) and ZiuZ. I would like to thank the members of both institutions for their guidance, wisdom, and medical insight along the way.

A warm thank you to Hung Chu who supervised and assisted with the technical aspects of the paper and proved to be instrumental in their thirst for knowledge and experimentation.

Many thanks to Ioannis Giotis from ZiuZ for their deep insight into applying deep learning models in the field of endoscopic imaging.

Last, but certainly not least, all of this would not have been possible without coffee. Praise caffeine!

Abstract

Building automated detection tools for endoscopy procedures has been a pursued interest in the field of machine learning to reduce the number of omitted polyps during endoscopies. Training such systems is difficult in the current landscape as the availability of medical images containing polyps is low. This thesis attempts to solve data scarcity by synthesizing images containing polyps using generative adversarial networks in order to augment existing polyp datasets used by detection models. The most promising model based on StyleGAN2-Ada produced realistic images that, when augmented into the original training set of the detector, obtained a mean average precision score of 92.13% compared to the 92.44% obtained by the detector trained on a non-augmented dataset. Although the performance of the model did not increase, the quality of the generated images was impressive from a realism standpoint and promising conclusions could be drawn regarding the possibility of manipulating the latent space and generative conditional embedding of the network to generate custom types of polyps.

1 Introduction

Endoscopy examinations serve as a staple procedure in detecting and classifying abnormal tissue growths. Bowel polyps (or lesions) are one of the most common types of abnormal tissue growths that occur. There are around 1500 unique polyps from which between $71\% - 75\%$ can turn to cancer. Once they turn cancerous, polyps are referred to as adenomatous polyps [1]. The timely detection of such growths in patients is crucial as it affords the patient the opportunity to seek medical treatment and increases the chances of survival and quality of life. Though medical practitioners are trained to spot abnormal growths using endoscopes, some omissions can involuntarily happen during the procedure. The effect of said omissions, whether it be from a lack of attentiveness, distractions in the procedure room, or the difficult nature of the texture and shape of the abnormality, can significantly impact the health of the patient since cancer left untreated can evolve and potentially spread. In the past, several assistive systems have been developed to detect the presence of lesions in real-time endoscopy procedures and have proven their efficacy by reducing the omission rates of medical practitioners [2, 3].

Although the detection rate of cancerous lesions has increased with the introduction of automated detection systems [4], several technical and operational challenges persist within the usage of the systems during the endoscopy procedure. The following challenges need to be solved to successfully implement a robust detection system.

Firstly, the medium through which current automated detection systems highlight the lesion is visually, on the endoscope's camera feed, through the creation of a box at the lesion's location, and the generation of a sound to alert the endoscopist that a lesion has been detected. Since medical systems skew to minimize the number of false negatives, the system is very sensitive. Because of this, as reported by clinicians, the system will generally report an abundance of false positives. This has the effect of decreasing the attentiveness of the endoscopist towards the system as the alerts coming from the system get discarded and true positives pass under the veil of false positives. A solution to this problem is to increase the reliability of the detection model by both experimentally tuning the model with an endoscopist present and introducing more examples of polyps within the training data. We will focus on the second point in this paper.

Secondly, as seen in Figure 18, polyps come in different shapes and sizes. Certain polyps are easier to spot than others. Endoscopists report that they themselves are inexperienced in recognizing certain polyps that present difficult characteristics, such as a homogeneous texture with the tissue around the polyp, the size and shape, etc. This inexperience is underlined by high missing rates during the procedure and the inability to decide on whether to remove the polyp or perform a biopsy to determine its nature. Due to this inexperience, assistive tools such as guidelines and flow charts have been created for endoscopists to be used in real-time during the procedure [5]. The guidelines report that flat and recessed lesions are more difficult to spot by an endoscopist than protruding lesions. This partially explains the inexperience of endoscopists as lower detection rates for recessed and flat lesions also mean that there are fewer examples of them to learn from. This has implications for the medical practitioner and engineers tasked with building automated detection systems as there will be a lower presence of such lesions in the data. Because of this, the missing rate of flat or recessed lesions is expected to be larger compared to protruding lesions and class imbalance in the data needs to be solved. Since there is a lack of images of recessed and flat polyps, generating the images ourselves could potentially solve the class imbalance problem and increase the performance of the detection model.

Thirdly, successfully building a system that learns to detect lesions in real-time implies that there is sufficient data to represent the distribution of the population of lesions to a degree that the system will be able to generalize on new, never seen before lesions, from a wide degree of angles and a range of different lesion types. However, a comprehensive survey [6], which focused on aggregating datasets that contain images or films taken during endoscopies, found that there is a great imbalance in the availability of such data. Generally, there is a wide range of images taken during endoscopies to be found, but there is a lack of images that actually contain lesions - at least in the public domain. It is evident thus that creating a robust detection system, that operates in a manner tailored to the medical practitioner, is challenging. The system needs to be able to catch protruding lesions, but also flat and recessed lesions, in real-time. Moreover, the availability of training data is scarce which further increases the difficulty of the challenge.

This paper attempts to solve the problem of data scarcity through data synthesis. Because existing training sets lack high-quality data of images containing lesions, the target scope of this paper will be to i) determine whether the synthesis of images containing lesions is viable, from a subjective standpoint, using state-of-the-art generative techniques and ii) whether the inclusion of synthetic images in the original training sets of the detection models will see an increase in detection performance. The paper will also include a comprehensive comparison between the images of the generative models from both a subjective ”realism” standpoint, but more importantly an objective evaluation metric that will measure the relative increase in detection performance of the detection model after it has been trained on synthesized images. Finally, the winning generative model will be the one that increases the performance of detection models the most.

1.1 Research Question

To summarize, this thesis focuses on the following problems:

Q1. Is it possible to synthesize subjectively realistic images of lesions using generative adversarial networks?

Q2. Can the performance of lesion detection models be increased through data augmentation using synthetic data generated by generative adversarial networks?

2 Background Literature

The formulation of the main research question should be interpreted as solving a data synthesis problem. Considerations towards solving this problem are present in this section. Formal concepts pertaining to the generative models will be given as well as other practical considerations of model building. As the literature is presented, it will be deliberated with the medical context in mind as the end product operates in a medical environment. Following the theoretical framework, an experimental pipeline entailing the required steps of determining the optimal model will be given which will showcase the data flow from the very beginning of the training process to determining the effect of synthetic data on the performance of the detection models.

2.1 Theoretical Framework

A set of critical concepts required to understand the methodology used will be explained in the following subsections. Then, a formal description of the artificial intelligence techniques that were used will follow.

The problem of detecting a phenomenon that occurs within the physical space of reality has been studied within the field of artificial intelligence extensively. Generally, two schools of thought have been researched that can solve the problem of detection: rule-based methodologies and machine learning methodologies. Said differently: deterministic approaches and probabilistic approaches. Both approaches have their appropriate applications and one should consider them for the problem at hand.

2.1.1 Detection

In the context of detection, rule-based methods define a set of rules which, when fulfilled, a phenomenon is detected. These rules can be of the form

$$if < condition > then < resolution > \tag{1}$$

and are typically chained together to form a conjunction of rules which, when satisfied, alert the user of the system that something has been detected. Such systems are optimally used whenever the nature of the phenomenon that is of interest behaves in a deterministic manner or under some natural law that can be modeled. Attempting to find a set of rules that captures a non-deterministic process is one approach for which scientists have coined the expression ”attempting to model the world”. If one takes the battered example of predicting the outcome of a coin flip and decides to solve it using a rule-based approach, one would need to consider the physical rules that act upon the coin and the environment in which it exists. The coin’s paint is modeled, the weight, the size, its initial angle, the force of the flip, the wind friction, etc. Given perfect knowledge, one could build a system that could perfectly predict the outcome of a coin flip. However, information gathering is noisy and perfect knowledge of a phenomenon is seldom obtainable.

The field of machine learning spawned as an alternative approach to world modeling a non-deterministic process. Instead of integrating as many elements that partially explain the phenomenon in the form of rules, machine learning attempts to generalize the underlying distribution of a generative process from a population sample to the actual population. It attempts to understand the influence of random components of the phenomenon which are present in the physical world but not in the data. These random components are named hidden variables or latent variables [7]. Moreover, machine learning also approximates the true distribution of the phenomenon from training samples that typically do not contain the full expected range of values of each component of the full population. At its core, the detection model used in this paper is based on a machine learning subset of models called neural networks. Images taken during endoscopies are inserted into the model and are then broken down by subsequent layers in the neural network to extract features. Given the immense scope of detecting a polyp that can take 1500 different shapes, having an automatic way of representing the different features of the polyp is the only feasible approach. The neural network model is based on the YOLO4 [8] deep neural network architecture.

2.1.2 Types of Learning

Three main paradigms of learning can be distinguished in the literature: supervised, unsupervised, and reinforcement learning. Although the task of detecting a polyp in an image can be represented in all three paradigms, the most fitting one for tackling detection in images is, at the time of writing, supervised learning. Intuitively, the model is presented with two components: images that contain polyps and labels that tell the models where the polyp resides within the image. An iterative process begins during which the model adjusts its internal state to the finite set of images and labels in the hopes that the model generalizes to new never-before-seen images.

2.1.3 Supervised learning

Given a sample x from a population p with classes $c_1, c_2, \ldots, c_n \in C$, we define a supervised learning model M with parameters θ. The model maximizes $p(c|x,\theta)$ where $p(c|x,\theta)$ is read as the probability of class c given a sample x and model parameters θ. c is shorthand for c_n given that the class of x is n. A prototypical mapping between samples and classes is learned by adjusting the model parameters θ iteratively over the training period of the model. θ folds in the shape of the distribution of the samples x_n. The purpose is to obtain a model M with parameters θ that can capture the real distribution only from population samples. The supervised approach allows the model to learn the structural similarities, differences, and features between all samples x of different classes without manually coding any rules ourselves.

2.2 Generative Methods

Typically, much like other supervised machine learning models, generative methods capture distributions from samples fed into them as training data. Unlike other machine learning models, these distributions are learned to generate new samples from the same distribution. The methods include Boltzmann machines, deep belief networks, directed generative nets, autoencoders, and generative adversarial networks among others. This paper focused on the application of generative adversarial networks. For an introductory overview of generative techniques refer to Chapter 20 [9].

2.2.1 Generative Adversarial Networks

A Generative Adversarial Network (GAN) is a model composed of two submodels: a discriminator (D) and a generator (G). In the context of generating synthetic images, each submodel has its purpose. G generates synthetic images by drawing samples from a latent space of arbitrary dimensionality. D distinguishes between real images and synthetic images. With increasing epochs, the generator becomes better at fooling the discriminator. Originally, the training process continued until the discriminator reached a coin flip loss - showcasing its inability to distinguish between real and

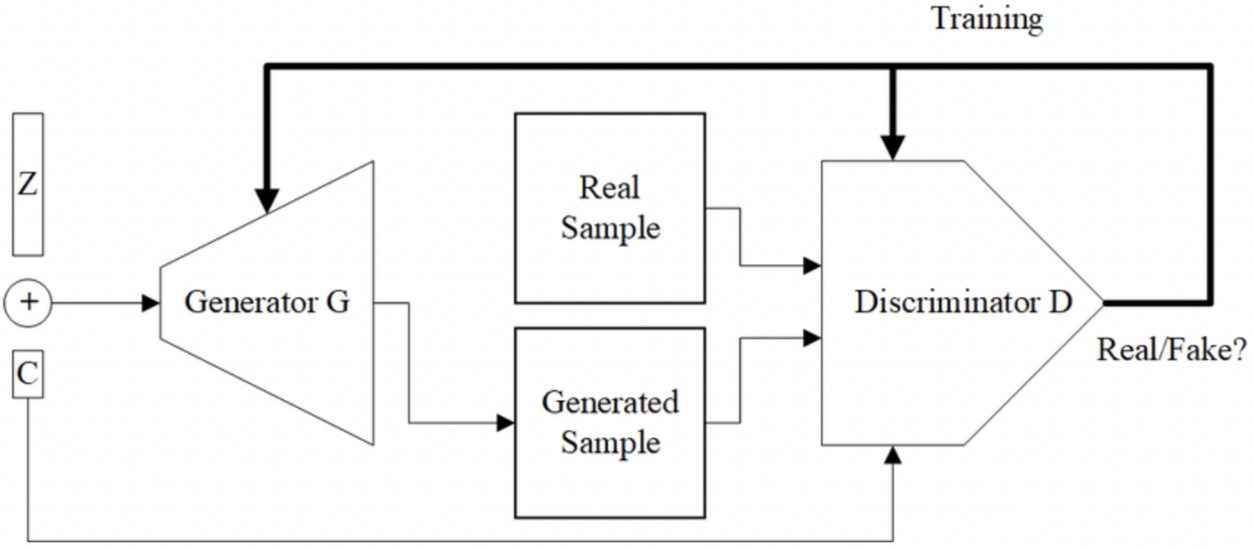


Figure 1: GAN architecture where Z is an $n-$dimensional noise variable that is used to generate samples and C are the optional image class labels. Image is taken from [10].

synthetic images. Other stopping criteria were developed in previous years that proved to obtain better results concerning the network's ability to generate samples closer to the actual distribution [9].

Formally, given data x and the generator's distribution p_g, a prior on input noise variables $p_z(Z)$ is defined. Then, a mapping to data space is represented as $G(z;\theta_g)$, where G is a differentiable function represented by a multilayer perceptron (MLP) with parameters θ_d [11]. The discriminator's distribution, also represented by a multilayer perceptron, is defined as $D(x;\theta_d)$ which outputs a scalar. $D(x)$ outputs the probability that x came from the real data distribution and not p_g. As D is trained, the probability of correctly distinguishing between samples drawn from G or real data increases. As G is trained simultaneously, it minimizes $log(1-D(G(z)))$. This adversarial relationship is defined as a two-player min-max game with value function $V(G,D)$[11]:

$$\underset{G}{min}\,\underset{D}{max}\;V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))]. \tag{2}$$

Instead of training G to minimize $log(1-D(G(z)))$, the model is tweaked to maximize $log(D(G(z)))$. This has been found to obtain stronger gradients during the initial training phases [11, 9]. Sensitive to the direction of the gradients is the crucial learning synchronicity of D and G. The conjoined adversarial model must improve the gradients of both submodels simultaneously during training as to avoid mode collapse by oversampling the same sample from z. This translates to training D for one or more epochs, then the generator for one or more epochs, etc. The generated images become negative examples and are fed into the discriminator as such. Conversely, the discriminator penalizes the generator for producing images that cannot fool it.

2.2.2 Performance Criteria

Monitoring the performance of a generative model is also different from a regular supervised model. The loss of a generative adversarial network does not correlate well with its performance as the loss is computed in an adversarial context where the discriminator gets worse at distinguishing real images from synthetic ones. As such, a decrease in loss does not equate with an increase in image quality.

In the literature, similarity metrics are the most common form of evaluating the performance of generative techniques. Similarity metrics compare the statistical distance between a fake image and a real image. The intuition behind this is that a lower similarity score will correspond with a realistic image since the score represents different statistical properties of the image. If the statistical difference between the two images is low, the low score will highlight a large structural similitude between the two sets of images. The Fréchet Inception Distance (FID) is one such metric that relies on the prediction of the inception during training. It computes the statistical difference between fake and real images by using the mean and standard deviation of the two [12]. Moreover, it is a valid method of determining not only the image quality of the generated images but also the variety of generated images [12, 13, 14].

$$FID = |\mu - \mu_w|^2 + tr(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}). \tag{3}$$

The Fréchet inception distance will be used as an objective quantifier to a subjective question. In determining how realistic an image generated by any method is, the structural similitude between real and fake images will be judged using the FID score and a ranking of the models will be generated based on this score.

2.2.3 Latent Space

The mechanism used to generate new samples will differ from one generative method to the next. For GANs, a probability distribution is learned through modeling a latent space. A GAN then synthesizes data by drawing samples from said latent space. The latent space is defined initially as an arbitrary $n-$dimensional vector where $n \in N$. The number of dimensions depends on the complexity of the problem that the model attempts to learn. During training, samples are learnt and reduced into this latent space where the proximity of two points showcases how semantically similar those two points are. As the network evolves, samples at different coordinates in the latent space will correspond to different features found in the training data.

To exemplify this point, in Figure 35, consider a set of synthetic images of polyps produced in this paper in which we see a linear interpolation between two $n-$dimensional vectors that are next to each other in the latent space z. Top left (0, 0) and bottom right (3, 3) are the vectors and linear interpolations are synthesized between them. As samples are drawn closer to (0, 0), the weight of (0, 0) is stronger and thus has a bigger impact on the output of the photo than (3, 3). Qualitatively, one can interpret this as manipulating significant features that make up the underlying distribution of a colon tract that contains polyps.

Linearly interpolating between two vectors in the latent space z is one of many strategies viable for manipulating the type of image G can generate. The true power comes from being able to play around with different vectors in z to distinguish which vectors influence which parts in the generated image. Having control over the different features allows for tuning the way the resulting image will look like.

2.2.4 Conditional GAN

Given a dataset X a GAN will learn to shape the latent space of the underlying distribution of X and generate samples from that space. Although it is possible to manipulate from which vector in the latent space the sample is drawn from, the distribution learned by the network will conform to the distribution of all samples $x_1, x_2, \ldots x_n \in X$ regardless of the classes of the members of X. As such, class distinctions are not enforced by any means in the learning process which causes class blending. Class blending arises when the model captures the underlying distribution of two or more samples belonging to the population but with distinct classes. In certain applications, class blending proves to be a useful feature of GANs [15]. To generate realistic polyps, however, it may lead to polyps that borrow anatomical structures from different classes. The erroneous nature of this process can be subdued in two ways:

1. Manually manipulate the vectors in the latent space to determine which ones are responsible for which structural elements present in the images.

2. Embed the class y_i alongside sample x_i in the input layer of the network.

The former process leads to extraneous amounts of manual labor as, depending on the complexity of the task, the latent space may be responsible for manipulating 1024 or more dimensions. Determining which combinations of operations within this space cause which anatomical alterations in the polyps are a computationally intensive mapping that can be avoided through the advent of Conditional GANs (CGAN) [16].

The latter option is what defines the conditional part of a conditional generative adversarial network. During training, a class label is embedded alongside the input to capture the class of the input. In doing so, a CGAN is capable of selectively generating samples belonging to different classes in a controlled manner.

To generate a synthetic dataset to be used in a detection model, CGAN can be used to create the ground truth labels of the location of the polyps. The ground truth is embedded in the input layer of the CGAN as four numbers: $(x_0, y_0)(x_1, y_1)$ representing the top left and bottom right points of the rectangle enclosing the desired location of the synthesized polyp. In using this representation, each drawn sample from the latent space will include the desired location of the polyp, avoiding not only a strenuous manual labeling process but also enabling great flexibility to the range of images that we can generate.

2.2.5 Common Issues with GANs

The advent of GANs has allowed for a decrease in computational complexity compared to other generative models. However, several problems arise:

- **Mode collapse** refers to the inability of G to produce a large variety of examples as G overfits on a particular type of example that manages to fool D. Since G maximizes $log(D(G(z)))$ and the generated example manages to fool D, G will produce only that example with small variations early in the training (see Figure **??** in Appendix for an example of mode collapse in our problem).

- **Vanishing gradients** in the generator are common whenever the discriminator is highly performant. To solve this problem, random noise can be added to the labels of D at each training step.

2.3 State of the Art

Following the introduction of GANs by Ian Goodfellow in 2014, NVIDIA has been a pioneer in building models that perform to a photo-realistic capacity. Undoubtedly, these models constitute the current state of the art and will thus be presented in this section.

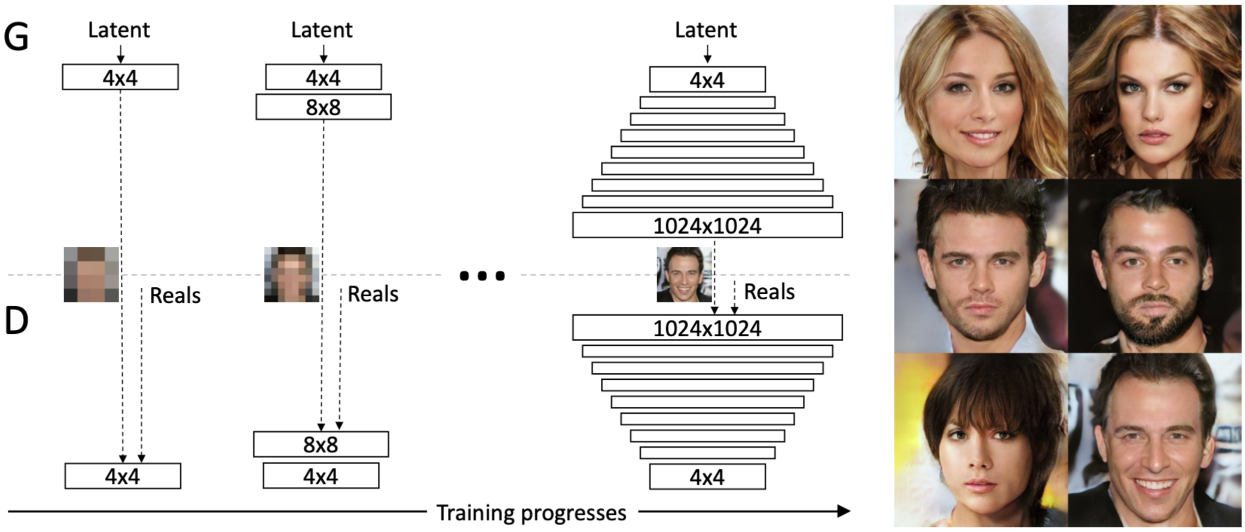


Figure 2: The architecture of Progressive Growing Generative Adversarial Networks (PGGAN). The network starts with a low image resolution and iteratively doubles the resolution during training, resulting in more learning stability and finer details. Figure is taken from [18].

2.3.1 PGGAN

High-resolution images are difficult to generate as the discriminator becomes better at distinguishing real images from fake images since there is more information present in the image that can be extracted into features. Learning stability is decreased on higher image resolutions because of this fact [17]. Moreover, larger images require smaller minibatches due to memory constraints, further decreasing learning instability [18]. Progressive Growing Generative Adversarial Networks (PGGAN) have been introduced to solve the stability and memory issues in synthesizing high-resolution images.

PGGAN grows both the discriminator and the generator as training progresses. The network starts with a low-resolution image to learn large-scale structures and iteratively increases the resolution of both the discriminator and the generator according to Figure 2. The intuition behind this is that lower resolution images are easier for the generator to fool the discriminator and, as higher-resolution layers are added, more details are introduced into the synthesized images. Multi-Adversarial Networks (MAN) inspired this approach by using multiple discriminators for different spatial resolutions [19].

Aside from finer details and stable learning, PGGAN achieves higher variation in the synthesized images using minibatch discrimination [20]. Instead of computing feature statistics from single images, they are computed across different minibatches, enforcing the minibatches across the layers to be statistically similar. A minibatch layer is added towards the end of the discriminator to achieve a mapping between input activation and an array of statistics [18].

Whenever the image resolution is doubled, the model ensures that the transition fades in new layers smoothly as to preserve the gradients and not incur exaggerated landscape jumps. Figure 3 showcases transitioning from 16x16 images to 32x32. As new higher-resolution layers are added in step (b), the lower resolution ones are kept. The weight α of the higher-resolution block increases linearly from 0 to 1. As training progresses, the stability conferred from going from lower-resolution images to higher-resolution images allows for the creation of images with fine details and variation.

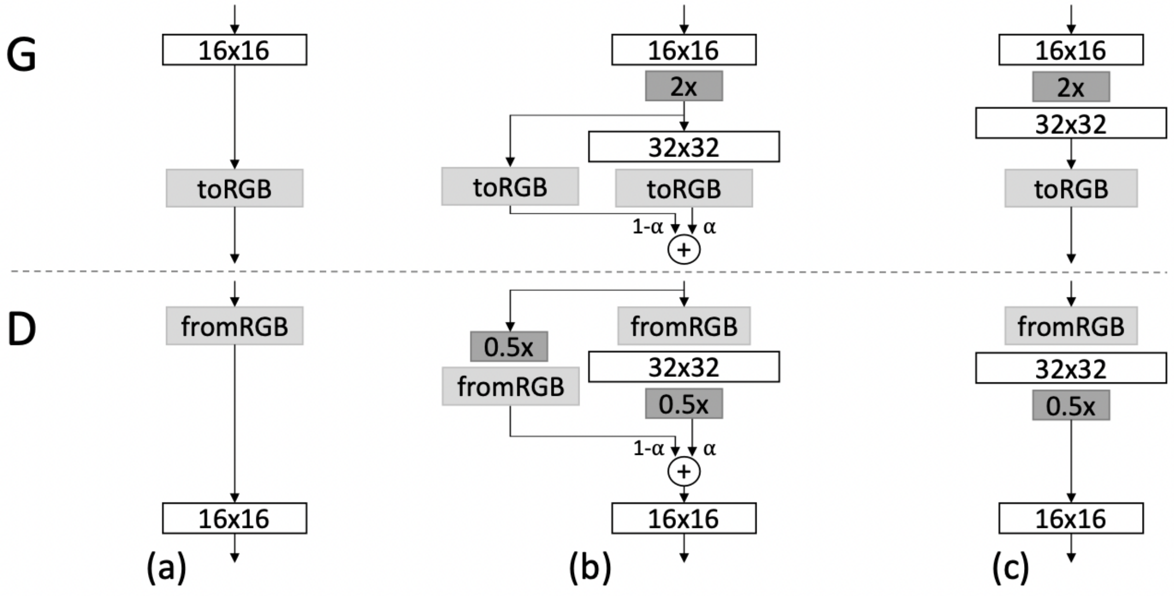


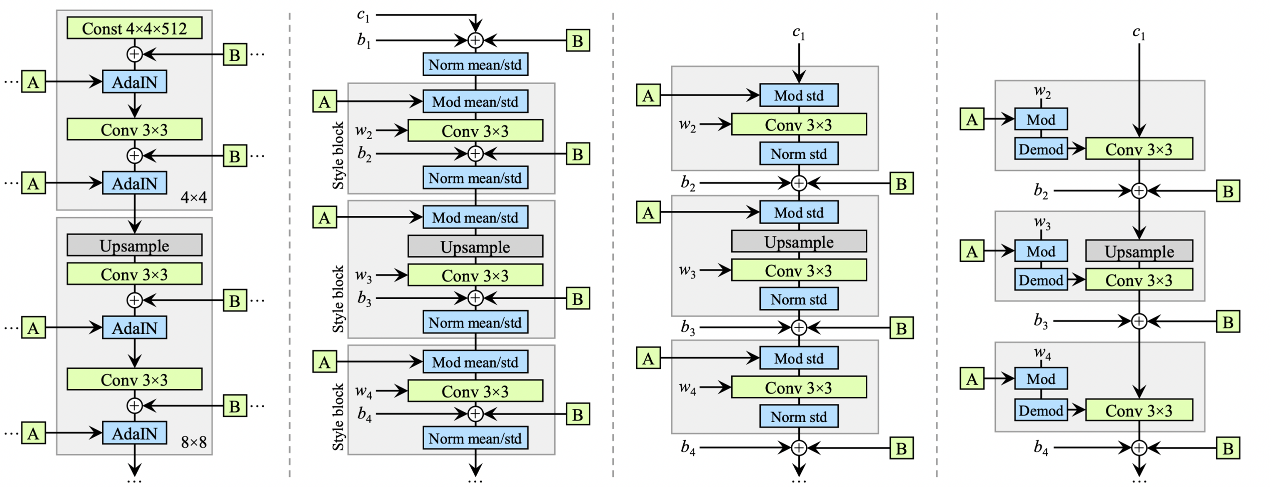
Figure 3: Phases of doubling the image resolution during training. Figure is taken from [18].

2.3.2 StyleGAN2

As an extension of PGGAN and subsequently StyleGAN, StyleGAN2 introduces the idea of manipulating the style of samples drawn from the latent space using an unconventional generative architecture. As opposed to feeding the latent codes $z \in Z$ into the input of the network only, StyleGAN introduces a mapping network f which transforms the input latent codes Z into intermediate latent codes $w \in W$ [13]. In doing so, the latent codes can be manipulated through various affine transformations to systematically affect the style of the generated sample at different layers throughout the network via adaptive instance normalization. Contrast this to having to pick one latent code z that conforms to the desired style and the power of this idea is highlighted: desired styles of classes along with the latent space Z can be applied to any latent code z drawn from the landscape. This allows for greater flexibility in the range of styles the network can produce. It also stabilizes learning as in order to learn n latent codes z in m different styles, the network no longer needs to fold across an $m \times n$ configuration of latent codes. This results in the intermediate space W being less entangled than the input space Z. Additionally, the affine transformations allow for the ability to produce samples of different styles by borrowing the mapped transformations from one latent code to the next.

A detailed overview of the architecture of the network can be inspected in Figure 4. For this paper, StyleGAN2 presented two sought-after properties for the generative task. Firstly, StyleGAN2 reduces the likelihood of learning a distribution that is typical of mode collapse, compared to conventional architectures, as minimizing the adversarial score of one of the models is circumvented through the use of intermediate affine transformations. Secondly, StyleGAN2 produces high-fidelity images, void of major artifacts that were present in previous iterations of the model, by changing the instance normalization procedure with demodulation applied to all weights associated with each convolutional layer [13].

Another major modification done to StyleGAN2 compared to its predecessors is the fact that the architecture remains fixed as learning progresses. PGGAN introduced the idea of adding layers to the network progressively during training. The advantages of this technique were quickly established in the literature as the technique allowed for a more stable learning experience for the network and



(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

Figure 4: Architecture changes from StyleGAN to StyleGAN2. The grey blocks are ”style” blocks and encompass a different combination of operations depending on the architecture. The breakthrough between the initial iteration of StyleGAN and StyleGAN2 was situating the bias and noise (B) outside of the style block. This approach obtains more predictable results compared to having to apply bias and noise within the style block [13]. Image taken from [13].

the convergence towards more realistic images. This was primarily because low-resolution layers promoted the learning of low-level features and, as higher resolution layers were added, the network could learn more high abstract concepts because it had a solid basis of low-level features.

One of the downsides of modifying the architecture during training is that the synthesized images had artifacts that were observed to be occurring systematically because of the progressive increases in resolution during training. The most occurring artifacts were objects being stuck at certain locations in the image that were learned during the initial phases of the learning process. When synthesizing human faces, the low-level features of the eyes would be learned to be placed in a certain location in the image and that location would not change as newer layers were added. Since the output of the network changes each time a new layer is added, layers that are in-between the output and the input of the network have high frequencies which compromise the shift-invariance property of the whole network [21]. In a face generation example, Whenever higher-level features were added, the low-level features of the eyes would appear to be floating or detached from the face which contained other higher-level features. Figure 5 shows the wrongful orientation of a person’s teeth as the camera rotates around the face.

StyleGAN2 introduces a different approach to adding layers progressively to the network during training. The new method is based on the contributions of [22] which introduced MSG-GAN, a network that has as many discriminators and generators as the number of times the image was upsampled or downsampled. The output of a generator for a given resolution would be matched to its corresponding discriminator that accepted images of the same image resolution. Skip connections allowed for the network’s architecture to remain fixed during training while the benefits of reaping features from different resolutions remained by propagating the activation maps of each resolution layer further into the network. The design of MSG-GAN was adapted to include residual connections and, instead of matching n discriminators and n generators to one another via skip connections, StyleGAN2 simply sums up the upsampled contribution of each generator. Each discriminator then receives the scaled-

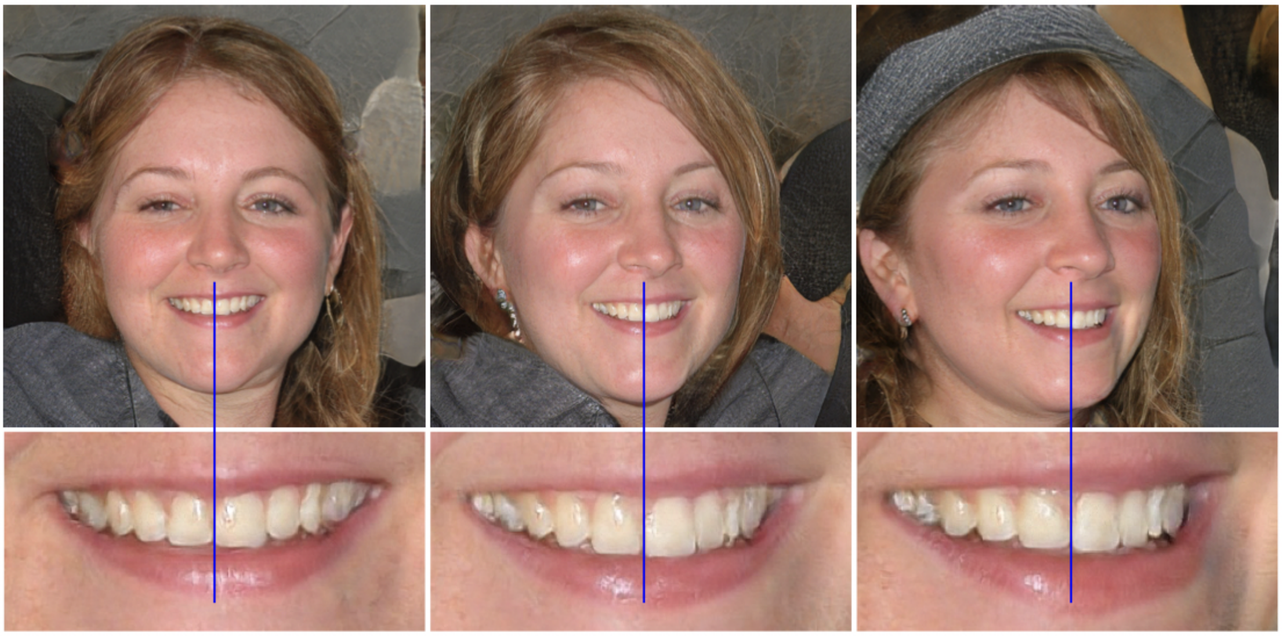


Figure 5: Example of common artifact produced by progressively-growing GANs. The orientation of the person's teeth does not change as the orientation of the head is rotated away from the camera. Image taken from [13].

down version of its preceding discriminator that produces images of a maximum set resolution. In this way, StyleGAN2 saw improved FID and PPL scores compared to progressively-growing architectures and the artifacts that were belonging to these networks were removed.

2.3.3 StyleGAN2-Ada

The last and most promising model to be considered for the generation task is an extension of StyleGAN2, namely StyleGAN2-Ada. This version of the model attempts to reduce the likelihood of overfitting whenever the network trains with a low amount of data. The major trick lies in the addition of a large number of affine image augmentation techniques.

We begin by defining a corruption process c that can be applied to a set of images x. An augmentation technique is said to be *non-leaking* if its corruption process c can be invertible across the data distribution of x. The purpose of image augmentation techniques, at their very core, is to introduce variance in the original data space as a means towards combating overfitting by covering a larger possible data space during training. Instead of evaluating the discriminator on an augmented data set of both real images and images produced by the augmentation process, StyleGAN2-Ada's discriminator is evaluated *only* on augmented images which result from applying non-leaking c's to x. The same is done to the generator during training [23].

The list of non-leaking augmentations that were designed can be consulted in the original paper [23].

3 Methods

Three different generative adversarial networks architectures were used to synthesize images of polyps: a baseline GAN1, PGGAN [18], and StyleGAN2-Ada [23]. From these selected candidate models, only the most promising will be chosen to augment the detection model's training dataset. The candidate models were compared using the Fréchet inception distance (FID) score. The most promising candidate model was the one that obtained the best FID score. Due to time and computational constraints, other model architectures were not able to be tested, however suggestions of such architectures are given in the discussion section of the paper.

3.1 Technical Implementational Challenges

The development of a generative model of synthetic images requires sufficient representative training images. As previously mentioned, a survey of existing datasets in the public domain containing images with polyps has found that there is a scarcity of training data available, limiting the number of usable training images to only 1196.

Since the images have to be annotated with the location of the generated polyp, it is not sufficient to only solve a generative problem, but a data labeling one as well. The detection model gets fed images that contain polyps at a specified location. The locations are manually annotated by medical practitioners. As the network generates new synthetic images, they should label the images with the location of the polyp to avoid manual annotation. A conditional GAN [16] version of the most promising model was developed to generate ground truth labels of the position of the polyp alongside the image.

Preferably, synthesized images should include examples of polyps that occur infrequently in existing data sets in order to most accurately represent the actual distribution of polyps [4]. To accomplish this, a list of underrepresented polyps and challenging angles will be compiled and the generation of said polyps seen from different angles will be synthesized by manipulating the latent space of the generative adversarial networks across the z-vector.

Finally, practical considerations on the computational complexity of the task were taken as the state-of-the-art methods presented in this work take input images that must be at least 416x416 in resolution. Though the neural networks used are embarrassingly parallelizable [24], conducting several parameter sweeps and hyperparameter optimizations across several experiments imposes an impractical computational limitation. All models were trained on a machine equipped with a single NVIDIA V100 GPU.

3.2 Model Comparison

An initial subjective evaluation of non-candidate models was conducted after 10 intervals of 100 epochs. Models that generated seemingly random images were discarded. In the most recent years in the literature, the Fréchet inception distance similarity metric was used as the golden standard for evaluating the performance of GANs. Models that were kept were compared using the FID score. To answer the research question of the paper, the FID score of each model was used to determine a ranking amongst the candidate models. Then, the main research question was answered by judging the relative increase of the detection model's performance after including its synthetic images in the training sets. Ranking them and choosing only the best model to augment the training set of the detection model is a practical way of going about things since, as mentioned before, the FID gives both an indication of image quality and image variety. Meaning that expensive compute power and

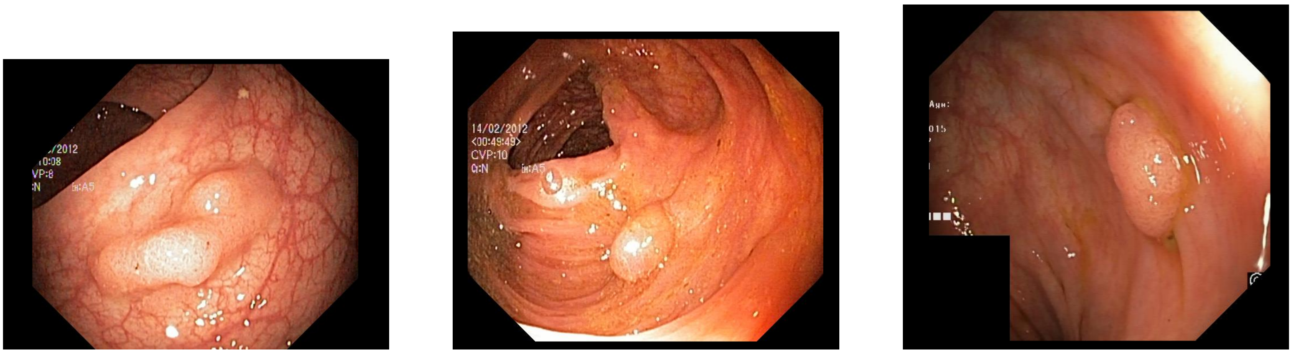


Figure 6: Three random example images taken from the Kvasir-SEG dataset. The images exhibited present protruding polyps; the predominant type of polyp present in this dataset.

time should be put to use in either devising models that obtain higher FID scores or testing only the best-performing model.

3.3 Data

The dataset used to train the generative networks consisted of 1196 images taken during endoscopies that contained at least one polyp photographed under white light imaging. Images within endoscopies that do not contain polyps were omitted as currently there is an imbalance in polyps datasets concerning the number of images that contain polyps versus those that do not. Since the sole interest of the output of the generative model is images that have at least one polyp in them, removing the images that do not contain a polyp makes sense from a machine learning perspective.

3.3.1 Data Collection

The data collection process focused on merging publicly available datasets that matched several criteria. Firstly, the datasets needed to provide ground truth labels with the location in the polyp. Secondly, the images must be frames that taken during endoscopy procedures that contain at least one polyp. Finally, the data labeling procedure between one dataset and the next should be similar enough to the point where the variability of the data labeling is not significantly different than the other as to not compromise the consistency of labels between datasets.
The following table summarizes the public datasets that were used:

dataset	**n_samples**	**bias**	**resolution**	GT
Kvasir-SEG	1000	0/1000	332x487 to 1920x1072	Mask + BB
ETIS-Larib	196	0/196	1225 × 966	Mask

Table 1: **resolution:** image resolutions found in the samples. **bias:** amount of healthy/polyp images. **GT:** ground truth. **Mask:** pixel-level segmentation of the polyp. **BB:** bounding box.

The data collection process resulted in 1196 images of varying resolutions from two datasets. The data labeling process was analyzed for both and, variables such as whether the data labeling team could zoom in on the images, how long they could spend to draw the bounding box, and what to include in the bounding box were the same in both. Controlling these variables is important as the resulting data would be consistent between datasets. The downside however is that being this

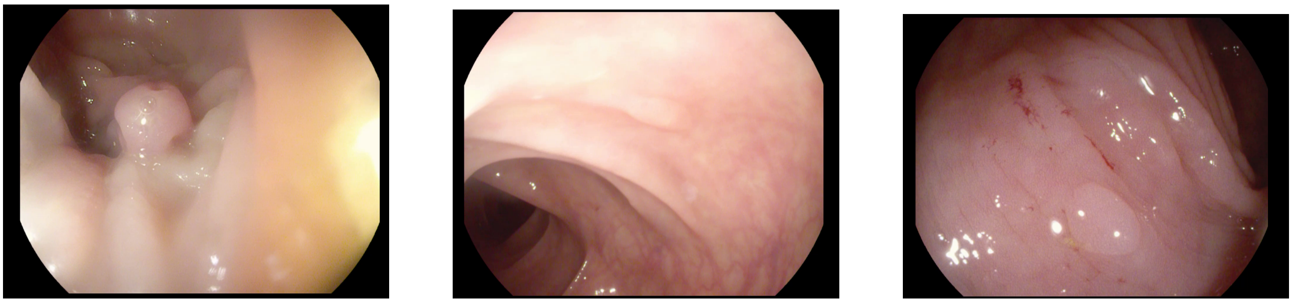


Figure 7: Three random example images taken from the ETIS-Larib dataset. The second and third images showcase polyps that are flat against the tissue; a more difficult type of polyp to detect.

restrictive reduces the number of images that are available and thus decreases the amount of data that can be used for the models. Another downside is the homogeneity present in the data as 80.3% of the data comes from one single dataset of polyps that are homogeneous in texture and shape.

Of particular note is the fact that the two datasets containing 1196 images were used to train the generative networks to generate images and then the detection model was tested on the same dataset with and without synthetic images present in the training split. This is the manner in which the second research question was tested. The detection model used was YOLOv5 in its small weights configuration.

3.3.2 Data Preprocessing

Each image from the dataset was resized to a minimum resolution of 416x416, the minimum acceptable image resolution for the detection model. Some models generated 512x512 images, a fact that is explicitly stated for experiments where this is true. The images were normalized such that each pixel value is in the range [-1 1]. Non-destructive affine image augmentation techniques were applied to the original datasets: rotations +-15°, horizontal and vertical flips. For the automated data labeling process, image augmentations that changed the orientation of the image were accounted for when computing the coordinates of the bounding box that defined the location of the generated polyp.

In StyleGAN2-Ada, built-in affine image augmentations were applied to the original data sets in place of our existing affine transformations [23]. This resulted in a corruption process pipeline that was experimentally found to significantly increase the variability and quality of images [23].

3.3.3 Data Labelling

The training sets contain the location of the polyp represented as either a bounding box defined by a pair of two coordinates (top left corner and bottom right corner) or as a segmentation map defining the topology of the polyp as can be seen in Figure 8. The location of the polyp was used during the training process as a feature extraction method to crop the polyp and feed it into the generative methods.

3.4 Models

The present paper analyzed the potential of three separate architectures: a baseline GAN, PGGAN, and StyleGAN2-Ada. Within each architecture, several parameter configurations were tried to determine their influence on the resulting images.

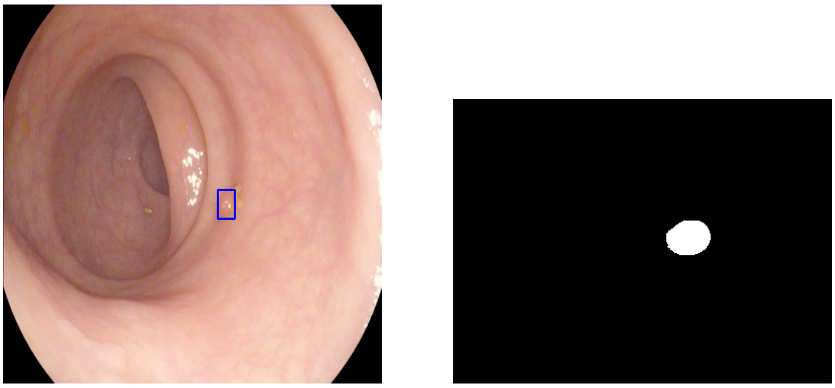

(a) Bounding box label. (b) Segmentation map label.

Figure 8: Ground truth representation used in training sets.

3.4.1 Baseline GAN

A baseline generative adversarial network was created to solve the synthesis task. The architecture of the network remained fixed during training and consisted of one discriminator D and one generator G. Within this architecture, several parameters were changed between experiments to determine the best configuration. Table 2 shows each parameter of the GAN and hyperparameter alongside the ranges in which they were tweaked between experiments.

3.4.2 PGGAN

PGGAN was used as one of the candidate architectures to solve the synthesis problem. Table 3 lists the parameter sweep conducted during experiments to determine the most performant configuration of parameters.

3.4.3 StyleGAN2-Ada

StyleGAN2-Ada was used as the final candidate model to solve the synthesis task. The adaptive discriminator augmentation version of StyleGAN2 was chosen as the experiments had to be conducted using a low number of images (1196). Table 4 lists the parameters that were swept during the experiments to determine the optimal configuration.

3.4.4 Conditional SyleGAN2-Ada

The labels provided with the dataset were embedded into the input layer of the network as two coordinates representing the top left and bottom right corners of the bounding box for each image. Bounding boxes were then generated by the network and sent to the generator to specify where to create the polyp. The network configuration is the same as StyleGAN2-Ada with the addition of this embedding operation.

3.4.5 YOLO5 Detection Model

The YOLOv5 detector and classifier were used as the main polyp detector. The model ran for 100 epochs training on the real dataset and the augmented datasets. The performance of the model was given by the mean average precision (mAP) obtained after 100 epochs of training using an intersection

Param	**Range**	**Description**
width	128, 256, 416, 512	Image width
height	128, 256, 416, 512	Image height
l-dims	128, 256, 512, 1024	Latent dimensions
d-lr	0.0001, 0.0002, 0.0003, 0.0004, 0.0005	*D* learning rate
g-lr	0.0001, 0.0002, 0.0003, 0.0004, 0.0005	*G* learning rate
lr-step	1, 10, 100, 999999	Learning rate step
label-noise	0, 0.05	Noise added to *D* labels
d-filters	16, 32, 64, 128, 256, 512	Nr. of *D* filters
d-kernel-sizes	3, 4, 5	Size of *D* filters (n^2)
d-strides	1, 2	*D* filter stride
d-n-down-blocks	`len(d-filters)`	Nr. of downsampling blocks
d-lrelu-alpha	0, 0.2	*D* leaky ReLU Alpha
d-dropout	0, 0.3, 0.5	Dropout probability
g-filters	16, 32, 64, 128, 256, 512	Nr. of *G* filters
g-kernel-sizes	3, 4, 5	Size of *G* filters
g-strides	1, 2	*G* filter stride
g-n-up-blocks	`len(g-filters)`	Nr. of upsampling blocks
g-lrelu-alpha	0	*G* leaky ReLU Alpha
aug	True, False	Whether geometric augmentations were on

Table 2: The set of parameters that were swept between experiments for the baseline model containing discriminator *D* and generator *G*.

Param	**Range**	**Description**
d-lr	0.001, 0.0001, 0.0002, 0.0003, 0.0004, 0.0005	*D* learning rate
g-lr	0.001, 0.0001, 0.0002, 0.0003, 0.0004, 0.0005	*G* learning rate
lr-step	1, 10, 100	Learning rate step
aug	True, False	Whether geometric augmentations were on

Table 3: The set of parameters that were swept between experiments for PGGAN. Most parameters were kept at the default values presented in the original paper [18]. The learning rates were adjusted to determine whether quicker convergence could be obtained to the domain of applicability of synthetic polyps.

Param	**Range**	**Description**
aug	True, False	Whether geoemtric augmentations were on

Table 4: All of the default values presented in the original paper [23] were preserved and used.

over union (IoU) threshold of 0.5. Table 5 summarizes the hyperparameters used for all experiments. Table 6 summarizes the models used during the experiments. 60%/20%/20% of real images were split into training/validation/test sets respectively for all experiments, whilst 80%/20% of fake images were split into training/validation sets for the experiments that included fake data.

The model with the lowest FID score also underwent a manual pruning process in which each image was inspected and removed if it satisfied any of the following criteria:

- Area within bounding box contains artifacts
- Bounding box covers less than 50% of the polyp
- Bounding box covers the vignette edges of the image
- No polyp in the bounding box

The manual pruning process was done to determine whether feeding the model the best examples of the best model did increase the performance of the detector. From 1000 images, 575 were pruned from the best performing model, yielding a total number of images to use for augmentation of 425.

Param	**Value**
weights	yolov5s.pt
lr0	0.01
lr1	0.2
momentum	0.937
weight_decay	0.0005
warmup_epochs	3.0
warmup_momentum	0.8
warmup_bias_lr	0.1
box	0.05
cls	0.5
cls_pw	1.0
obj	1.0
obj_pw	1.0
iou_t	0.22
anchor_t	4.0
fl_gamma	0.0
hsv_h	0.015
hsv_s	0.7
hsv_v	0.4
degrees	0.0
translate	0.1
scale	0.5
shear	0.0
perspective	0.0
flipud	0.0
fliplr	0.5
mosaic	1.0
mixup	0.0
copy_paste	0.0

Table 5: YOLOv5 hyperparameters used for detecting polyps. For a detailed description of each parameter, consult the documentation.

Model	**Epochs**	**Number of fakes**	**Manual Prune**
No Fakes	N/A	0	N/A
StyleGAN2-Ada	2400	500	False
StyleGAN2-Ada	2400	1000	False
StyleGAN2-Ada	2400	425	True
StyleGAN2-Ada	3200	500	False
StyleGAN2-Ada	3200	1000	False

Table 6: Table of experiments conducted to determine the relative increase in detection performance. Models were selected after running the experimental pipeline as described in section 4. **model:** which model was used to synthesize fake polyps. **epochs:** number of epochs the model was trained for. **number of fakes:** how many fake images were inserted into the real images train/val sets. **manual prune:** whether a manual inspection of the generated images was conducted to remove images containing artifacts.

4 Experimental Setup

The task of generating synthetic images to be used for data augmentation was carried out in a structured data pipeline involving multiple possible permutations of candidate models, data sets, and a variety of preprocessing methods applied to the data. All three components within this structured data pipeline are extensively presented and discussed in the following sections. As an overarching theme, the task of developing synthetic images to improve a detection model requires the development of both a generative model and a detection model. The detection model used to test the detection of polyps was YOLOv5, the latest version of the YOLO architecture [8]. YOLOv5 was picked due to its preexisting weights which allowed to test detection performance on our limited in a relatively fast manner compared to other detectors. It is also the architecture upon which ZiuZ' detector is based on. The inner workings and intricacies of YOLOv5 or Zius' model will not be deliberated in this paper; for details about both detectors please refer to either the original YOLO paper [8] or the authors at Zius.

The experimental setup is branched into two approaches: data-fixed and model-fixed. In a data-fixed setting, the models are tweaked between experiments whilst the input data for the models remains the same. In model-fixed, the models stay the same whilst the data gets augmented or replaced in some manner. Within data-fixed or model-fixed, the generative models are trained on the data and the output is visually inspected after every 100 epochs. The inspected output consists of a 4x4 grid containing 16 samples of generated images. Having fixed some data configuration and some model, we refer to a (model, data) pair as a model that will run in an experiment. The model is then determined to be a **candidate model** or not. The process is repeated for all models configurations as seen in Figure 9. As for the model-fixed paradigm, since data was scarce, the only model-fixed experiments that were conducted included image augmentations of affine transformations on the original data.

The process of determining a candidate model lies in the subjective evaluation of the output where the discriminator function that separates candidate models from non-candidate models discerns whether the output is nonsensical. Each model was trained for a minimum of 1000 epochs. This benchmark was chosen based on empirical evidence showcasing gradient convergence within 1000 epochs.

The output of each candidate model was then compared objectively to the real dataset the network is trying to imitate by measuring the Fréchet inception distance of each model. The model with the lowest FID score was deemed the winner as seen in Figure 10. That model then augmented the detection model to determine whether a relative increase in the detection model's performance could be seen as described in Figure 11.

4.1 Tools and Technologies

The models were developed in either Tensorflow or PyTorch using Python 3.7. Data preprocessing was conducted using standard data science libraries: Pandas, NumPy, CV2, SciPy. The models were trained on the Peregrine cluster belonging to the University of Groningen on one NVIDIA V100 GPU. Multiple GPUs were not available and this represented one of the limitations of the study.

4.2 Performance Criteria

The performance of the winning network was measured by how much the generated images improve the performance of the detector. The detector was evaluated using the mean average precision (mAP) metric with an intersection over union threshold (IoU) of 0.5 (4).

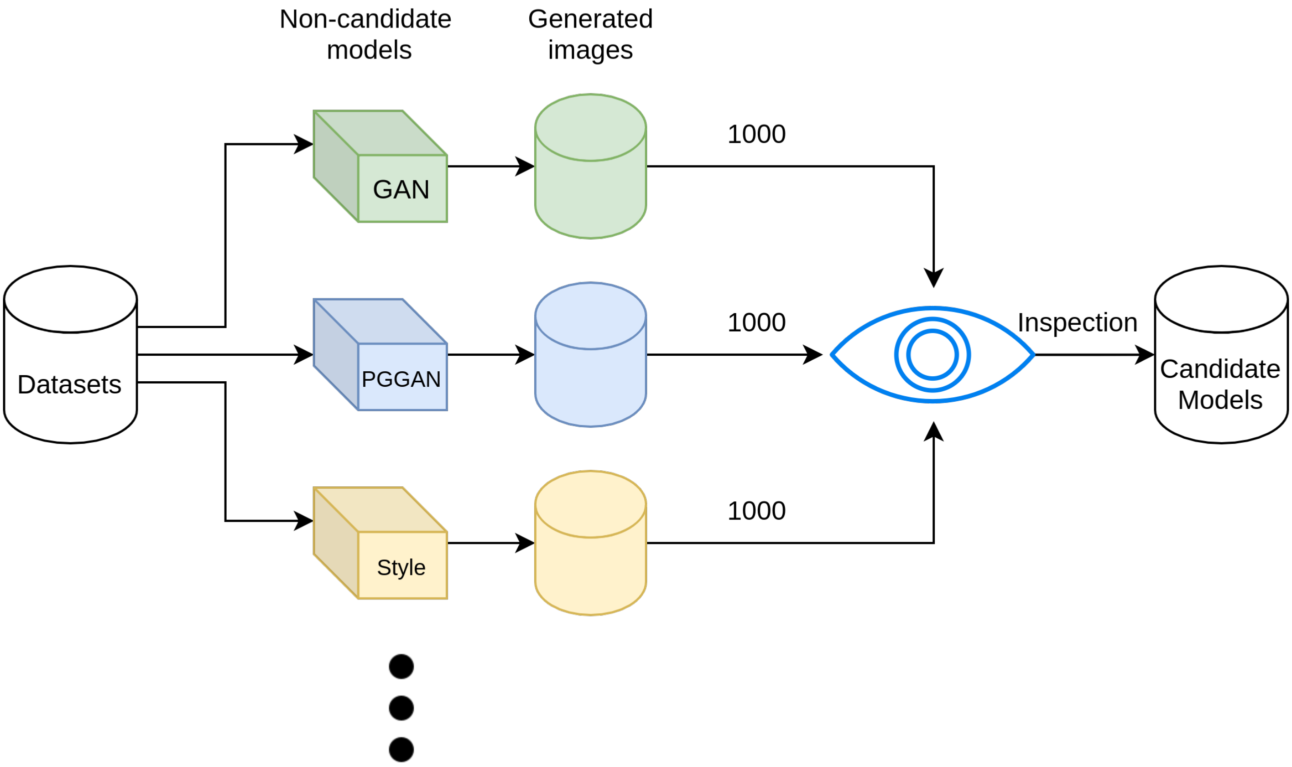


Figure 9: Experimental setup for selecting candidate models from non-candidate models in a data-fixed approach. Each model was trained for a minimum of 1000 epochs and generated 1000 images that were then visually inspected. This inspection occurred in increments of 100 epochs to determine the general convergence direction of the weights. If the inspection concluded that the result was nonsensical, non-candidate models did not get promoted to candidate models. Examples of nonsensical images of non-candidate models can be seen in the appendix 19.

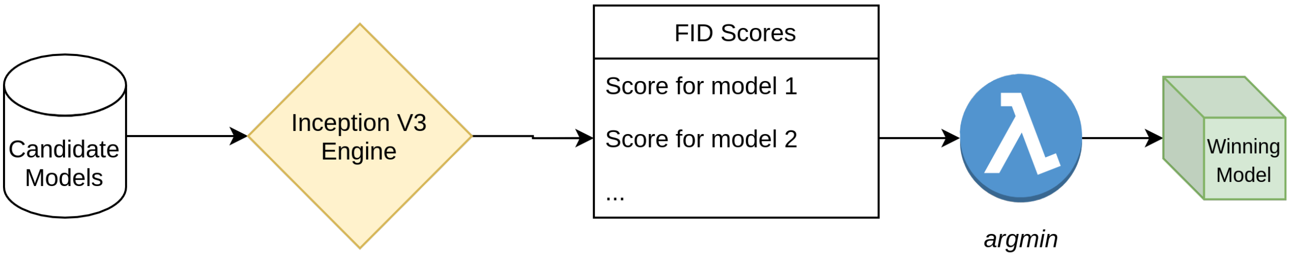


Figure 10: All candidate models generated 1000 images that were then plugged into the pre-trained Inception V3 network. The network produced an FID score for each set of images originating from each model. The scores were ordered and the model with the lowest FID score was deemed the winner.

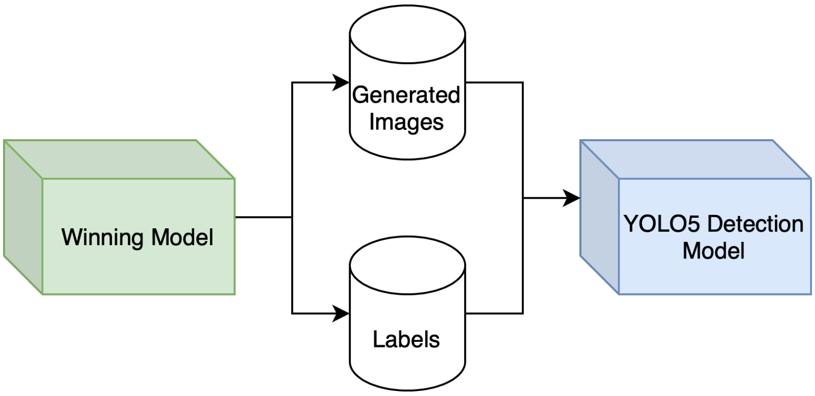


Figure 11: The model with the lowest FID score was deemed the winner and was tested on its ability to improve the detection rate of the detection model after augmenting the detector's training set with generated images and corresponding labels. We remind the reader that the labels are also synthesized by this network.

$$mAP = \frac{\sum_{i=1}^{I} AvgP(i)}{I} \tag{4}$$

where i is an image, $AvgP(i)$ is the average precision score obtained for image i and I is the number of images. Since $IoU = 0.5$ the mAP@0.5 will represent the average precision obtained for each image where the area of intersection over the area of union of the predicted bounding box and the ground truth is above 0.5.

Sets of generated images that manage to increase the mAP obtained by the detector with no augmentations will positively answer the research question of this paper.

5 Results

The relative increase in performance for the winning candidate model is reported at the end of this section. Images of non-candidate models are presented in the conducted order of the experiments as meaningful insights of the parameter space could be drawn from running these experiments. The FID scores of the candidate models were computed and reported.

5.1 Images

5.1.1 Baseline GAN

The following table showcases the different configurations used for training the baseline GAN. The parameter sweep from these experiments yielded non-candidate models but also showcased promising interactions between parameters.

aug	k-size	depth	l-dim	res	id	epochs	change from previous	ref
False	[4 3]	2	128	128x128	1	1000	**baseline**	19
False	[4 3]	2	256	128x128	2	1000	double ldims	20
False	[4 3]	2	512	128x128	3	1000	quadruple ldims	21
True	[4 3]	2	128	128x128	4	1000	image augmentations	22
True	[4 3]	2	128	416x416	5	1000	desired resolution	23
True	[4 4 3]	3	128	416x416	6	1000	increase k-size	24
True	[4 3]	2	512	416x416	7	1000	increase ldims	25
True	[4 4 3]	3	512	416x416	8	1000	increase k-size	26
True	[4 4 3]	3	512	416x416	9	2000	increase epochs	27

Table 7: Examples of baseline GAN non-candidate models. **Aug:** whether image augmentations are applied to the training set. **k-size:** the size of the discriminator kernels ordered from input to output. **depth:** number of hidden layers used on the discriminator. **l-dim:** number of latent dimensions used to represent the latent space. **res:** output resolution. **id:** corresponding image id in the appendix. **change from previous:** what was changed from the previous experiment. The reasoning behind each change is included in the appendix for every corresponding experiment. **ref:** link to image in appendix.

Firstly, affine image augmentation operations were found to be a crucial addition as the number of images fed into the network from the original datasets is only 1200. For experiments where image augmentation was turned off, the network converged into mode collapse as is most prominently seen in Figures 19 and 20. As such, image augmentation proved to be crucial in order to avoid mode collapse and for the models to converge to promising results.

Secondly, increasing the image resolution but keeping the number of hidden layers in the discriminator fixed did not translate in qualitatively similar results between one resolution and the next. Since the discriminator shifts a number of kernels of a particular size across the image, the features needed to capture the intrinsic characteristics of a polyp will be scaled. As such, modifying the number of layers as well as their size was a crucial requirement when increasing resolution.

Thirdly, the number of latent dimensions was shown empirically to depend on the image resolution. The higher the number, the more complex features could be represented in the space. When increasing the resolution, the granularity of the features propagated changes As such, modifications not only at the level of the kernels need to be made, but also to the number of latent dimensions used to represent the data. As a rule of thumb, the higher the resolution the higher the number of latent dimensions required for an appropriate representation.

The candidate model was determined after numerous experimentation, the nature of which resembling that exhibited in Table 7. The candidate model configuration can be seen in Table 8.

Param	**Value**
aug	True
k-size	[3 3 3 3 3]
depth	5
l-dim	512
d-lr	[0.0005 0.0004 0.0002 0.0001]
d-filters	[32 64 128 256 512]
d-strides	[2 2 2 2 2]
d-lrelu-alpha	0.1
d-dropout-p	0
g-lr	[0.0005 0.0004 0.0002 0.0001]
g-filters	[512 256 128 64 32]
g-strides	[2 2 2 2 2]
g-lrelu-alpha	0.1
lr-steps	[1 10 100 999999]
res	512x512
epochs	7900

Table 8: Best-performing baseline GAN configuration yielding the baseline GAN candidate model.

5.1.2 PGGAN

The results of the best-performing model based on PGGAN can be inspected in Figure 30. Since PGGAN incrementally adds layers of increasing resolutions during training, it became infeasible to continue training after 9220 epochs on our current hardware, so no other results could be obtained.

Param	**Value**
g-lr	0.001
d-lr	0.001
lr-step	1
aug	True

Table 9: Best-performing PGGAN configuration yielding the PGGAN candidate model.

5.1.3 StyleGAN2-Ada

The results of the best-performing model based on StyleGAN2-Ada can be inspected in Figures 31, 32, and 34. The images were created using the same model after 2400 epochs and 3200 epochs of training respectively. These epoch checkpoints were selected based on the observation that GANs tend to become worse as training progresses after a certain point. As such, both models were used to generate images as a subjective distinction between the two could not be made.

5.2 FID Scores

The most promising candidate models from each architecture were compared head-to-head concerning their FID scores. The results of the FID computation can be seen in Table 10. Out of the three candidate models, StyleGAN2-Ada had the lowest inception score, meaning that it produced the most realistic and varied images of the three candidate models. During the comparison, we noticed that the model's performance degraded as training progressed. This effect can be seen by the higher FID score obtained at 3200 epochs compared to 2400 epochs.

Although the FID of 3200 epochs was higher, both the 2400 and 3200 epoch models were used to test the augmentation hypothesis as the outputs of both models were indistinguishable to the human eye.

Model	Epochs	FID	Manual prune
Baseline GAN	7900	191.30	False
PGGAN	9220	93.71	False
StyleGAN2-Ada	3200	72.49	False
StyleGAN2-Ada	2400	62.11	False
StyleGAN2-Ada	**2400**	**51.03**	**True**

Table 10: FID scores obtained for implemented models. **Manual prune** refers to whether the generated images were then manually inspected to remove images that contained artifacts. The winning model of all architectures is StyleGAN2-Ada after 2400 epochs of training and manual pruning.

5.3 Conditional StyleGAN2-Ada

StyleGAN2-Ada was tuned to embed the desired location of the polyp in the input layers. Figure 12 shows examples of generated images where the polyp should be placed in the bounding box.

5.4 YOLOv5 Detector

The mean average precision of all winning models is reported in the following graphs and Table 11.

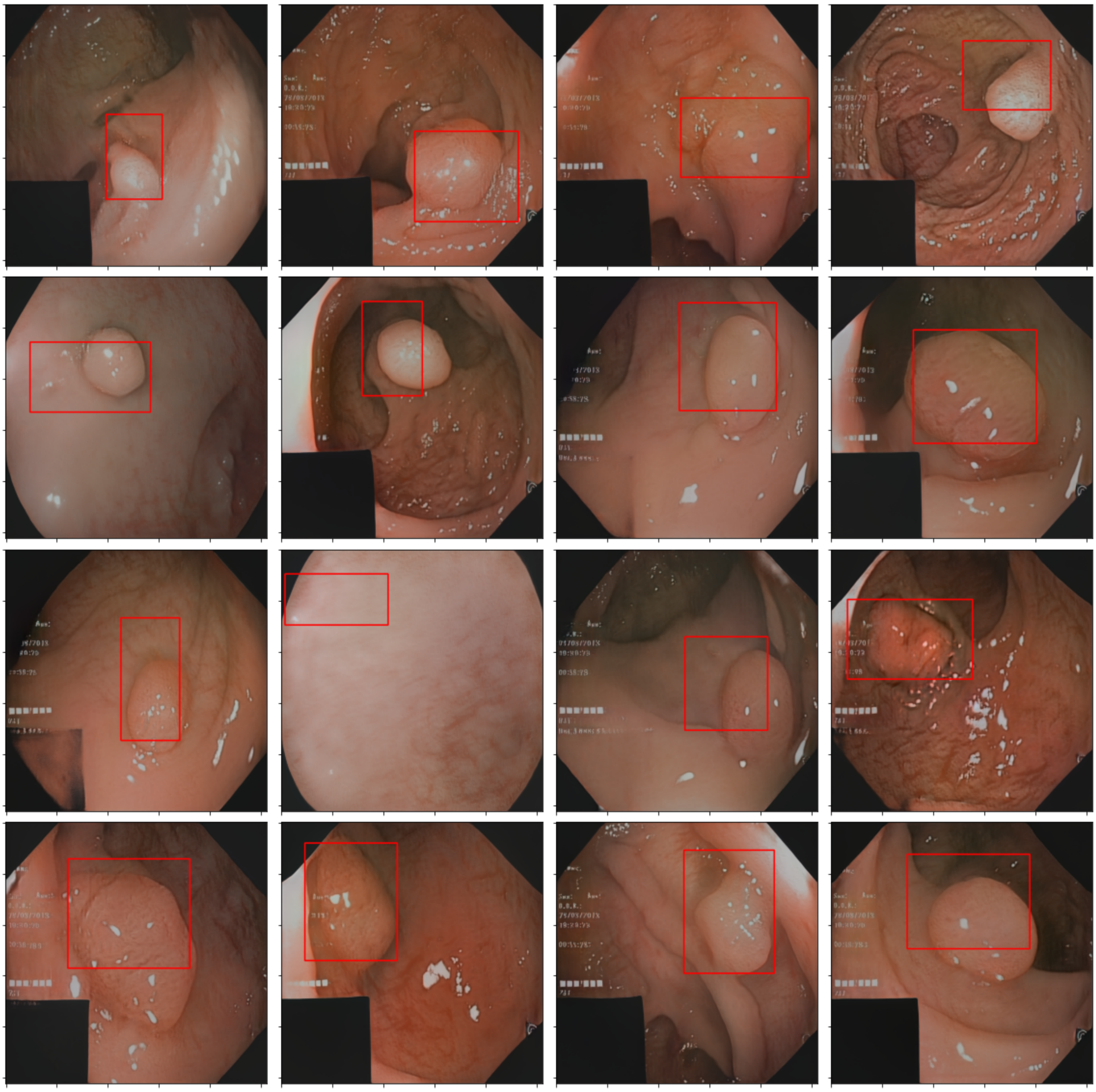


Figure 12: Conditional version of StyleGAN2-Ada generating polyps within the given bounding box location. The bounding box is specified first and then ideally the polyp is generated within the bounding box.

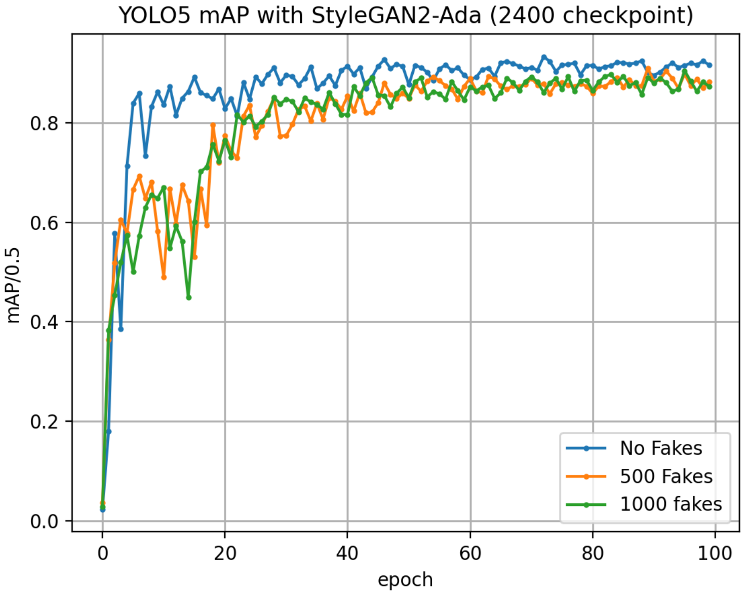


Figure 13: YOLOv5 mAP comparison between no augmentations added to the training set and 500/1000 fake images added from the StyleGAN2-Ada model after 2400 epochs.

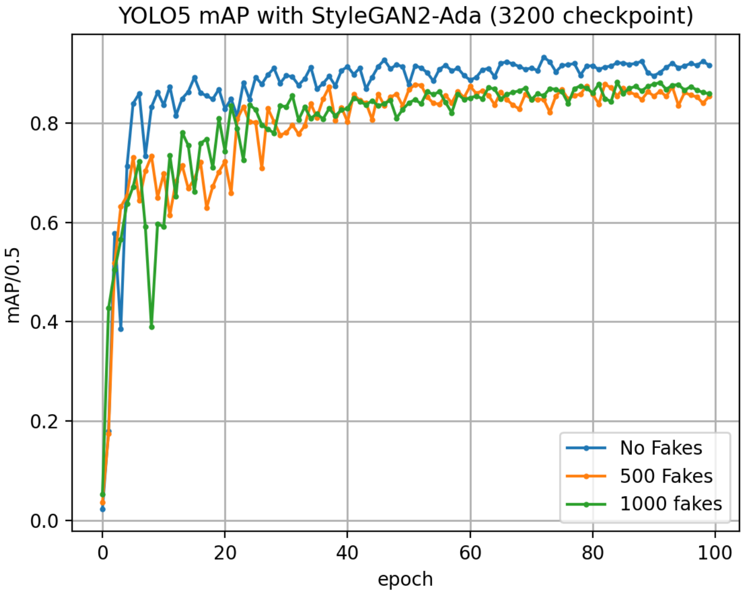


Figure 14: YOLOv5 mAP comparison between no augmentations added to the training set and 500/1000 fake images added from the StyleGAN2-Ada model after 3200 epochs.

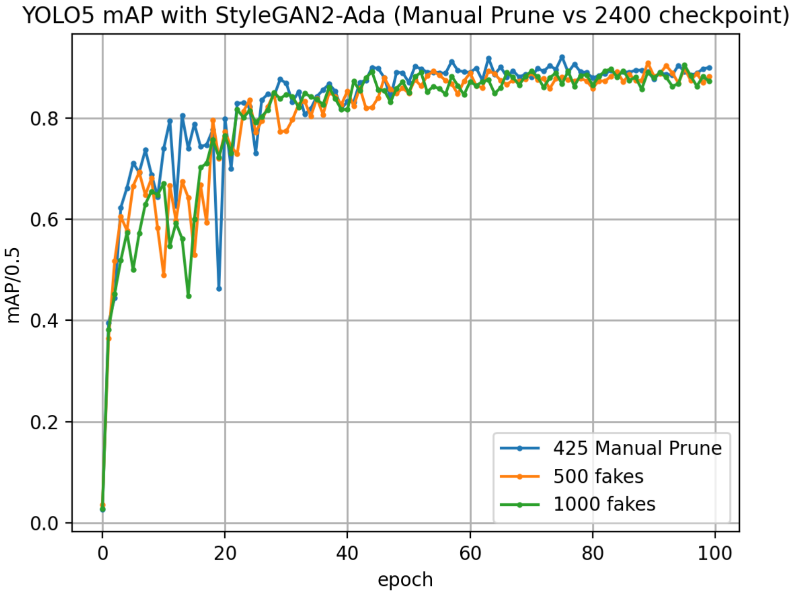


Figure 15: YOLOv5 mAP comparison between 500/1000 fake images from StyleGAN2-Ada after 2400 epochs and manually pruned images from the same model.

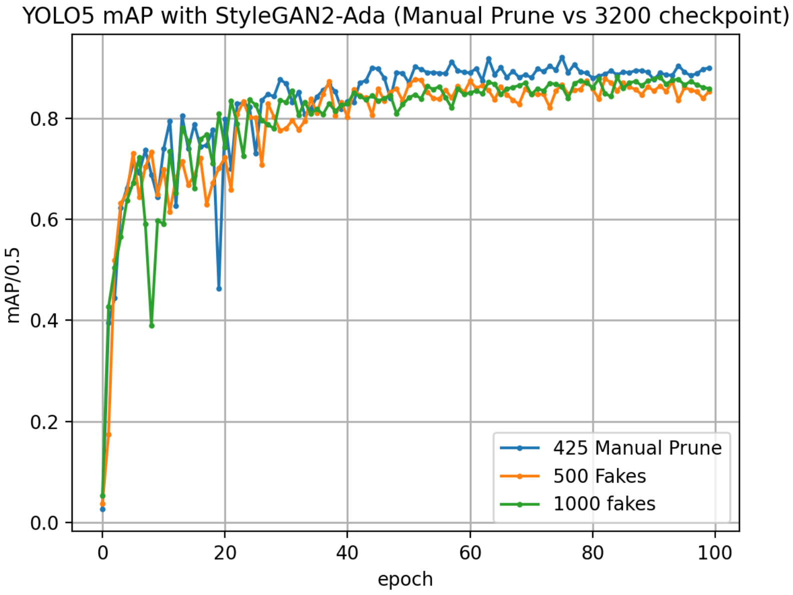


Figure 16: YOLOv5 mAP comparison between 500/1000 fake images from StyleGAN2-Ada after 3200 epochs and manually pruned images from the same model.

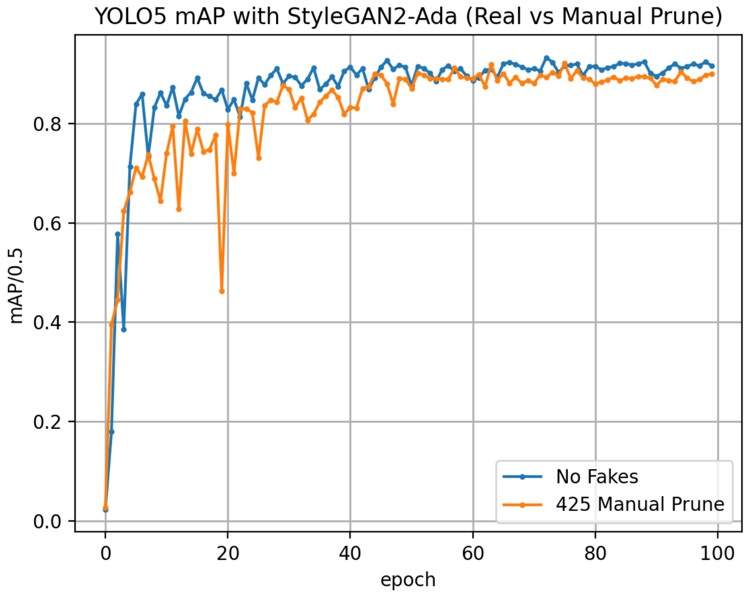


Figure 17: YOLOv5 mAP comparison between no augmentations and manually pruned set of fake images generated by StyleGAN2-Ada after 2400 epochs.

Model epoch checkpoint	Number of fakes	mAP	Manual Prune
N/A	**0**	**0.92448**	**N/A**
StyleGAN2-Ada 2400	500	0.90937	False
StyleGAN2-Ada 2400	1000	0.90466	False
StyleGAN2-Ada 2400	**425**	**0.9213**	**True**
StyleGAN2-Ada 3200	500	0.87794	False
StyleGAN2-Ada 3200	1000	0.88195	False

Table 11: Mean average precision scores for varying degrees of augmentation obtained by YOLOv5. StyleGAN2-Ada with manual pruning yielded the largest mAP among fake augmentation experiments whilst no augmentation proved to yield the largest mAP overall.

6 Discussion

We summarize the paper by providing a couple of discussion points from both medical and machine learning perspectives to determine the current state of the systems and potential future directions given the obtained results. A discussion of the results and an attempt to answer the proposed research questions based on the results follow.

6.1 Research Question 1: Image Realism

The first objective of this study was to determine whether the generation of realistic synthetic polyp images would be possible. Three networks were built and evaluated based on subjective analysis and an objective similarity metric.

Of the three models, StyleGAN2-Ada obtained the lowest FID score, suggesting that the statistical difference between the generated images and the original training set would be the lowest of all models and that the generated images would be the most realistic of the three. From a subjective standpoint, the images seen in 31 and 32 contain anatomical properties that can be observed in the real training set such as protruding polyps, tissue, and even elements from the UI of the endoscope such as the date of the procedure and the vignette of the camera. On the other hand, image artifacts such as shearing, noise, and black spots can be observed in a subset of the generated examples.

It is clear that given a sufficient amount of data and the right architecture, image realism can be obtained in synthetic endoscopy images using GANs. The results obtained using StyleGAN2-Ada were most promising as several accounts from medical practitioners reported that they believed the images to originate from real endoscopies. Moreover, we ran a trained polyp detection model from ZiuZ on 30 random synthesized images generated by StyleGAN2-Ada (2400 epochs) and, for 83% of them, the model reported the existence of a polyp with at least 22% confidence. 22% confidence was chosen as the optimal confidence discriminator value for establishing the existence of a polyp for this particular model.

Given the obtained results, we are happy with the degree of realism exhibited by the most promising model.

Examples of the images that managed to fool this detector as well as the bounding boxes estimated by this detector for our generated images can be seen in Appendix section C.

From a quantitative standpoint, StyleGAN2-Ada obtained the lowest FID score; meaning that it was the closest to being identical to the real images. However, the FID score makes sense relative to other generative models and not as an absolute metric since it can be arbitrarily high. As the concept of realism is subjective and inherently not quantifiable, it is not possible to unequivocally answer the first research question from a quantitative perspective. What can be said is that StyleGAN2-Ada generated some images that contained, what several medical practitioners claimed to be, polyps. As such, we are happy with the obtained results. The results obtained were promising in that most of the generated images contained polyps and, given several improvements to the experimental setup, newer candidate models may be built to attempt to beat the current model. A benchmark has been set.

We propose the following improvements to the experimental pipeline to increase image realism and variety:

- increasing the number of images to train on
- increasing the variety of polyps present in the dataset
- increasing the variety of angles and lighting conditions

- improving the CGAN architecture to produce labels that cover the entirety of the polyp consistently
- building different GAN architectures

6.2 Research Question 2: Detection Increase Through Synthesis

The second objective of this study was to determine whether the inclusion of synthetic images into existing datasets improved the performance of automated polyp detection systems.

The winning generative model showed near-equal or slightly lower mAP scores during testing. The decrease in mAP could partially be attributed to artifacts present in some of the produced images. We showed that by manually removing problematic images, the performance of the detector trained on the augmented set was equal to the performance of the detector trained on the non-augmented set. Therefore, although the synthesized images did not increase the performance of the detection model, improvements in the automated generation of endoscopy images might still lead to an increase in mAP.

6.2.1 Automated Data Labeling

One limitation of the current model is that the generated images need to be manually inspected to remove images that contain artifacts. More often than not, the artifact that occurs the most is a wrongfully generated label covering the polyp only partially and including part of the background as well. On such common occasions, images do contain high-quality polyps, but they are improperly labeled by the conditional GAN.

At least two approaches exist for this problem:

- A binary classification method for determining whether a generated image contains an artifact could be developed to act as a filter for the generator. The method could consist of a separate discriminator that gets fed images that contain artifacts and images that do not and, in the same adversarial fashion, would compete with the existing StyleGAN2-Ada generator.
- Improving the overall performance of the network to deter it from generating artifacts in the first place would constitute an obvious solution to the problem. Seeing as the most promising generative model that was not manually pruned was within a 2% margin of the performance of the manually-pruned dataset, optimizing the model and including more images in its training set could be the necessary forces it needs to surpass the baseline of 92%.

Regardless of the solution, fixing the labels and excluding background information to focus the bounding box only on the polyp would shift the problem from an improper automated labeling technique to increasing the quality of the features fed into the model.

6.2.2 Latent Space Interpolation

The second limitation is that the generated images do not contain sufficient variability for the type of polyp that gets created. Since the generative models are trained on a relatively low set of homogeneous images for deep learning models, synthesized images will also be homogeneous. One way to avoid this is to manipulate the generative process in some controlled manner. Exploring the latent space of the network to determine which vectors z influence certain properties of the polyps could yield a mapping from vector to property. This mapping could then be used to influence the output of the network to match any combination of properties that the user wants. An example of manipulating the style of a polyp using latent vector codes can be seen in Figure 35. Another form of controlling the generation process is the addition of polyp classes as an additional embedding feature to the input of the GAN. Should a dataset be fabricated in which each label is associated with a polyp type, an embedding similar to the one developed for bounding boxes could be done to assert the type of polyp to be generated.

6.3 Further Extensions

This study focused on answering two questions:

- Is it possible to synthesize realistic images of endoscopy images containing polyps?
- Would augmenting existing endoscopy training sets with synthetic endoscopy images increase the detection performance of a detection model?

However interesting these questions are, the following ideas are possible extensions that were too ambitious to be covered in the context of a single master thesis that could be built on top of existing research as separate publications.

1. Investigating the plausibility of augmenting polyp datasets with selected classes of polyps. As previously mentioned, there is a varying degree of presence of polyps pertaining to different classes which is partially caused by the existence of different polyps in the real world. Increasing the number of polyps of low-frequency polyp classes in training sets could ideally allow for not only an increase in detection performance but an increase in classification performance as well. As such, further research could look into the feasibility of data synthesis but from a classification standpoint. As medical practitioners have stated, the trickier to spot polyps would be more valuable to automatically spot than the obvious protruding ones as they are more likely to be omitted during the endoscopy procedure. The means towards achieving this could lie in embedding class information in conditional versions of GANs, manipulating the latent space in a manner that would shift the style of the polyp from one type to the next.

2. Integrating a wider range of images into the GAN should aid the network in capturing a wider distribution that is more representative of polyps. Although the images generated by StyleGAN2-Ada could be deemed ”realistic” from a subjective perspective, they are homogeneous since they originate from a latent space folded over a dataset that is also homogeneous. Integrating images from different datasets that contain different lighting conditions, different angles of the same polyp type, and generally as much diversity as possible would increase the robustness of the learning process to fit a distribution akin to that of a real-world polyp.

3. Analyzing the performance of the automated labeling process created via conditional versions of GANs by comparing the automated labels versus manual labels. The results of the automated labeling process have shown promising bounding boxes drawn over more than half the surface area of the polyp. However, there is much improvement to be covered concerning its accuracy. If the automated labeling process is improved, the ground truth that the detection model uses to extract the polyp would include a similar structure as one marked manually by a medical expert. Minimizing the discrepancy between the two labels originating from the two mediums will positively impact the performance of the augmentation technique.

4. AutoML: The experiments suggest an inverse correlation between FID and mAP. This indicates that optimizing the generative process to minimize FID could constitute a valid automated way of improving the models. Training on better hardware could enable more experimentation in the variety of models trained as well as allowing for automated pipelines of model selection and hyperparameter optimization. For running generative methods, NVIDIA recommends the usage of 8 V100 GPUs in a DGX-1 configuration, a rig that, at the time of writing, costs \$149000. If such a rig would be available, faster feedback loops to both ML practitioners and Auto ML pipelines would enable the creation of better models in a shorter amount of time.

5. One of the biggest factors that influenced the realism and quality of the images were artifacts that appeared in a portion of the generated images. The networks were, most of the time, apt enough to generate images containing polyps, but on some occasions, the images contained a wide range of artifacts. Some pertained to the image quality being corrupted by noise-like spots and shearing. Others pertained to the label generated by CGAN being shifted, causing the bounding box to cover less than 50% of the surface area of the polyp. Removing the artifacts mentioned here could drastically increase the performance of the model, something that we have seen by manually pruning images containing artifacts.

6. Labels being off of the actual polyp was detrimental in degrading the performance of the generator. Although the FID has merits in its ability to quantify realism, it might not be as important as some measure that would quantify how accurate the generated labels are. This would imply that we would have some form of measuring the accuracy of a bounding box within a generated polyp. However, optimizing for such a metric would yield more accurate bounding boxes which, as we have shown, are critical to the success of the detector.

7 Conclusion

After building three distinct generative adversarial network architectures, an automated label creation method, and testing a wide array of augmentation configurations and latent space manipulations it is time to answer our research questions.

7.1 Image Realism

StyleGAN2-Ada obtained the lowest FID score of all three architectures, making it the model that should output the most similar images to the original dataset. After an inspection of the generated images, several medical practitioners commented that the images produced by StyleGAN2-Ada appeared to include realistic traits of protruding polyps. Due to the very nature of research conducted in this paper, tackling a subjective matter such as realism is difficult and concluding whether the images produced by our best model are realistic is near-impossible. Nevertheless, with the degree of realism obtained we are happy to concolude on a positive note on the first research question in that the most promising model is capable of producing images with a degree of realism that can trick medical experts. On top of that, the fact that we are capable of producing realistic images, wherever we want, and make it look however we want, we see these achievements as proud steps in the right direction.

7.2 Detection System Performance

The best performing model obtained a score equal to that of one that did not train on an augmented dataset. Though disheartening, we have outlined the possible causes and solutions to this phenomenon and state that even though we have obtained a negative result to our second research question, the realm of possibilities for improving the developed models is both extensive and exciting!

8 Bibliography

[1] A. J. Tresca, “An overview of adenomatous polyps,” Jun 2021.

[2] M. Yamada, Y. Saito, H. Imaoka, M. Saiko, S. Yamada, H. Kondo, H. Takamaru, T. Sakamoto, J. Sese, A. Kuchiba, T. Shibata, and R. Hamamoto, “Development of a real-time endoscopic image diagnosis support system using deep learning technology in colonoscopy,” *Scientific Reports*, vol. 9, 09 2019.

[3] I. Barua, D. G. Vinsard, H. C. Jodal, M. Løberg, M. Kalager, Ø. Holme, M. Misawa, M. Bretthauer, and Y. Mori, “Artificial intelligence for polyp detection during colonoscopy: a systematic review and meta-analysis,” *Endoscopy*, vol. 53, pp. 277–284, Mar. 2021.

[4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, pp. 115–118, Feb. 2017.

[5] W. W and G. DG, “Mistakes in the management of gastric polyps and how to avoid them,” Apr 2021.

[6] A. Nogueira-Rodríguez, R. Domínguez-Carbajales, H. López-Fernández, Águeda Iglesias, J. Cubiella, F. Fdez-Riverola, M. Reboiro-Jato, and D. Glez-Peña, “Deep neural networks approaches for detecting and classifying colorectal polyps,” *Neurocomputing*, vol. 423, pp. 721–734, 2021.

[7] G. Elidan, N. Lotner, N. Friedman, and D. Koller, “Discovering hidden variables: A structure-based approach,” 08 2001.

[8] A. Bochkovskiy, C.-Y. Wang, and H.-y. Liao, “Yolov4: Optimal speed and accuracy of object detection,” 04 2020.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[10] C. Ongun and A. Temizel, *Paired 3D Model Generation with Conditional Generative Adversarial Networks*, pp. 473–487. 01 2019.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Advances in Neural Information Processing Systems*, vol. 3, 06 2014.

[12] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” 12 2017.

[13] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of StyleGAN,” Dec. 2019.

[14] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger, “An empirical study on evaluation metrics of generative adversarial networks,” 06 2018.

[15] H. Wu, S. Zheng, J. Zhang, and K. Huang, “Gp-gan: Towards realistic high-resolution image blending,” 03 2017.

[16] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 11 2014.

[17] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” 10 2016.

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” Oct. 2017.

[19] I. Durugkar, I. Gemp, and S. Mahadevan, “Generative multi-adversarial networks,” 11 2016.

[20] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” 06 2016.

[21] R. Zhang, “Making convolutional networks shift-invariant again,” 04 2019.

[22] A. Karnewar and R. Iyengar, “Msg-gan: Multi-scale gradients gan for more stable and synchronized multi-scale image synthesis,” 03 2019.

[23] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” June 2020.

[24] M. Herlihy and N. Shavit, *The Art of Multiprocessor Programming, Revised Reprint.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1st ed., 2012.

[25] J. Bernal, N. Tajbakhsh, F. Sanchez, B. Matuszewski, H. Chen, L. Yu, Q. Angermann, O. Romain, B. Rustad, I. Balasingham, K. Pogorelov, S. Choi, Q. Debard, L. Maier-Hein, S. Speidel, D. Stoyanov, P. Brandao, H. Córdova, C. Sánchez-Montes, and A. Histace, “Comparative validation of polyp detection methods in video colonoscopy: Results from the miccai 2015 endoscopic vision challenge,” *IEEE Transactions on Medical Imaging*, vol. PP, pp. 1–1, 02 2017.

Appendices

A Literature

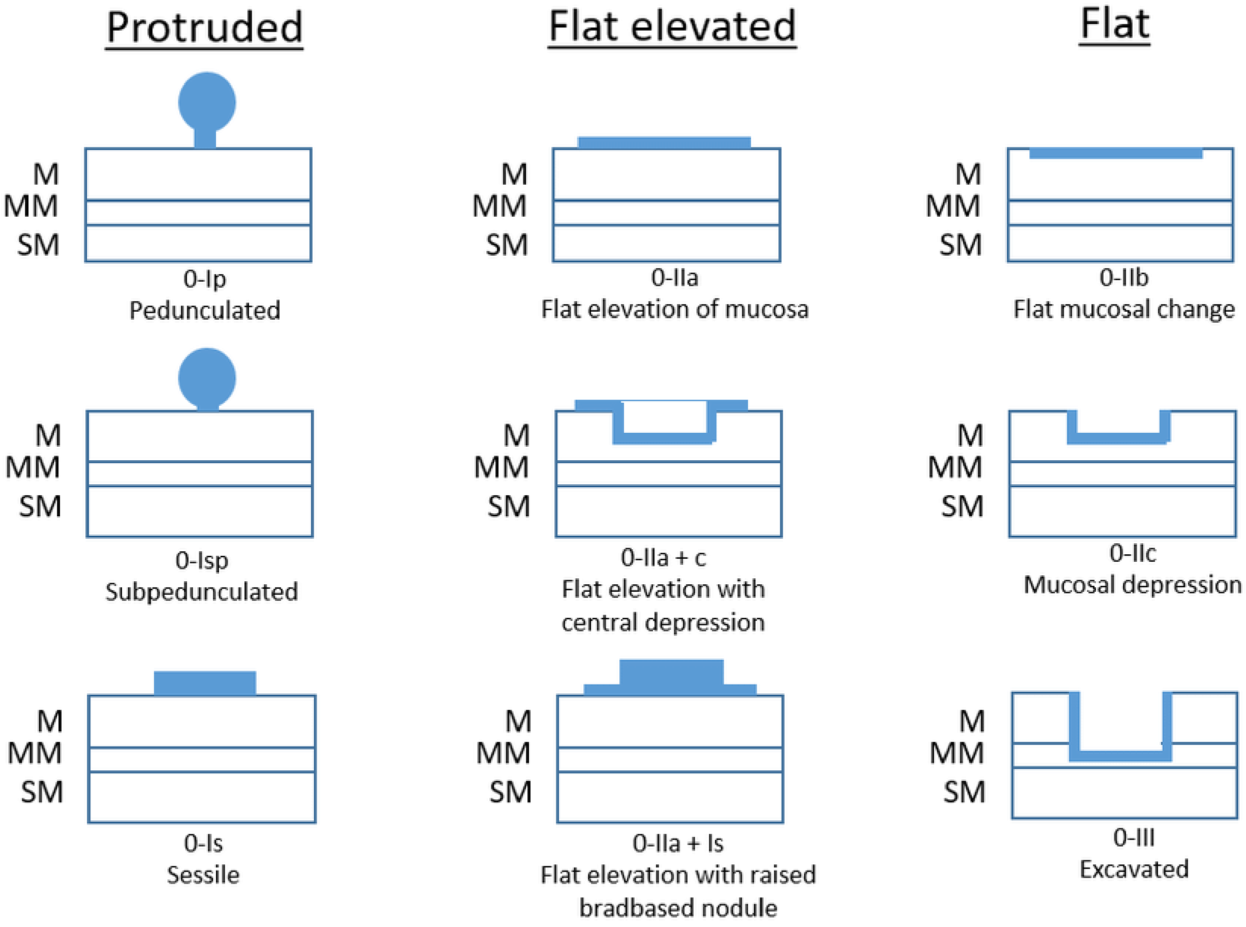


Figure 18: The different shapes a polyp may have. Flat and recessed polyps present a particularly more difficult challenge to detect because they blend easier with the walls of the tract. Image taken from [25].

B Generated images

B.1 Baseline Candidate Model

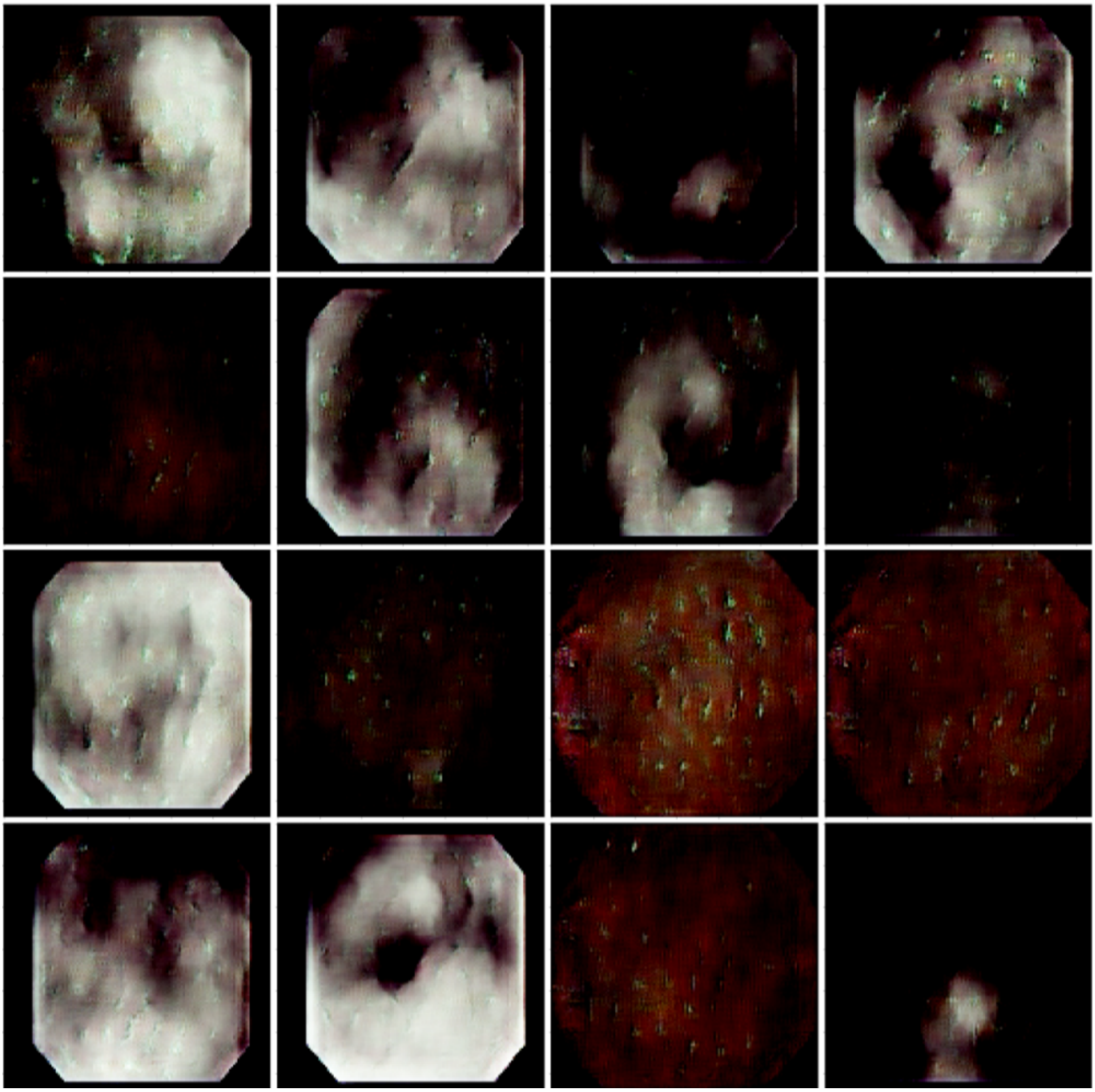


Figure 19: **ID: 1** Baseline model showcasing mode collapse after training on 1200 images with no image augmentation.

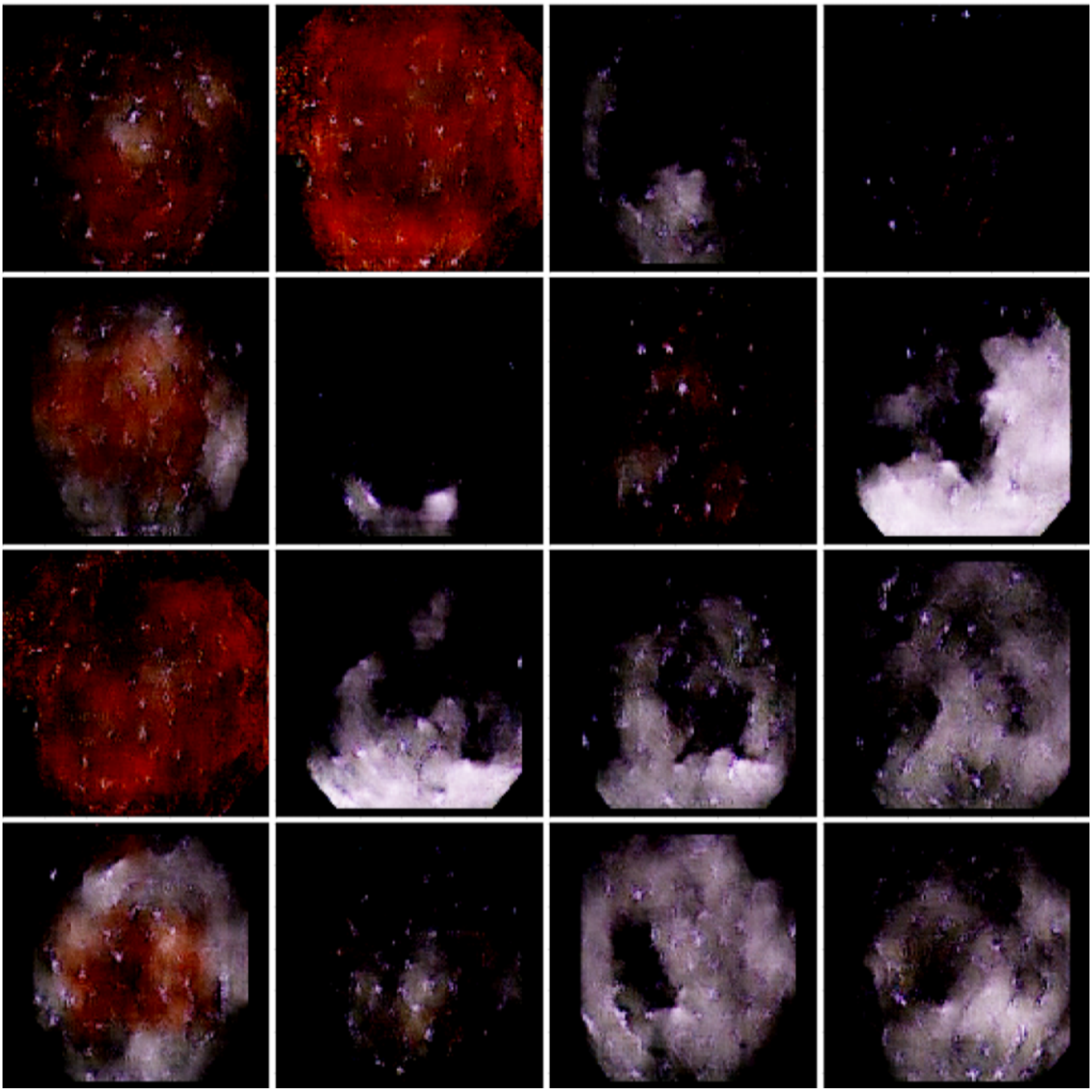


Figure 20: **ID: 2** Since the previous network converged to mode-collapse, one potential solution would be to increase the number of dimensions of the latent space. Doubling the number of latent dimensions for the baseline model to increase the representative power of the network did not yield significantly different results. However, a step in the right direction can be observed since the images seem to have more color in them.

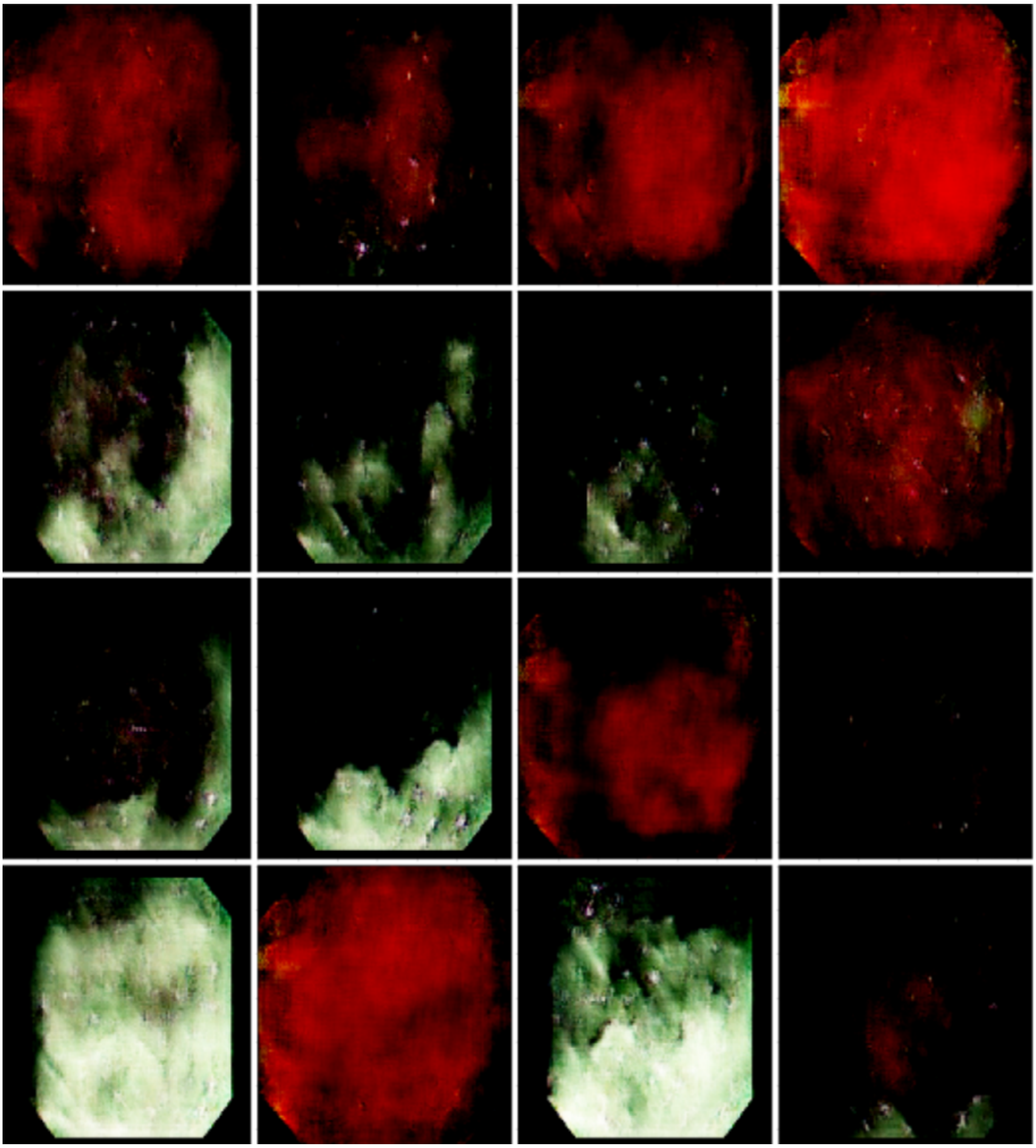


Figure 21: **ID: 3** With 512 latent dimensions the network captures the concept of color more profoundly. Nevertheless, changing only the number of latent dimensions for such a simple architecture seemed to be insufficient at this point.

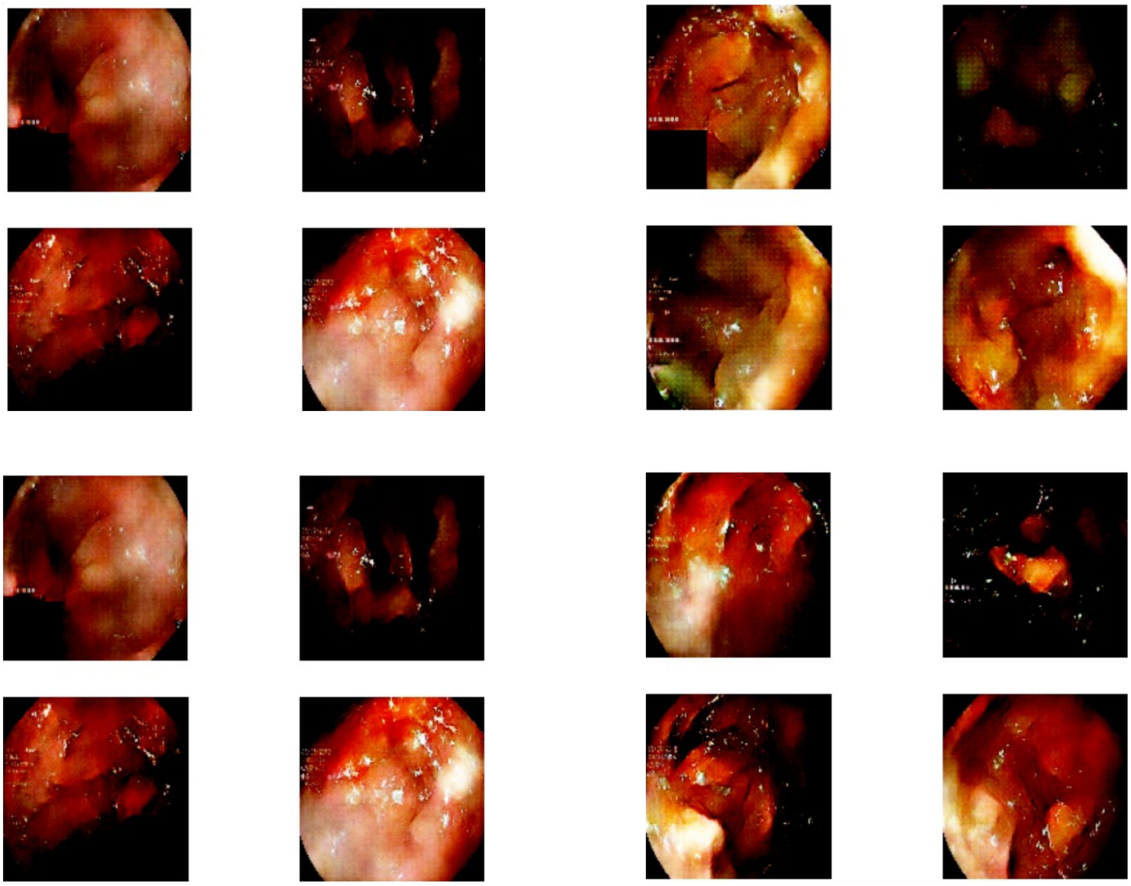


Figure 22: **ID: 4** Baseline model employing affine image augmentations. The simple architecture is able to capture certain features akin to that of images taken during an endoscopy. This was the first major breakthrough that showed that the first research question posed in this paper might have a positive outcome.

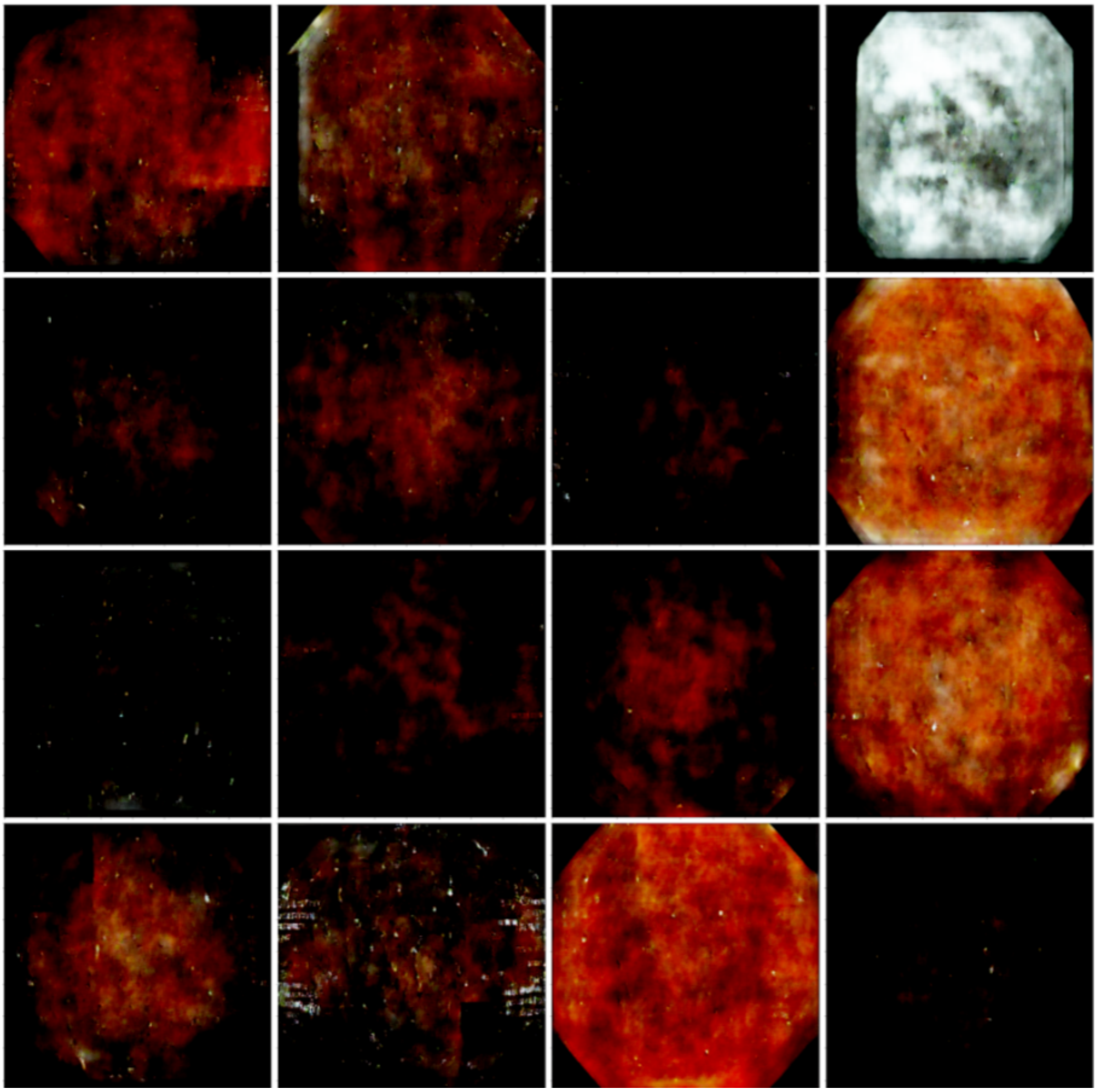


Figure 23: **ID: 5** Baseline model outputting 416x416 images which corresponds to the desired, minimal resolution accepted by YOLO. At higher resolutions, the network's performance does not carry over as adjustments in the number of filters and their sizes have to be made to accommodate for the change.

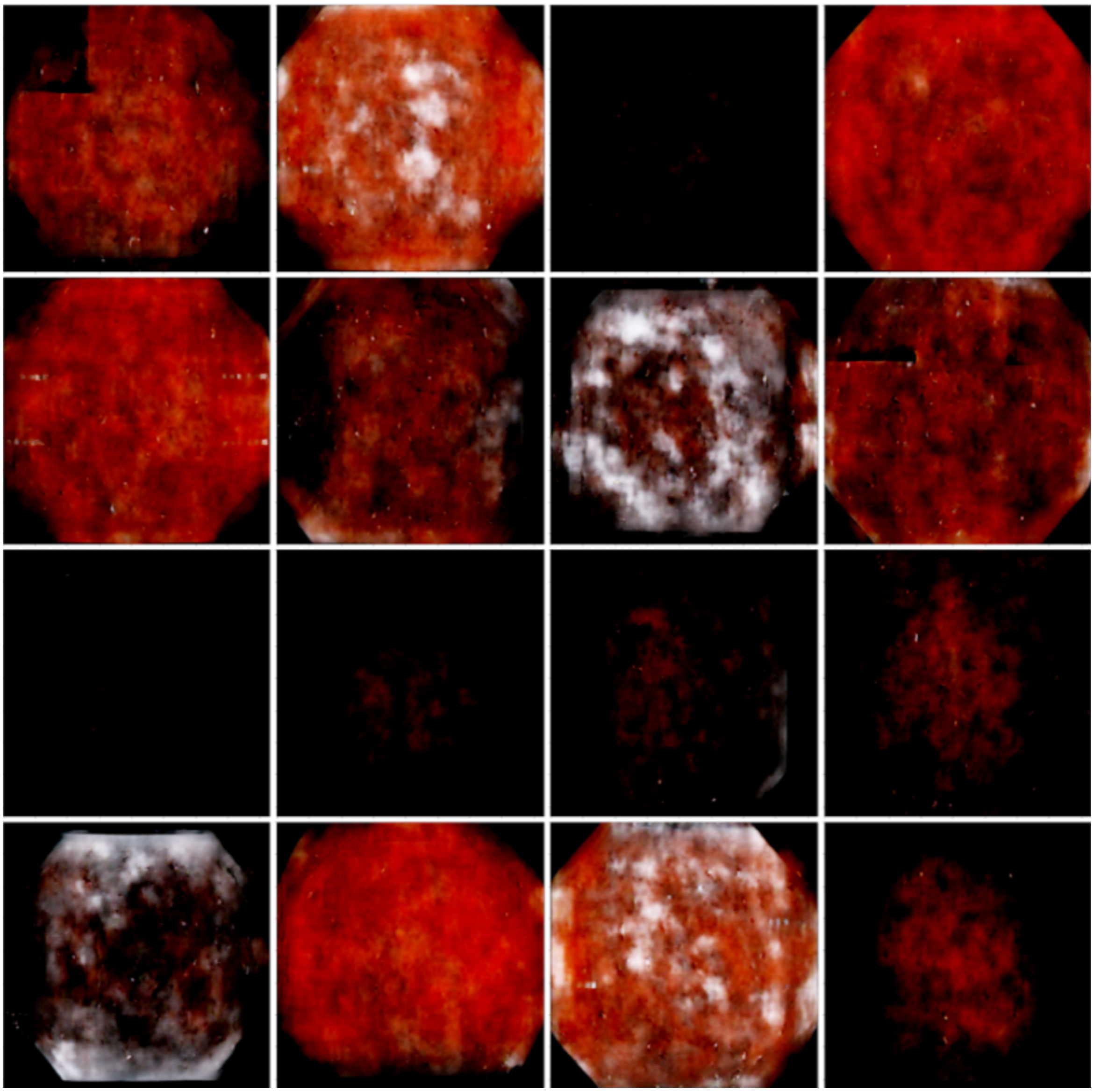


Figure 24: **ID: 6** Baseline model with increased kernel sizes from 3x3 to 5x5 outputting 416x416 images. An increase in kernel size did not correspond with an increase in image quality.

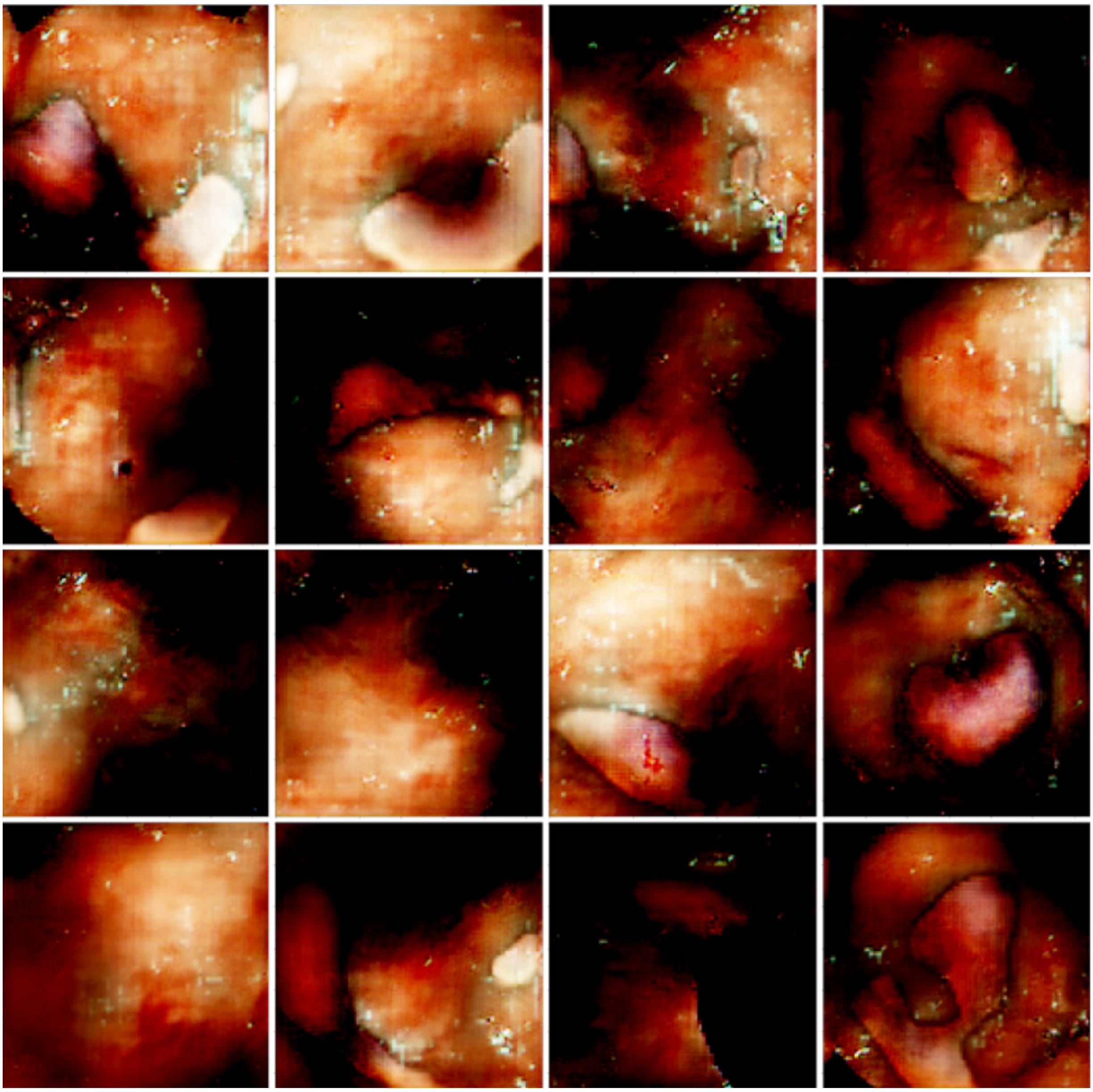


Figure 25: **ID:7** Baseline model with increased latent space dimensionality from 128 to 512. With a larger number of dimensions, the generator captures more diverse features from the images compared to previous models.

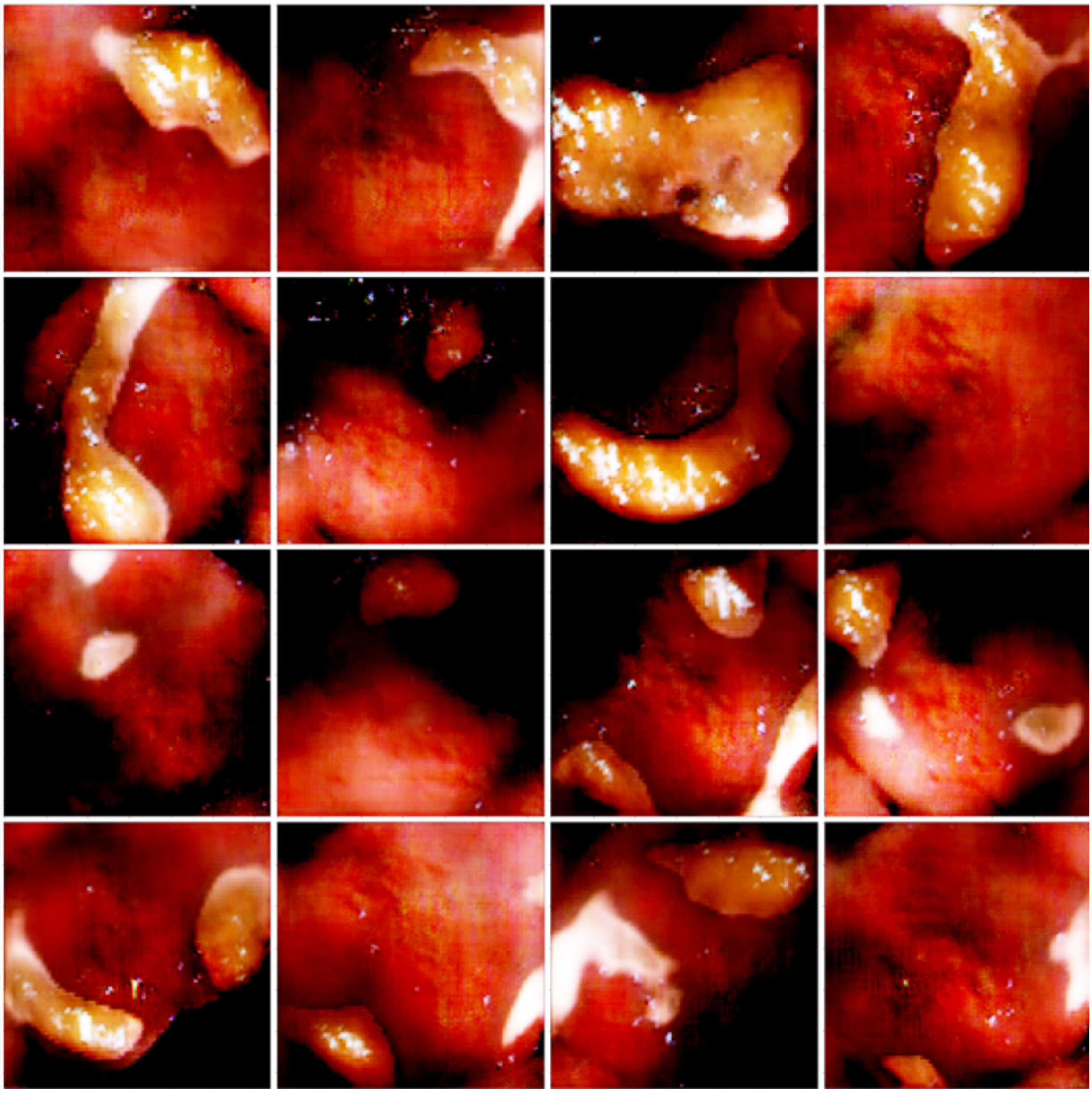


Figure 26: **ID:8** Baseline model with increased latent space dimensionality and increased k-size. Results are similar to **ID:7** in image quality and **ID:6** in effect: an increase in kernel sizes from 3 to 5 does not influence the performance of the generator.

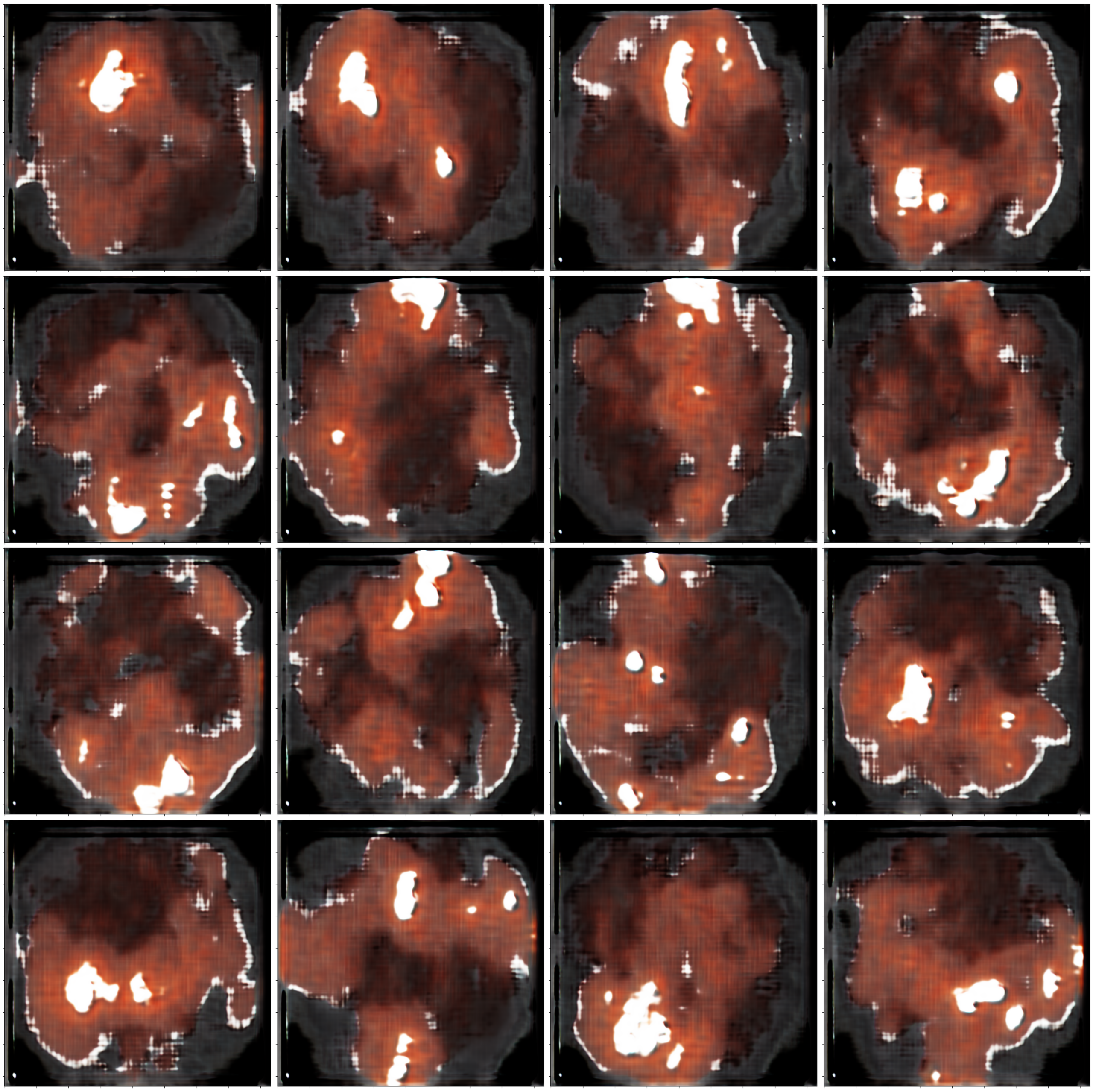


Figure 27: **ID:9** Baseline model using same config as **ID: 8** but trained for 2000 epochs as opposed to only 1000. The model seemed to have mode-collapsed into a less optimal minima compared to **ID: 8**.

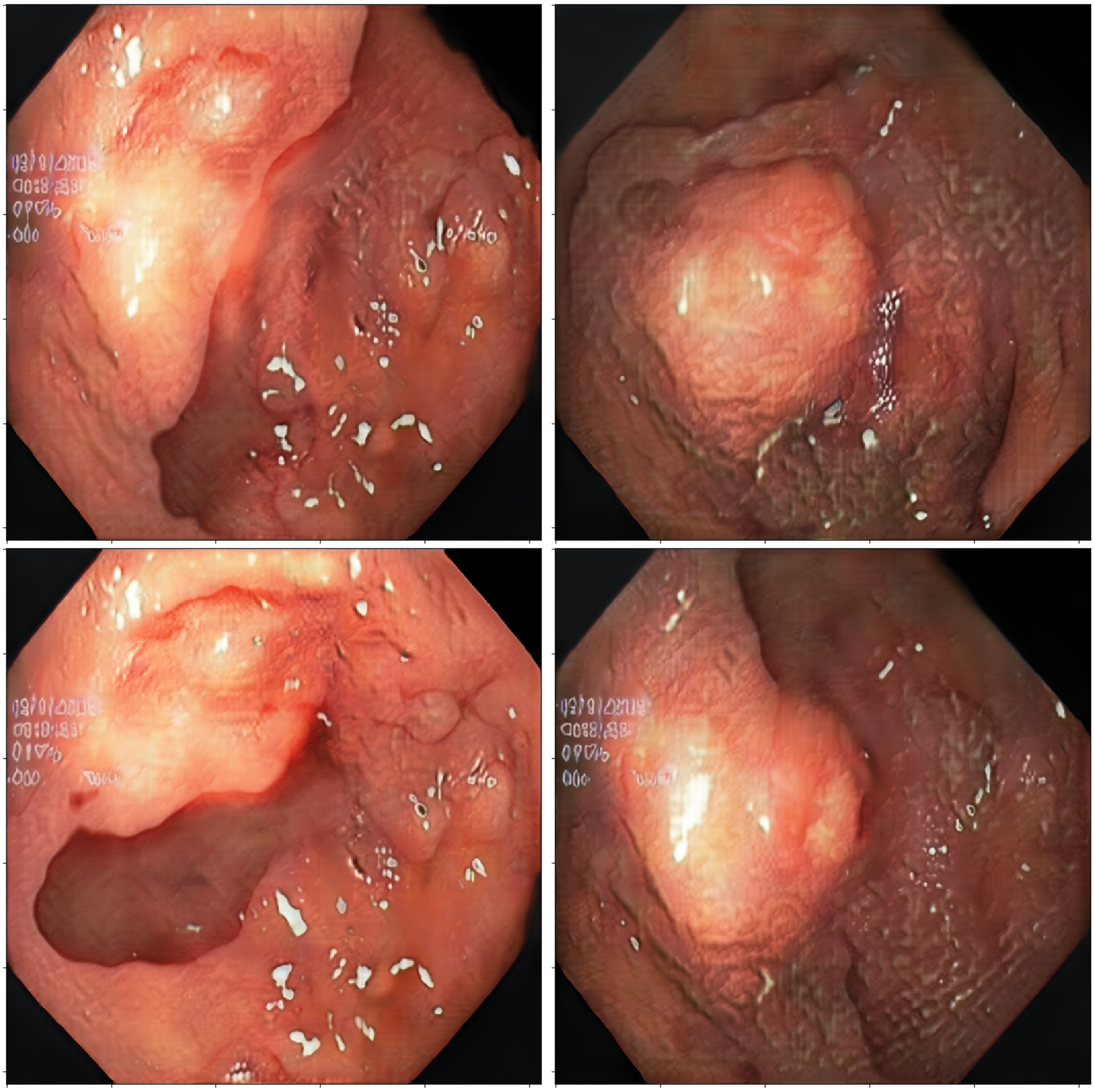


Figure 28: Baseline candidate model obtained after various experiments. Configuration of network can be referenced in Table 8.

B.2 PGGAN Candidate Model

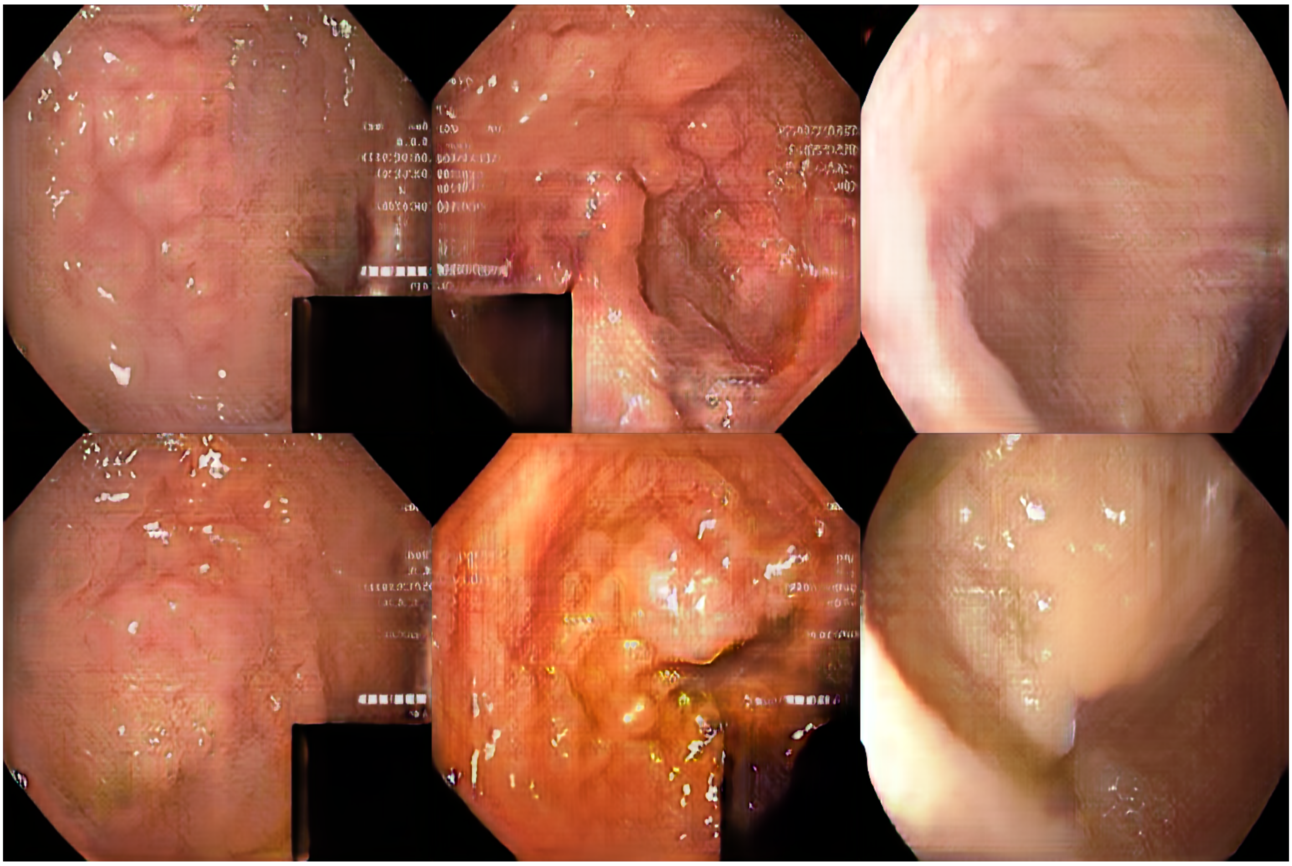


Figure 29: Generated images by PGGAN after 9220 epochs of training.

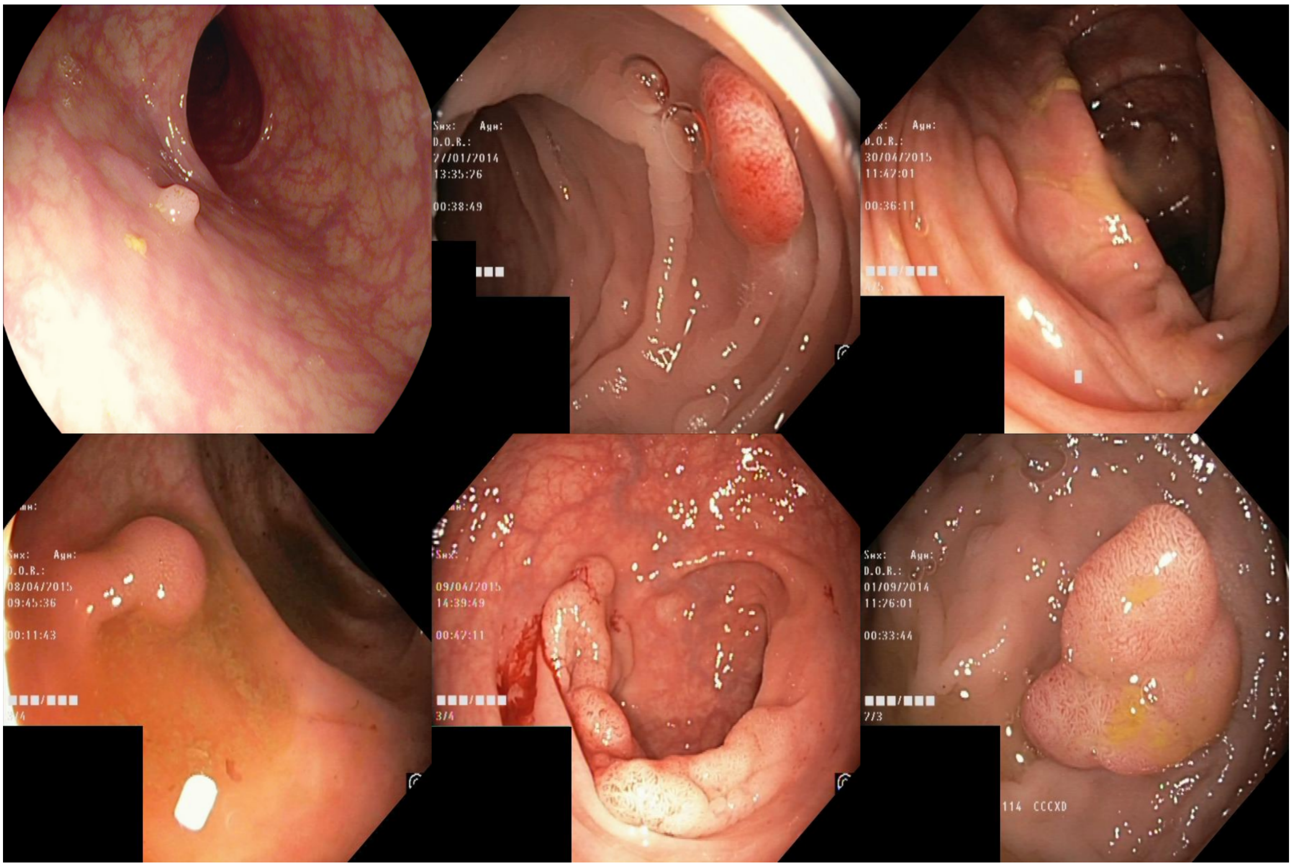


Figure 30: Real images PGGAN is trying to imitate.

B.3 StyleGAN2-Ada Candidate Model

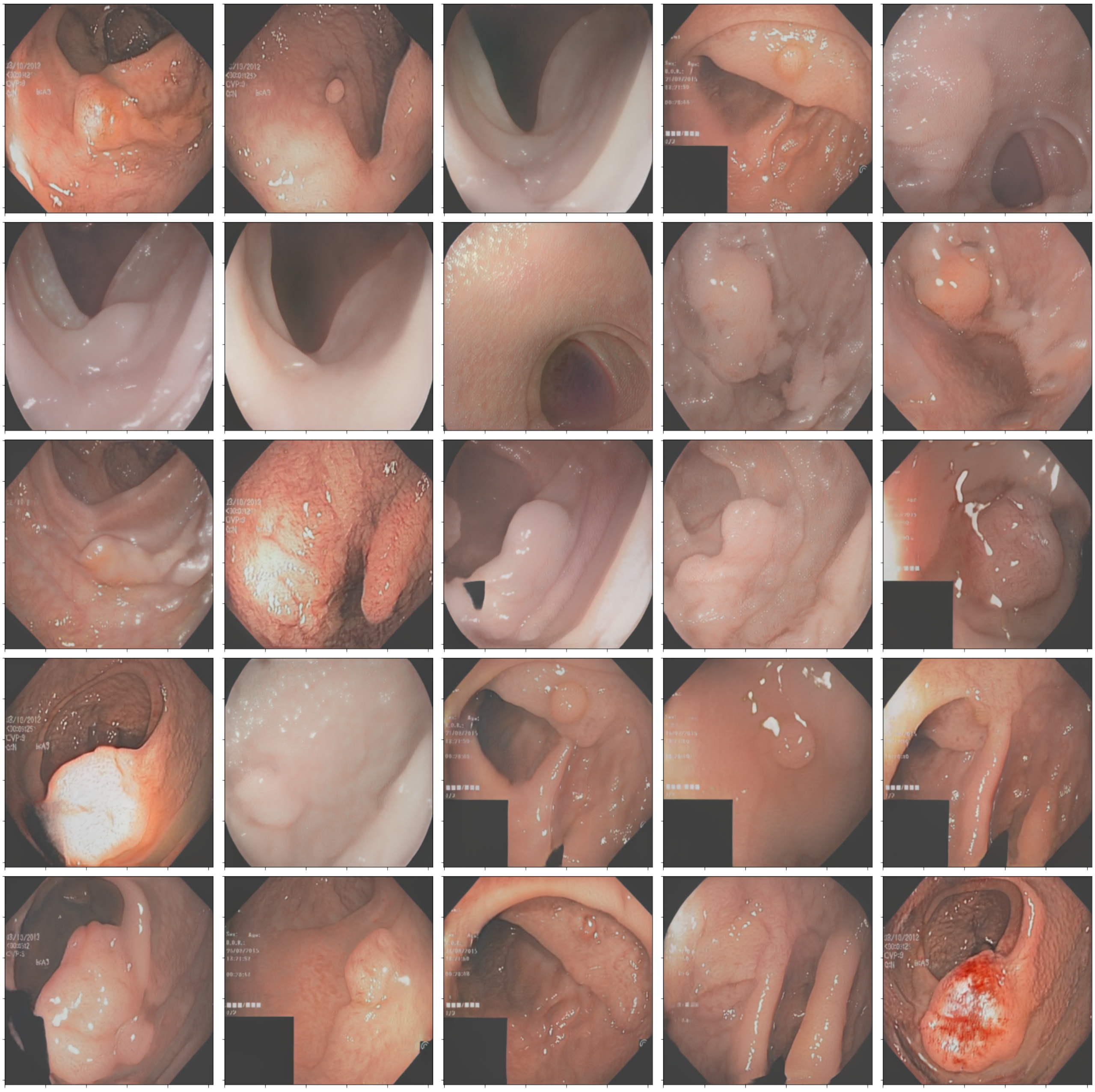


Figure 31: Example of images generated by StyleGAN2-Ada after 2400 epochs.

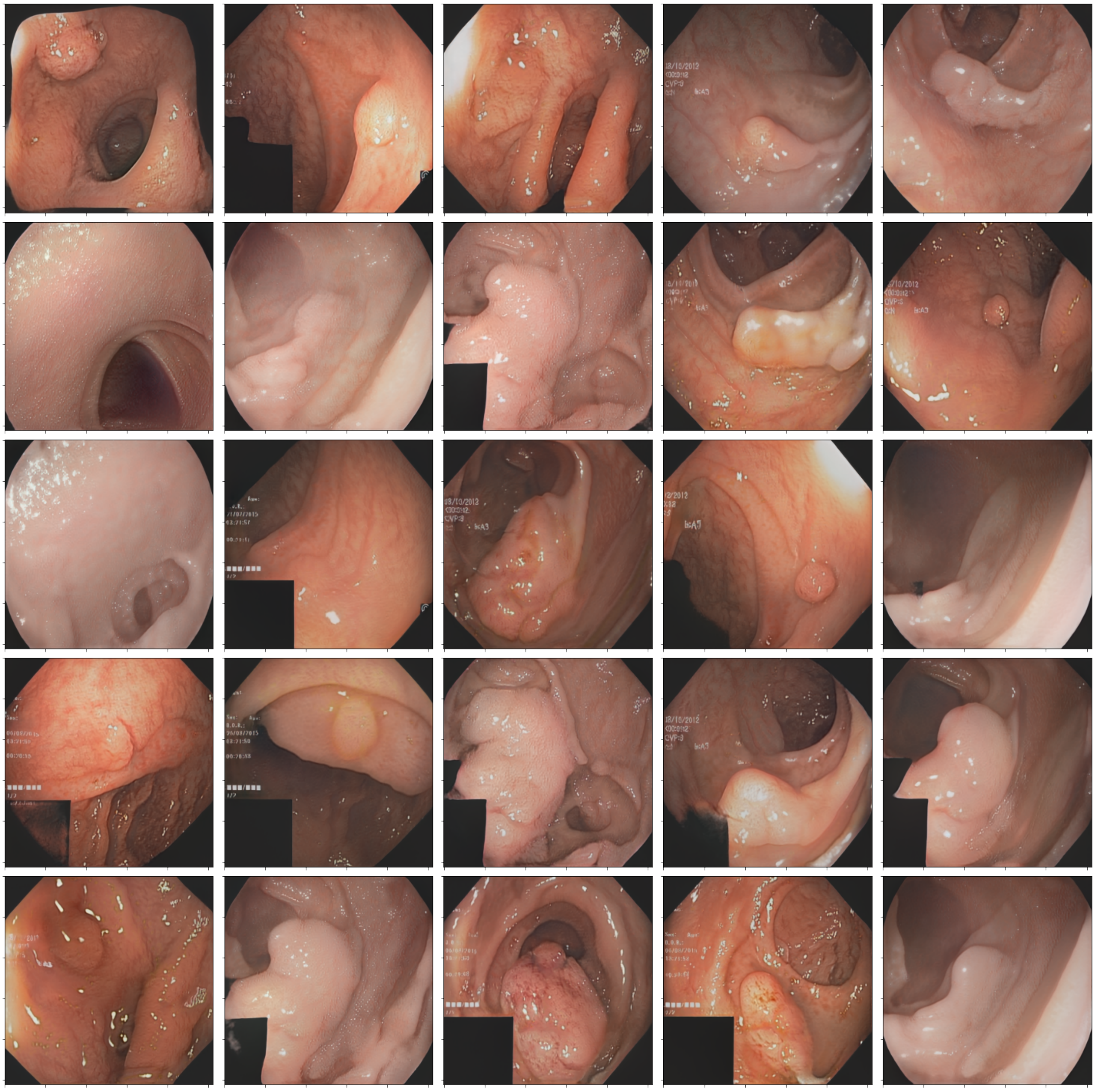


Figure 32: Example of images generated by StyleGAN2-Ada after 2400 epochs.

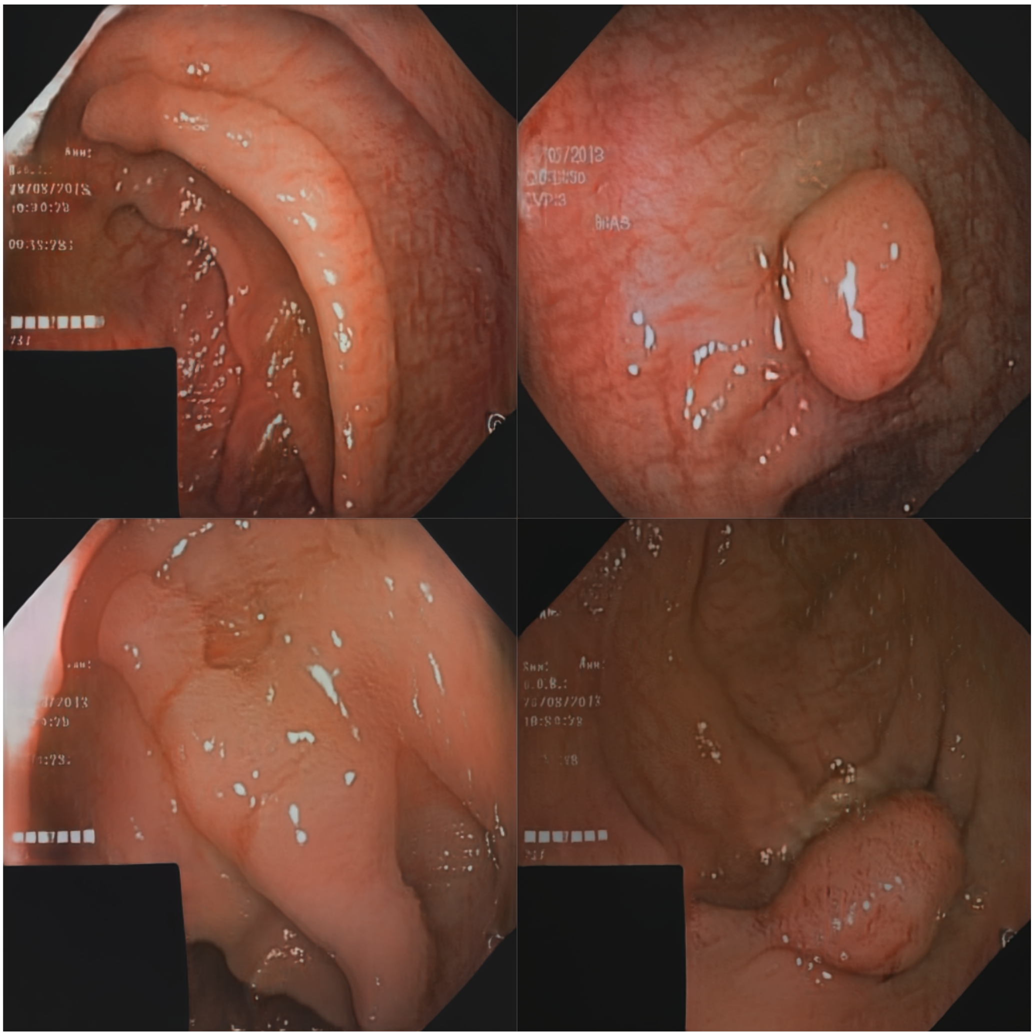


Figure 33: Selection of images generated by StyleGAN2-Ada after 2400 epochs.

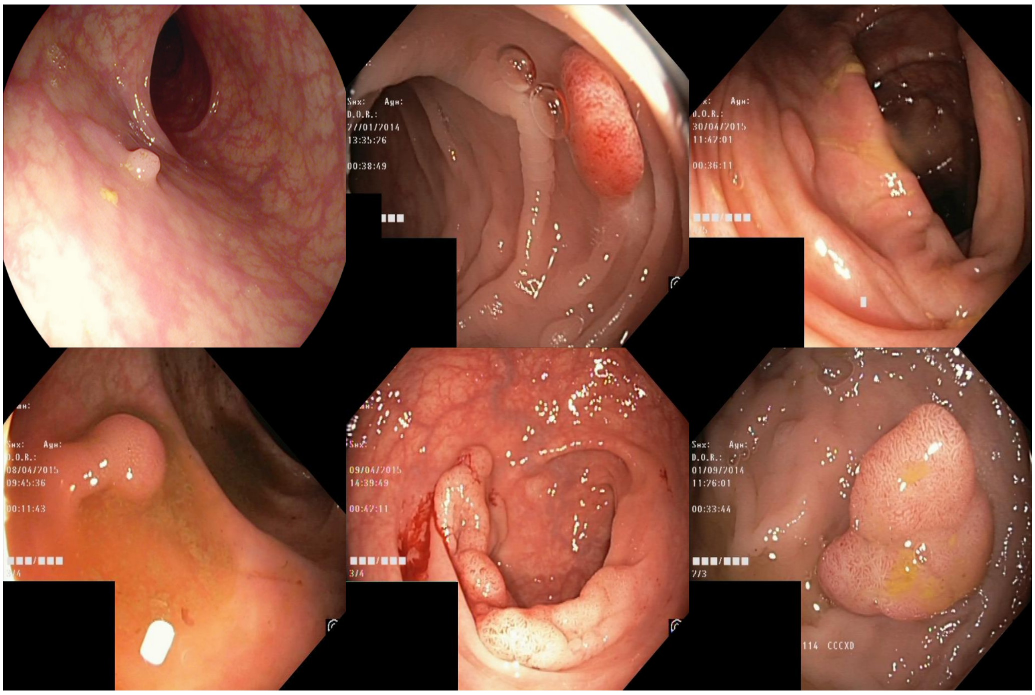


Figure 34: Selection of real images StyleGAN2-Ada is trying to imitate.

B.4 Latent space interpolation

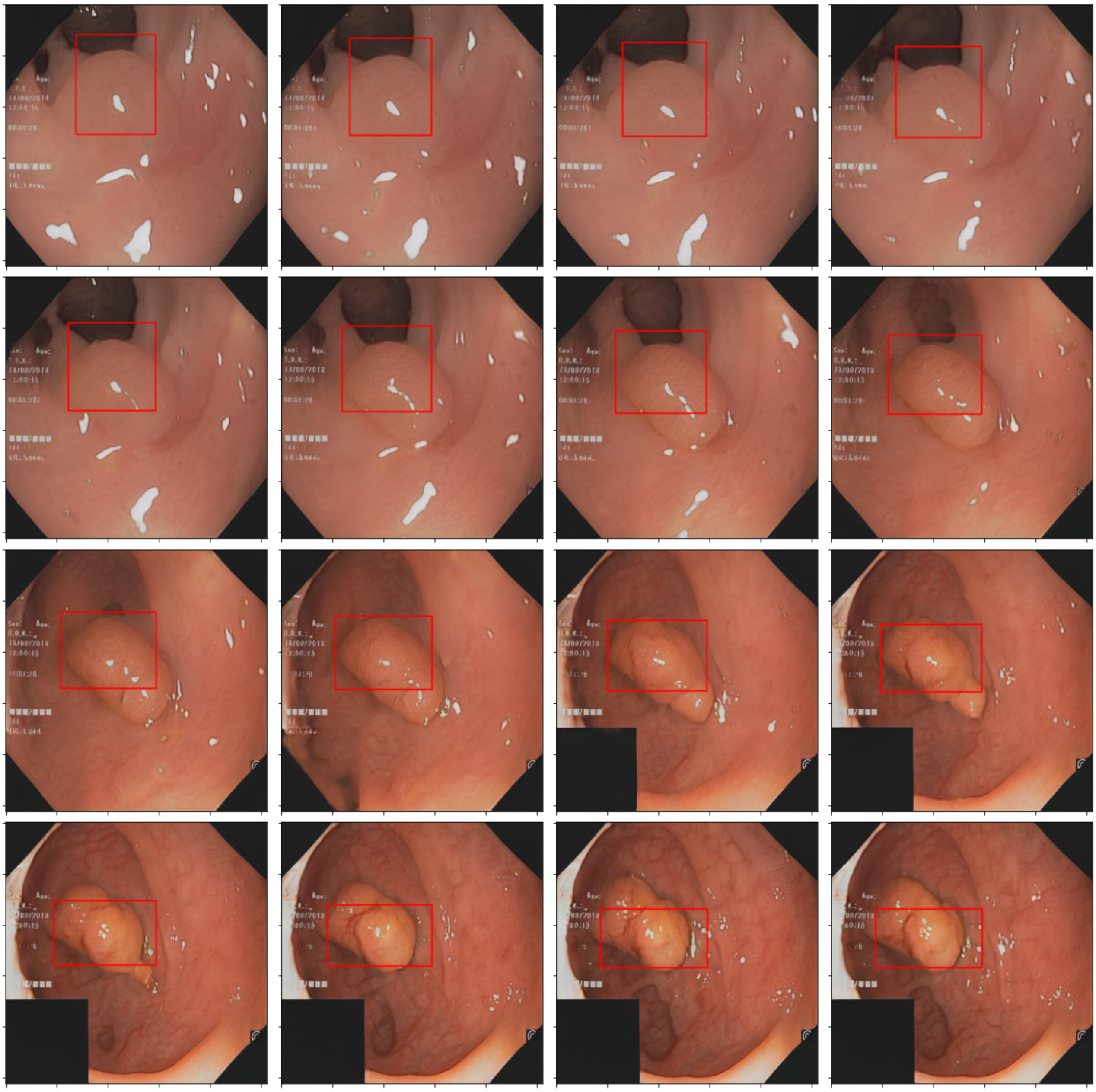


Figure 35: Linear interpolation between two samples drawn from the latent space z. Top left (0, 0) and bottom right (3, 3) images are next to each other in the latent space. In-between them are linear interpolations where the closer an image is to (0, 0) or (3, 3) the more it is influenced by the features of said images. Images generated using conditional StyleGAN2-Ada.

C ZiuZ detector on fakes

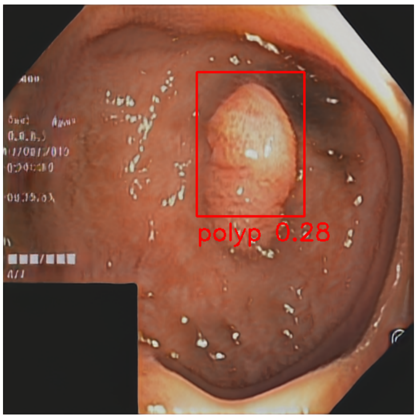


Figure 36: Bounding box drawn by ZiuZ' YOLOv4 model on an image generated by StyleGAN2-Ada after training for 2400 epochs. 0.28 was the lowest confidence obtained above 0.22 which is considered to be the discriminator value to determine the existence of a polyp in an image. The drawn bounding box nearly covers the entirety of the polyp.

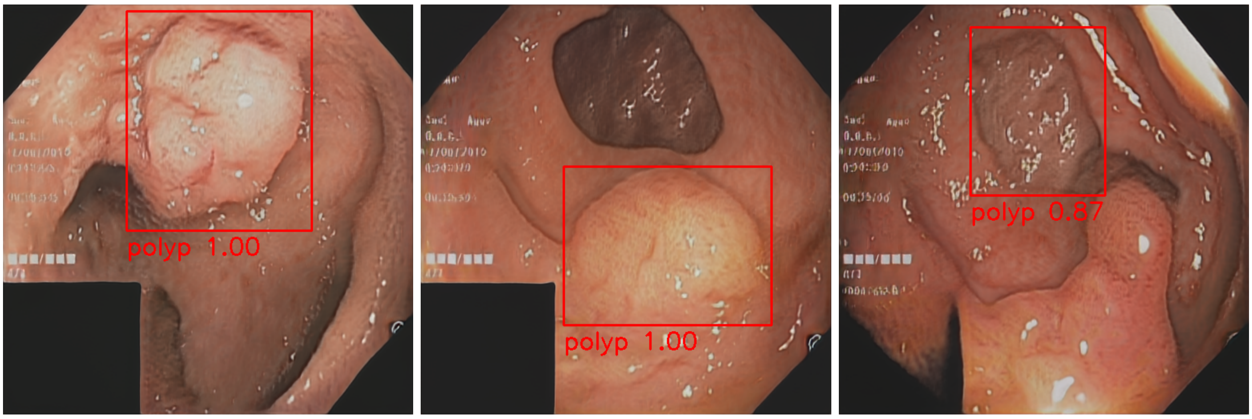


Figure 37: High confidence scores obtained by ZiuZ' YOLOv4 model on generated images by StyleGAN2-Ada after training for 2400 epochs. Although the rightmost image has a 0.87 confidence score, drawn bounding box does not contain a polyp.