Surgical Image Generation using GANs and Adaptive Style Transfer

Bachelor Thesis

Julius Schmitt

Division Translational Surgical Oncology
NCT Dresden

Reviewer:	Prof. Dr.-Ing. Stefanie Speidel
Second reviewer:	Dr. Micha Pfeiffer
Advisor:	Danush Kumar Venkatesh

Duration: December 13, 2023 – February 28, 2024

NCT
NATIONAL CENTER
FOR TUMOR DISEASES
PARTNER SITE DRESDEN
UNIVERSITY CANCER CENTER UCC

TECHNISCHE
UNIVERSITÄT
DRESDEN

I hereby declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Dresden, 27 February, 2024

Abstract

This thesis explores possibilities to increase the accuracy of Deep Learning models in the domain of cholecystectomy, by using artificially created training samples. The possibilities of a GAN architecture in combination with arbitrary style adaptation are explored. It will be found, that the contrastive unpaired image translation decreases performance on a segmentation task compared to a non-artificial dataset. This decrease will be less if an Adaptive Instance Normalization or Adaptive Attention Normalization architecture is used, to guide the translation with auxiliary information about the visual style of the target domain during training.

Contents

1. Introduction

1.1 Motivation

Deep Learning models have seen immense growth in popularity and accuracy in various fields in recent years. With that trend comes the possibility of deep vision models being used in various applications in the surgical domain. To date, training such models often requires huge amounts of example data. However, the number of example images that may be obtained, is heavily limited by the complexity of the domain. Especially annotating such image examples is a laborious task that can only be done by experts. In the past various methods [1, 2, 3, 4] had some success with artificially creating such samples, which may then be used in combination with real samples to train segmentation models with a focus on laparoscopic surgical interventions. The models will be given some sort of descriptive image of what can be seen by the camera of a laparoscope. The descriptive image might for example be a computer-simulated segmentation mask or 3D rendering. The generation problem can be reduced to an unpaired translation problem. In this class of problems, two semantically similar domains A and B of images are present but no direct link between any two pairs a and b. There have been various approaches for this problem class in the past including the usage of dedicated models [5, 6, 7, 8]. One of the most recent is the contrastive unpaired translation (CUT) [8]. It shows better computational efficiency than its predecessor CycleGAN and had slightly improved FID scores in the authors' experiments.

Recent tests with this model, qualitatively implicated that the target domain was not learned well enough. Lots of details were generated that made little sense in their spatial context, which would lead to bad accuracy in the downstream task. Using other models, this problem would also appear and would for example be addressed by adding a Multi-scale Structural Similarity [9] loss between descriptive image and generated image [1, 2]. One reason might be the network facing a shortage of training examples and therefore training leading to bad representations of the target domain. To provide auxiliary information to the model, access to an example of the target image shall be given, granting the possibility to pull features from the target images during generation instead of completely having to rely on patterns, the model insufficiently learned during training.

1.2 Goals

The goal of this thesis is to improve generated image quality with regard to using them as training examples for a segmentation model. These steps are taken towards this goal.

- Modify the CUT architecture[8] to include an arbitrary style transfer step mechanism, with AdaIN[10] and AdaAttn[11].
- Train a baseline model based on the CUT architecture and two models based on the modified architectures and generate test image sets. A dataset[1] containing synthetic and real images of cholecystomic procedures is used.
- Evaluate the FID[12] scores of the test images in comparison to the dataset.
- Evaluate label preservation capabilities of the test images using a segmentation model.
- Evaluate the test performance of a segmentation model that was pretrained with the test images.

2. State of the Art

2.1 Image translation and style transfer

Zhu et al., introduced the CycleGan [6]. In conjunction with GANs [13], the architecture employs cycle-consistent adversarial networks to learn mappings between domains without requiring paired training data.

Gatys et al. [14]. proposed an approach to image style transfer using convolutional neural networks. It utilizes feature representations at different layers of a pre-trained CNN to separate and manipulate content and style in images.

AdaIN, proposed by Huang et. al. [10], enhances the flexibility of image style transfer by dynamically adjusting the mean and standard deviation of features. This adaptive normalization technique allows for arbitrary style transfer, enabling the generation of diverse results.

MUNIT, presented by Huang et al. [5], use representations of content and style, enabling the model to generate diverse outputs for a single input. It uses a GAN loss and cycle consistenly similar to the CycleGan but with disentangled latent representations for style and content information.

The concept of Adaptive Attention Normalization has been introduced by Liu et. al. [11] as a refinement to normalization techniques like AdaIn. By modulating normalization parameters based on attention maps, it has demonstrated improved performance in capturing visual patterns.

2.2 Image Generation for surgical imaging

Image synthesis has been proposed as a means to generate medical images using simulated and real domains for other downstream Machine Learning tasks [15]. Among others, the approach is successfully used in cholecystectomy-related laparoscopic image generation. The following methods approached the generation problem.

Pfeiffer et al. [1] simplified the MUNIT architecture [5] by assuming, that only one style is present in the origin domain. They also identified a problem with the networks inventing additional details. This was countered by adding an MS-SSIM [9] loss. They improved the segmentation results of a model by pretraining it with their generated images.

Dowrick et. al. [4] used a 3D Model of the liver to create renderings in unity and used them for training a segmentation model.

Kaleta et. al. [3] could generate highly realistic-looking images. They used a Stable Diffusion (SD)[16] model, which can be conditioned on text or images. The model was adapted to a new image domain using real images and text with the DreamBooth [17] technique. The fine-tuned model generated realistic images from synthetic data and text prompts, aided by ControlNet [18] architectures for additional control. The method significantly improved baseline results, achieving a high mean Intersection over Union.

Rivoir et. al. [19] introduce an approach for the generation problem with videos from simulated surgical scenarios to realistic images. The method learns neural textures [20] representing global scene texture, and an unpaired image translation module based on MUNIT is used in conjunction with synthetic images for generation. A lighting-invariant view-consistency loss ensures consistency across different views and frames.

Venkatesh et. al. [2] added an MS-SSIM loss to the CUT model and named the resulting method ConStructS. They used two methods to assess the quality of the model's results. In the first method, synthetic masks were attempted to be recovered after translation. A segmentation model was trained in the second method, and its results were tested on a test split. ConStructS, along with various other models, including the baseline CUT and CycleGAN [6, 8], were then evaluated using this approach.

Similar to Pfeiffer et. al., in this thesis, a style transfer and a GAN mechanism shall be utilized, but using the new CUT architecture which does not use cycle consistency, has only one translation direction, and does not employ style transfer by default. This is an addition to the architecture, which may be used with other modifications like the one from Venkatesh et. al.

3. Theoretical Background

3.1 Minimally Invasive Surgery and Laparoscopy

Minimally invasive surgery (MIS) is a surgical practice, where only small incisions are made to minimize body trauma in comparison to traditional surgery. A special subcategory of MIS is laparoscopy, where a thin, flexible tube with a camera (laparoscope) is inserted through small incisions in the body to guide the surgeon. The images can be digitally recorded and transferred, setting the basis for digital analysis. Cholecystectomy is a laparoscopy procedure, where the gallbladder of a patient is removed.

3.2 Cross domain image generation to increase segmentation training

Using real-world data in machine learning applications is often very challenging, as datasets may be of bad quality or contain too few data points as capturing data may be restricted privacy, safety, and regulational concerns [21]. This is especially the case in the medical domain, where additional data recording may be costly. To overcome this shortcoming, often synthetic data generation is used to artificially boost datasets. In the case of image generation generative models can be used to adapt simulated images to realistic-looking ones. Fig. 3.1 shows an example of one image being adapted. A more detailed explanation and comprehensive list of methods is given in [21].

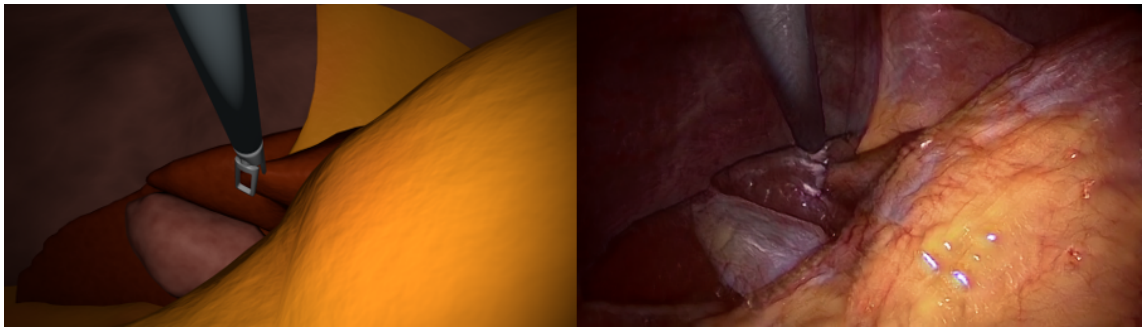


Figure 3.1: A simulated image (left) is transferred to have a realistic look (right). Images by Pfeiffer et. al. [1]

3.3 Arbitrary Style Transfer

Arbitrary style transfer [10] is a problem class, founded on the premise, that an image consists of semantic content and style information. For example, a painting can depict a scenery with the complete same objects and their positions the same way, as a photograph could. If this was the case, both images would share the same content but are represented in different styles. On the other hand, two images depicting different scenes might be drawn by the same artist, with the same technique. These images would share the same style but have different content. The goal of arbitrary style transfer, as described by Huang et. al. [10], is to combine the content of one image with the style of another image. The result would then be an image still showing the original scenery but appearing visually different (See figure 3.2).

3.3.1 AdaIn and AdaAttn

Adaptive Instance Normalization (AdaIN) [10] and Adaptive Attention Normalization (AdaAttn) [11] both utilize an Encoder-Decoder Architecture, where a pre-trained frozen encoder network f is used. A content image c and an arbitrary style image s are fed through the encoder up until a certain layer. The feature maps of the encoder's layers will then be used in the AdaIn or AdaAttn Mechanism M to produce target feature maps

$$t = M(f(c), f(s)), \tag{3.1}$$

which are then used in a decoder H to generate the respective styled output image

$$T(c, s) = H(t). \tag{3.2}$$

AdaIN aligns the statistics of two feature maps. The style of an image is hereby seen as a distribution of features in the embedded input image. The mean $\mu(F_s^l)$ and standard deviation $\sigma(F_s^l)$ of every feature at a chosen layer l is taken from the processed style input $f(s)$. These are then applied to the features F_c^l of the processed content input $f(c)$, by suppressing the current mean and standard deviation and applying the ones from s. The mechanism is formulated as

$$\text{AdaIN}(F_c^l, F_s^l) = \sigma(F_s^l)\left(\frac{F_c^l - \mu(F_c^l)}{\sigma(F_c^l)}\right) + \mu(F_s^l) \tag{3.3}$$

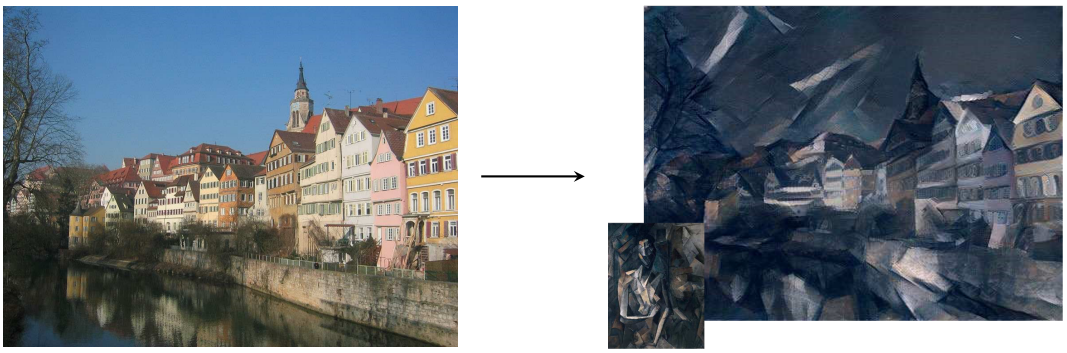
AdaAttn is an extension of the feature alignment concept of AdaIn to a per-pixel level and attention to weighing different features from multiple layers of the encoding feature maps F_*^l, where $*$ may either be the content input c or the style input s. AdaAttn adopts a multi-layer strategy by concatenating down-sampled features of multiple layers with

$$F_*^{1:x} = D_x(F_*^1) \oplus D_x(F_*^2) \oplus \cdots \oplus (F_*^x), \tag{3.4}$$

where D_x is a bilinear interpolation, downscaling to the same shape of (F_*^x), and $\oplus$ is the concatenation along the channel axis.

$\text{AdaAttn}(F_c^x, F_s^x, F_c^{1:x}, F_s^{1:x})$ is then computed by the following steps. Querry Q, Key K, and Value V of the attention mechanism are calculated by

$$\begin{aligned} Q &= f(Norm(F_c^{1:x})), \\ K &= g(Norm(F_s^{1:x})), \\ V &= h(F_s^x), \end{aligned} \tag{3.5}$$



Content Image Result styled by a style image

Figure 3.2: Example illustrating Arbitrary Style Transfer. Images adapted from Gatys et. al. [14].

where f, g, and h are 1×1 convolutions and $Norm$ is the channel-wise mean-variance normalization. Note that V is only dependent on features from the current style layer l. The attention Map A describes the similarity between content and style features. It is calculated by

$$A = Softmax(Q^\top \otimes K) \tag{3.6}$$

where $\otimes$ is the matrix multiplication. With that, an attention-weighted mean

$$M = V \otimes A^\top, \tag{3.7}$$

and an attention-weighted standard deviation

$$S = \sqrt{(V \cdot V) \otimes A^\top - M \cdot M} \tag{3.8}$$

is calculated, where $\cdot$ is the element-wise matrix product. Finally, for each position and channel of the normalized content feature map, the corresponding attention-weighted standard deviation S and attention-weighted mean M are applied to the style feature maps to generate transformed feature maps

$$t = S \cdot Norm(F_c^x) + M. \tag{3.9}$$

Further details can be found in [11].

3.4 Semantic Image Segmentation

Semantic image segmentation [22] is a technique, where if an image is given, a partition of pixels shall be found in a way that every pixel corresponds to one of n predefined classes. These classes usually relate to kinds of physical objects, which may be present in the scene and are relevant to the problem at hand. An example of such a problem in laparoscopic surgery is shown in fig. 3.3. Anatomical tissues such as the abdominal wall, liver, etc... are annotated per pixel. A segmentation algorithm may now attempt to find this partition. An overview of methods is provided by Minaee et. al. [23].

3.5 Generative Adversarial Networks

Generative adversarial networks (GANs)[13] are a type of machine learning architecture for creating certain types of images similar to a training dataset. If for example a GAN

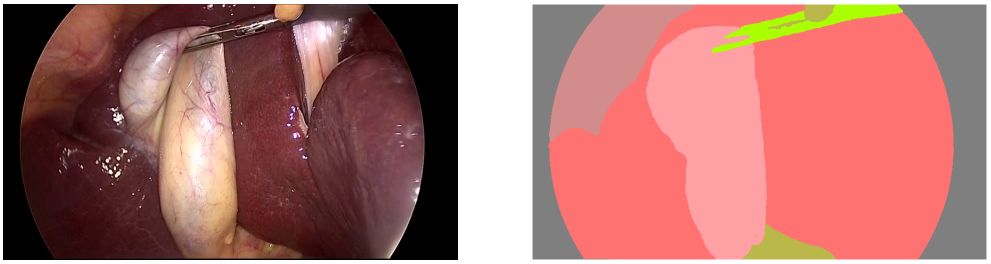


Figure 3.3: Example of a segmentation. An image of a laparoscopic scene is partitioned into a border mask, abdominal wall liver, tools, fat, and gallbladder. Images from the CholecSeg8K Dataset [24]

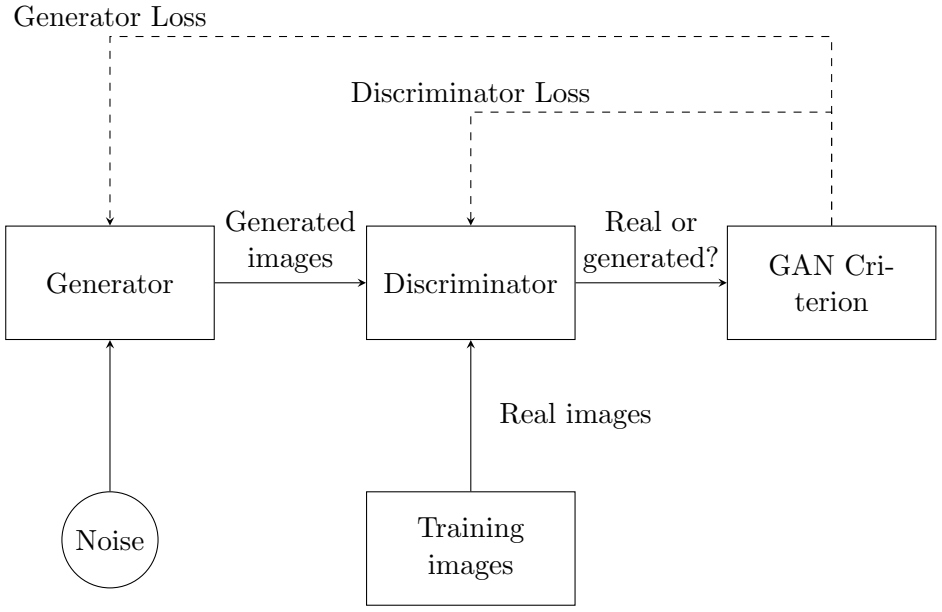


Figure 3.4: Principle of Generative Adversarial Networks (GANs)

is trained with a set of images of human faces, it may output an image of a human face, which resembles characteristics of the training dataset but is itself not part of it. The GAN architecture is made up of a generator network G and a discriminator network D. G usually accepts a random noise vector z as input and outputs an image $G(z)$ with the same dimensions as the ones in the dataset. D is presented with w which is either the output of G or an image from the dataset and outputs a scalar s, which represents a choice whether D classified w as part of the dataset or as created by G. Both networks are then optimized with contradicting goals. G's objective is to lower the accuracy of s regarding the actual origin of w and D's objective is to increase this accuracy. Fig. 3.4 illustrates the GAN mechanism. Therefore during training a competing methodology is created where D gets better at detecting "fake" images and G gets better at "faking" images.

3.6 Contrastive unpaired translation

The unpaired image to image problem consists of an input domain $\mathcal{X} \subset \mathbb{R}^{H \times W \times C}$, an output domain $\mathcal{Y} \subset \mathbb{R}^{H \times W \times C}$ of images and two sets $X \subset \mathcal{X}$ and $Y \subset \mathcal{Y}$ where no direct semantic link $\equiv$ exists between any pair: $\forall x \in X, \forall y \in Y : x \not\equiv y$. CUT learns a translation from an input image $x \in X$ to an output image y, where the semantical similarity between x and y shall be achieved and the output image should look like an image from the output domain: $y \in \mathcal{Y}$. For the training, there is a generator network G and a discriminator

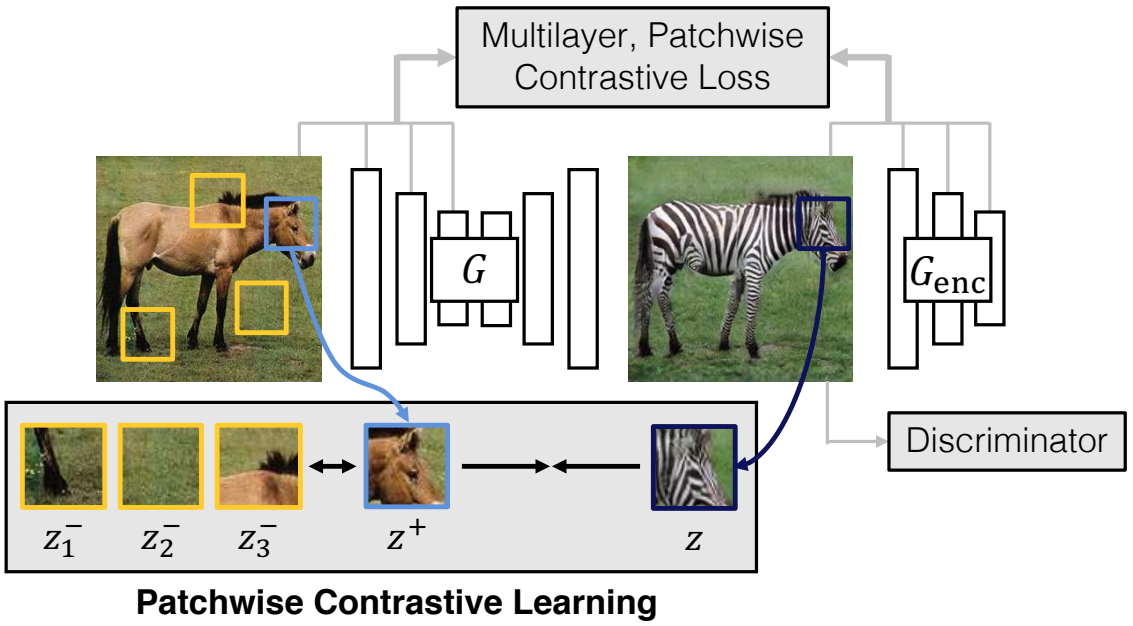


Figure 3.5: Illustration of the CUT architecture. Generator-patches z_1^-, z_2^-, z_3^-, z^+ of features of the input and are compared with the patch z of the translated input, which is re-inserted into the generator. The correspondence between z_1^-, z_2^-, z_3^- and z is decreased by training, while the correspondence between z^+ and z is increased. The translated image is decided by the discriminator to train the model to match the output domain style. Image by Park et. al.[8]

function D set up to be used as a generative adversarial network [13]. It is trained to resemble images from Y and produces the generator loss

$$\mathcal{L}_{GAN,G}(G, D, X, Y) = \mathbb{E}_{y\sim Y} \log D(y) + \mathbb{E}_{x\sim X} \log(1 - D(G(x))). \tag{3.10}$$

To achieve semantical similarity with the generator, the authors introduce another loss called $\mathcal{L}_{\text{PatchNCE}}$. For that, x is forwarded through the generator. If the generator is made up of L layers and every layer $l \in L$ has spatial locations $s \in S^l$, then $z_l^s \in \mathbb{C}^{C_1}$ describes a patch of a feature of uniform size at this location and layer which was passed through an MLP H_l, for x and C_1 as the number of features in the layer. The same will be done with the generator's output image $G(x)$ as it is passed through the generator for a second time resulting in $\hat{z}_l^s$. Also $z_l^{S\setminus s} \in \mathbb{R}^{(S_l-1)} \times C_l$ are generated from several features which are at different spatial locations than s. A cross-entropy function ℓ is used to minimize similarity between z_l^s and $z_l^{S\setminus s}$ and maximize the similarity between z_l^s and $\hat{z}_l^s$.

$$\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{x\sim X} \sum_{l=1}^{L} \sum_{s=1}^{S} \ell(\hat{z}_l^s, z_l^s, z_l^{S\setminus s}). \tag{3.11}$$

Together the generator loss $\mathcal{L}_G$ is formulated as

$$\mathcal{L}_G = \mathcal{L}_{GAN,G}(G, D, X, Y) + \lambda_X \mathcal{L}_{PatchNCE}(G, H, X) + \lambda_Y \mathcal{L}_{PatchNCE}(G, H, Y). \tag{3.12}$$

Fig. 3.5 illustrates the architecture.

4. Methods

The learning process of the target domain relies, among others, on the GAN mechanism [13] in the contrastive unpaired translation (CUT) model by Park et. al. [8]. On the other side there are style transfer models [10, 25, 11]. We want to use part of their architecture, to guide the process towards better matching our target domain. To do this, we implement an arbitrary style transfer (AST) mechanism into CUT. Pfeiffer et al. [1] used data from the 3D-IRCADb-01 (IRCAD, France) database to create 3D renderings resembling minimal invasive surgeries from the perspective of a laparoscope's camera. These are in the following referred to as "synthetic images". Another dataset used was the Cholec80 [26] dataset where a set of frames was extracted from. The dataset contains videos of cholecystectomy procedures. These will be referred to as "real images". A synthetic and a real image will be fed through the generator of CUT. Similar to CUT's contrastive approach, features from different layers will be extracted. The features from the real image will then be used as a style input for the AST mechanism alongside features from the synthetic image which will be used as content input. The result is then reinserted into the generator, after the layer where features were taken from, and further processed back into an image.

4.1 ResNet-based Encoder-Decoder Generator

The generator is a ResNet-based encoder-decoder [27] as used by Park et. al. [8]. In contrast to the methods by [10] and [11], there is no frozen pre-trained encoder, but a randomly initialized generator G. It can be divided into three functional Blocks.

4.1.0.1 Down-scaling Layers

The initial layers of the generator focus on down-scaling the input image. A reflection padding layer is first applied to handle edge effects, followed by a 7×7 convolutional layer with 64 filters, instance normalization with no learnable parameters, and a Rectified Linear Unit (ReLU) activation function. Subsequent downscaling is achieved through a 3×3 convolutional layer with 128 filters, instance normalization, and ReLU activation. Downsampling operations further facilitate the process, each employing reflection padding and 3×3 convolutional layers with increasing filter sizes (128 to 256).

4.1.0.2 ResNet Blocks

Then are ResNet blocks $r_1, r_2, ..., r_n$, which operate on the down-scaled feature maps and make up most of the generator layers. Each ResNet block follows a common structure,

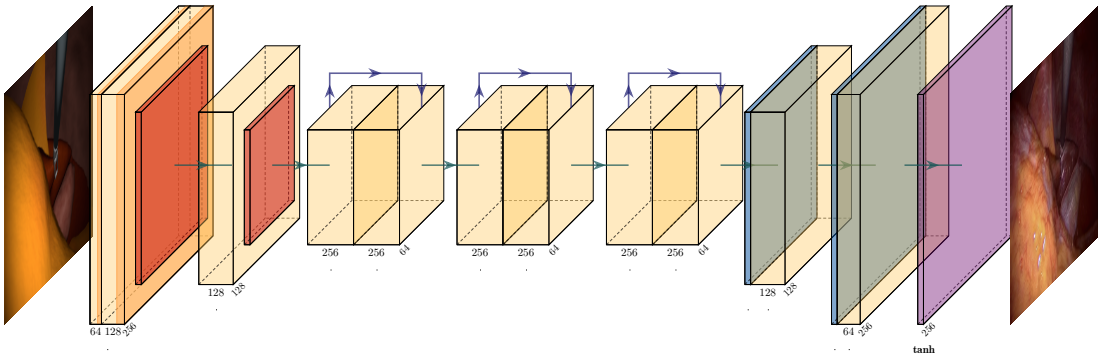


Figure 4.1: Structure of a ResNet-based encoder-decoder. The input will be down-scaled by two pooling layers (red). Three ResNet Blocks with two convolutions (yellow) each follow, to translate the down-scaled features. Finally, the features are scaled up again (blue) two times and the three color channels pass a tanh activation function. The translated image is extracted.

consisting of two 3×3 convolutional layers with 256 filters, instance normalization, and ReLU activation. The use of residual connections helps mitigate vanishing gradient issues [28].

4.1.0.3 Up-scaling Layers

Following the ResNet blocks, the generator includes up-scaling layers responsible for restoring the spatial dimensions of the feature maps. These layers employ upsampling operations, each utilizing replication padding and 3×3 convolutional layers with decreasing filter sizes (256 to 128). The final layer consists of a 7×7 convolutional layer with three filters, instance normalization, and a hyperbolic tangent (Tanh) activation function to ensure the output pixel values fall within the range [-1, 1].

An illustration is available in fig. 4.1.

4.2 Style Adaptation

The above-mentioned style transfer mechanisms shall be incorporated into the generator of CUT. Let $G_{r_k}(\alpha)$ be the activation maps after the r_k ResNet block of the generator G by processing image α. Style adaptation uses a synthetic image $a \in X$ as semantic content and a real image $s \in Y$ as style. A style s adapted version t of the activations of a after r_k is calculated by

$$t_a = M(G_{r_k}(a), G_{r_k}(s)). \tag{4.1}$$

A style s adapted version is also calculated of s itself. This is done to keep the forwarding method consistent between both a and s as s might be needed for a style adaptation step at a later ResNet block, as t_a is forwarded through the layers of G which come after r_k and style adaptation may reoccur multiple times. The layers after which style adaptation happens are added as a hyperparameter to the architecture. The image is finally scaled up again to be the output image b.

In the case of AdaIn, if u, v are activation maps, the mechanism M is defined as

$$M(u, v) = \text{AdaIn}(u, v). \tag{4.2}$$

In AdaAttn, aggregated versions of activations of multiple layers of the encoder are used for M. Only the activations of one layer will be used, as the generator is not frozen during

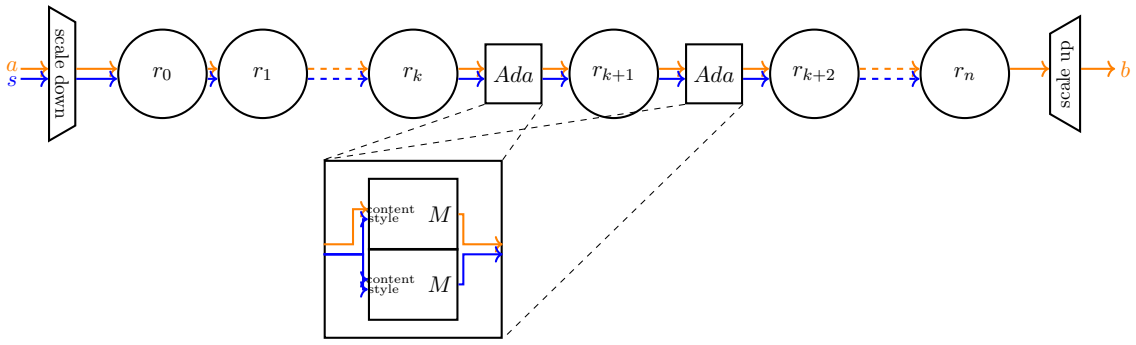


Figure 4.2: Generator with style adaptation added in between ResNet blocks

training and therefore able to leave important information in place for M. Therefore M is defined as

$$M(u, v) = \text{AdaAttn}(\text{u}, \text{v}, \text{u}, \text{v}), \tag{4.3}$$

if AdaAttn is used. M is then trained along with the generator and its loss function $\mathcal{L}_G$.

5. Results

A downstream task is applied to see the usefulness of the images generated by a model. Semantic segmentation can tell how well a tissue type can be recognized and differentiated from other types. Multiple methods are picked for evaluation, which all include training and testing a DeeplabV3 Architecture [29] for segmentation Tasks. Resnet34[28] is used as an encoder for the segmentation model, which is trained on labels for the background, the abdominal wall, the liver, fat tissue, tools, ligament tissue, and the gallbladder. The CholecSeg8K [24] dataset is split into 3,233 test images and 4,849 training images and an independent test model is trained. 10% of the training dataset will be held out of training and be used for validation at the end of each epoch. All following segmentation training and finetuning procedures are done by training the architecture for 100 epochs and picking the checkpoint with the best validation image-IOU value. This value is calculated by averaging all the Jacquard indices (IOUs) of all images. Let N be all images then n is the number of images, and $T_{pos}, T_{neg}, F_{pos}, F_{neg} \in N$ which are truly positive, truly negative, falsely positive and falsely negative classified pixels independent of class. The IOU s is calculated by

$$s = \frac{1}{n} \sum_{k=0}^{n} \frac{T_{pos}}{T_{pos} + F_{pos} + F_{neg}}. \tag{5.1}$$

Four different methods to asses generated image quality are applied. Label preservation is evaluated to pick the best epochs for all models. A segmentation model can then be trained with these best epochs, and this downstream task's usability can be evaluated. To get further insight into the results, an FID score is taken per epoch and a visual examination is done.

5.1 CUT training

CUT and the style transfer are randomly initialized before epoch zero. The settings of the experiments of Park et. al. [8] are followed. A training subset of 9,888 synthetic and 11,607 real images is used to train five different models for each AdaIN and AdaAttn as style transfer mechanisms. Zero to four Adaptation blocks will be used in these models. Each model is trained up to 30 epochs and a checkpoint of the model is saved after an epoch is completed. A generator with 9 ResNet blocks is used, where each Style Adaptation Block is inserted between two ResNet blocks. Multiple training attempts with this setup went to unstable training. Therefore PatchGAN discriminator [30] network's training is regularized using Spectral Norm [31] and a learning rate $\mathrm{lr} = 0.00002$ is used. See fig. 5.1

Experiment Name	Adaptation Mechanism	Adaptation Layers
Baseline	-	-
AdaIN 1	AdaIN	0
AdaIN 2	AdaIN	0,1
AdaIN 3	AdaIN	0,1,2
AdaIN 4	AdaIN	0,1,2,3
AdaAttn 1	AdaAttn	0
AdaAttn 2	AdaAttn	0,1
AdaAttn 3	AdaAttn	0,1,2
AdaAttn 4	AdaAttn	0,1,2,3

Table 5.1: Hyperparameters for different models

for a list of model hyperparameters. A test split of 2333 synthetic and 2,000 real images is then applied for each model and epoch and the generated images are saved. An overview of generated images is given in fig. 5.1.

5.2 Mask Preservation Task

Mask preservation refers to using an available segmentation model following [8, 2, 32, 33, 34]. During the process of rendering the synthetic images, the synthetic segmentation masks were rendered alongside. The independent test model is applied to our generated test images from all epochs and experiments and the resulting masks are compared to their respective synthetic masks, acting as ground truth. In this test, different models are compared against the unmodified CUT model. Resulting IOU scores are plotted in fig. 5.2 and fig. 5.3. All curves shown there hold little statistical significance, as all experiments could only be done once due to their high demand for computing power. As the results heavily oscillate and overlap each other, for both generator types, no statement can be made about the superiority of one over another by this method. The average of all methods goes down over the epochs and stagnates around epoch 12 for AdaIn and epoch 14 for AdaAttn. This indicates, that during training, all models seem to lose focus on details that are important for the segmentation model. The decline in mask preservation over generator epochs indicates, that the training with all generators reduced the stylistic similarity and therefore removed visual information, which could have been used by a segmentation model.

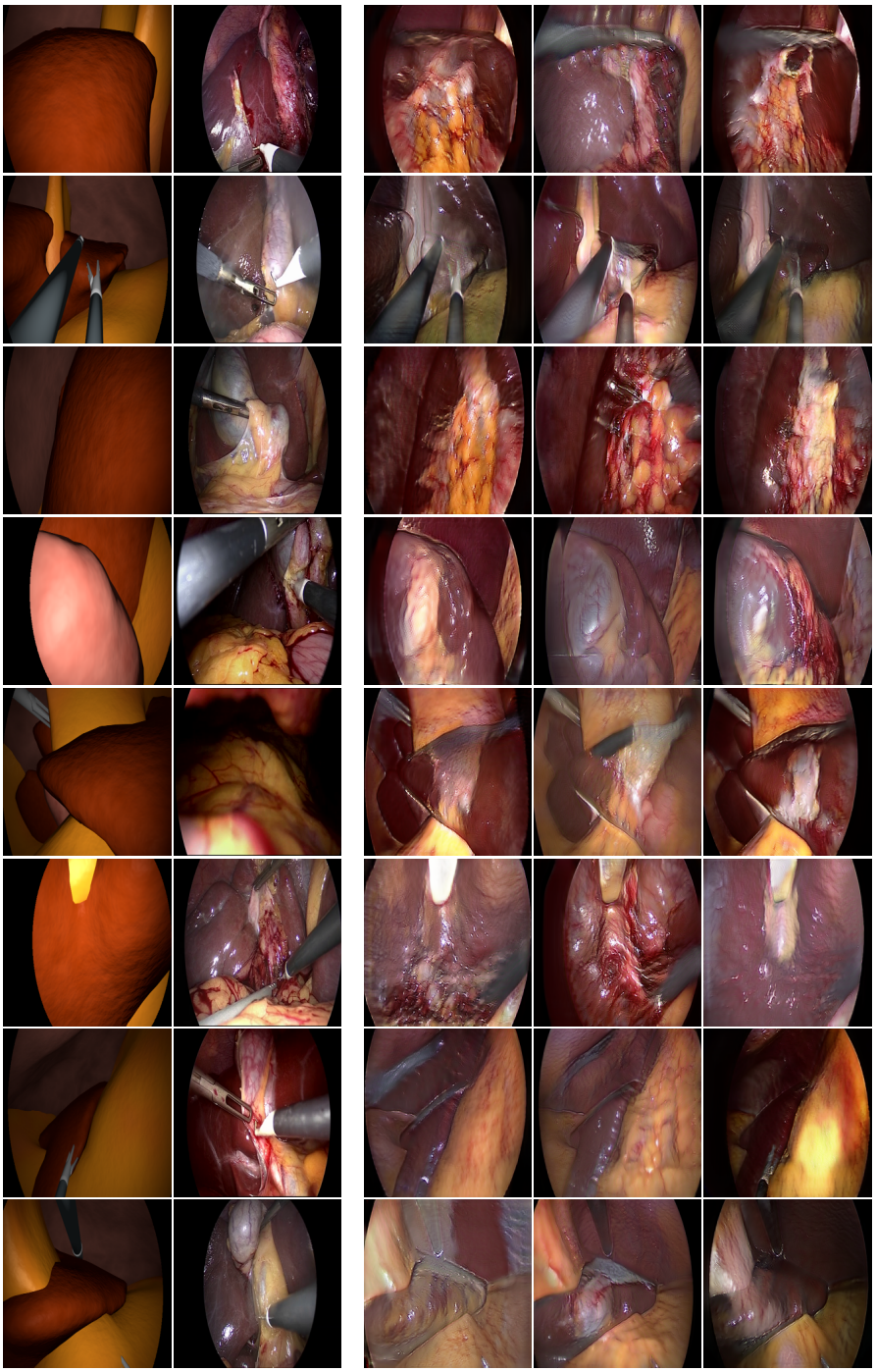


Figure 5.1: Overview of images generated by the three different generators.

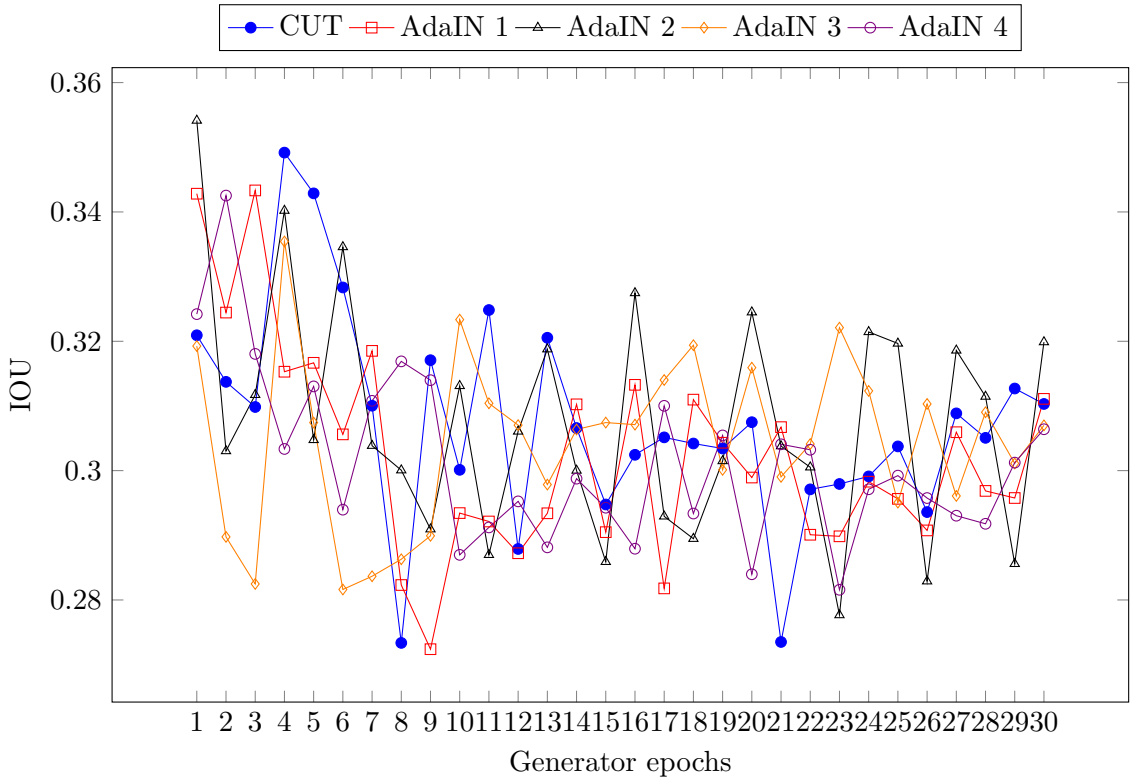


Figure 5.2: IOU scores of the generators with different numbers of supporting AdaIN layers and the baseline. Plotted in x-direction are the number of epochs the generator was trained for.

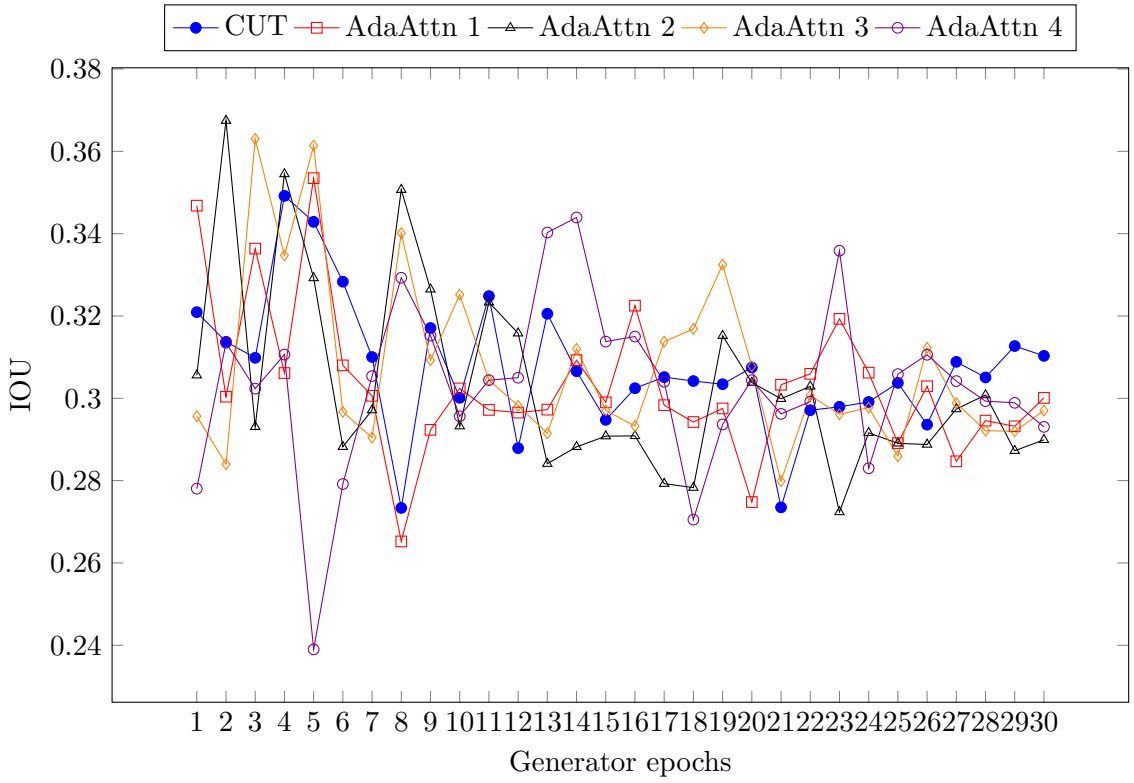


Figure 5.3: IOU scores of the generators with different numbers of supporting AdaAttn layers and the baseline. Plotted in x-direction are the number of epochs the generator was trained for.

5.3 Training segmentation models with generated images

A segmentation model is trained with the generated images and real images which were annotated by hand. This evaluation matches the task we set as the goal of increasing the segmentation performance. It is therefore the most representative result. It is interesting if the addition of the style adaptation layers increases the segmentation performance and if the addition of the generated images in general increases the segmentation performance. The experiments from CUT with no style adaptation blocks (SAB) added, will serve as a baseline and be compared against all other experiments. The best epoch is taken from the experiments with no SAB as well as one from experiments where SABs were present for the label preservation task. The latter are AdaIN and AdaAttn experiments. A segmentation model will be trained with the respective generated images and synthetic mask of the best-generating model with regards to results from mask preservation (section 5.2). The segmentation model will be fine-tuned with the train split of CholecSeg8K [24] to evaluate the effects of combined training with generated and human-annotated images. Note that the latter is larger. A test split of CholecSeg8K is then used to retrieve mean IOU scores in the same way as mentioned above. A segmentation model is trained with only the train split of the cholec8K dataset and no pretraining is done by the generated images. The difference between the CUT baseline and no-pretraining is that the first can be used to compare the usefulness of the additions to the architecture and the second can be used for a statement about the usefulness of pretaining with the generated images. CUT showed a significant drop of 3% over no-pretraining. AdaIN and AdaAttn increased the results over the CUT baseline, except for AdaAttn4, but also remained short of the no-pretraining results. That means, that although they showed an improvement, they are still not capable of increasing the overall segmentation performance. This might hint at those generators being better at preserving information, that is vital to the segmentation algorithm. Results are shown in fig. 5.4.

5.4 FID

To compare visual similarity between real and fake images, the Fréchet inception distance (FID) [12] is taken between the generated and real images from the test dataset of CholecSeg8K. Features of InceptionNetV2 [35] and a batch size of 100 is used. Results are shown in fig. 5.5 and fig. 5.6. Until around epoch 12 of training, a clear improvement of all generator's scores can be seen. Then all generators will normalize around a distance of 100 for the rest of the training. This is contrary to the label preservation task, where the quality of the results decreases at the beginning of training. No improvement of AdaIN and AdaAttn can be recognized over unmodified CUT[8] after the initial descent of distances.

5.5 Visual Examination

Sets of synthetic-, style-, baseline-, AdaIN-, and AdaAttn- images where visually compared side by side. A few notable examples will be discussed.

Style transfer: Synthetic and style image, showed the same organ, and the AdaAttn mechanism was able to transfer a texture (fig. 5.7). A problem arising from the fact, that random images are paired one to one during training is, that style information needed for a synthetic scene might not be available for the style image.

Halloucinations: It occurred multiple times, that images, which contained big uniform surfaces, would have a lot of inadequate details added to them by all three generators. The surface of the liver for example was split into two different kinds of tissues by the AdaIN generator(fig. 5.8). A lot of hallucinations can also be spotted in the overview in

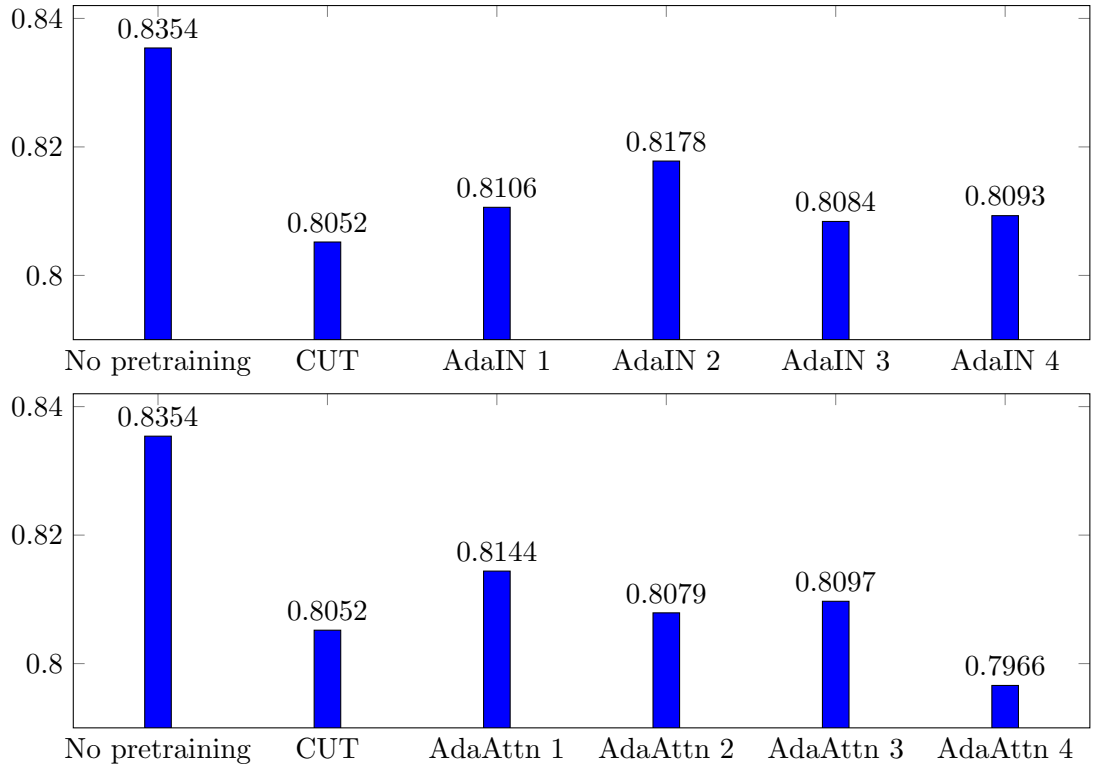


Figure 5.4: Test results of a segmentation model trained with generated images. No pretraining relates to training a model with the train split of the cholec8K dataset and no generated images.

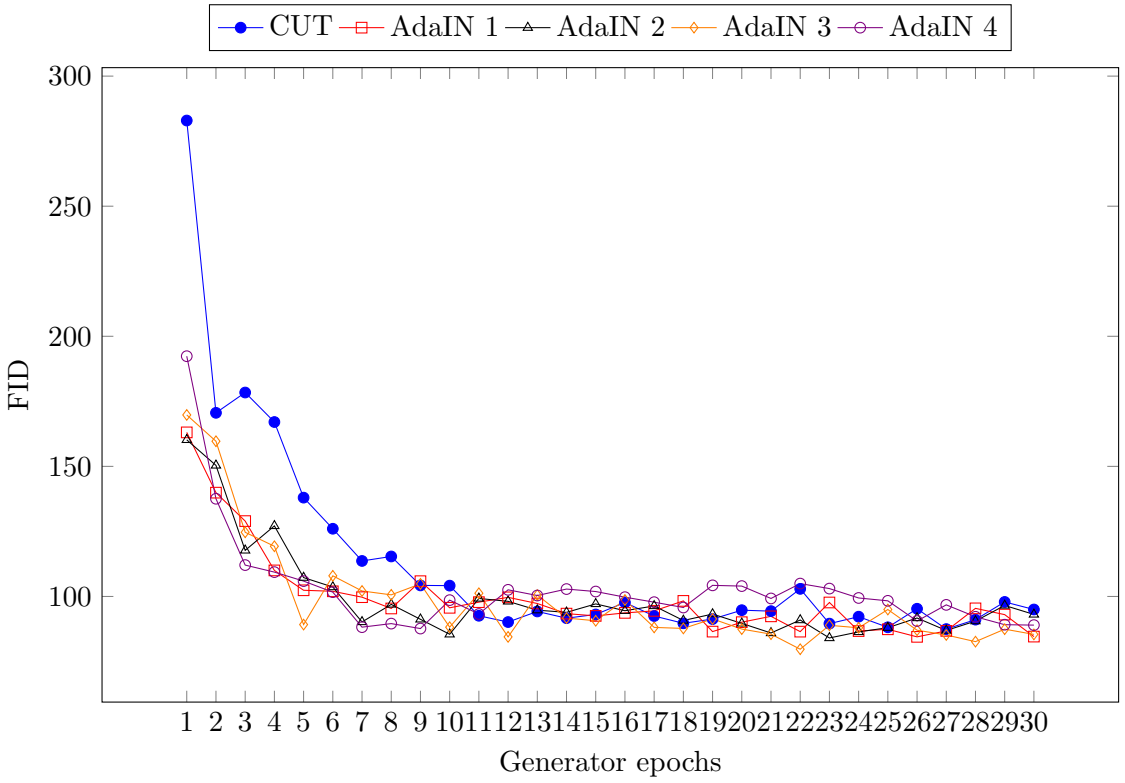


Figure 5.5: FID scores of the generators with different numbers of supporting AdaAttn layers and the baseline. Plotted in x-direction are the number of epochs the generator was trained for.

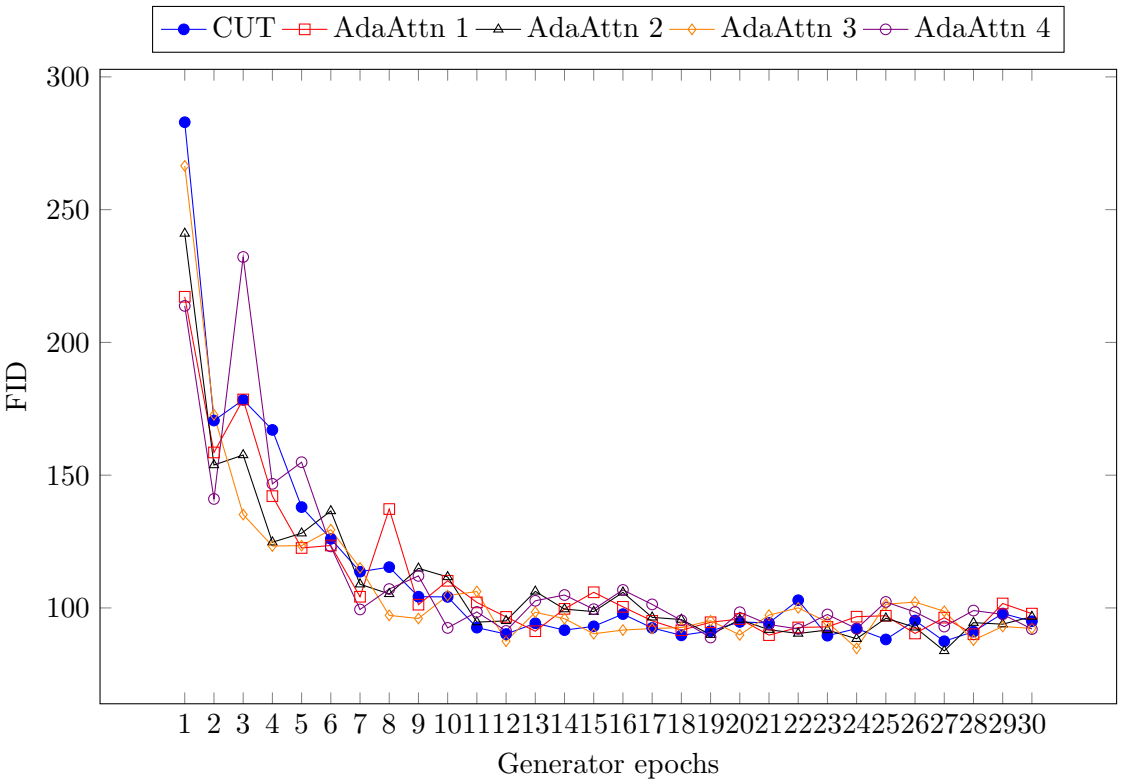
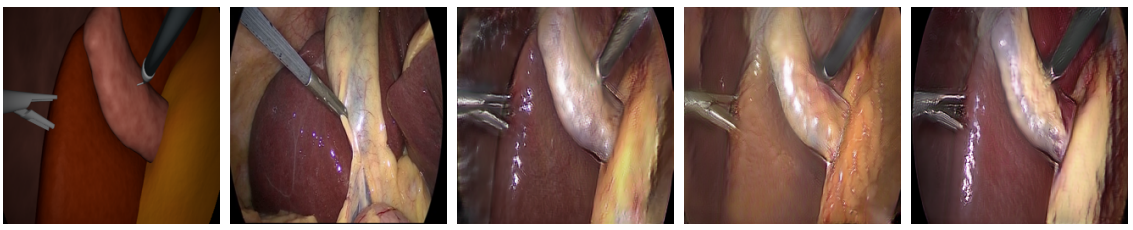


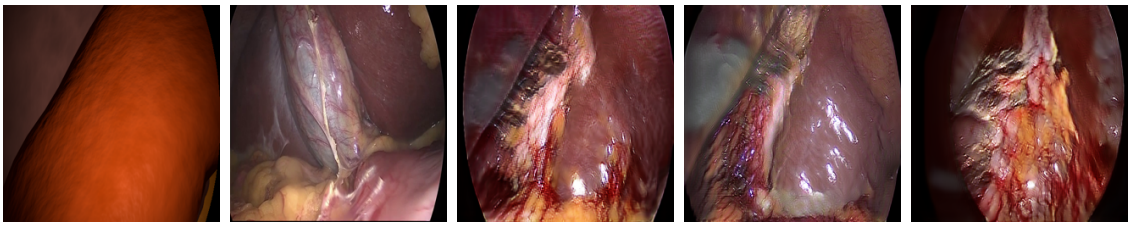
Figure 5.6: FID scores of the generators with different numbers of supporting AdaAttn layers and the baseline. Plotted in x-direction are the number of epochs the generator was trained for.

fig. 5.1. The authors of CUT mention, that it is less able than CycleGAN[6], to adapt to statistical imbalances in the input and output domains. A property of an output of CUT will for example be more likely to match the percentage of area of the target domain than to match the same property of the input image. This inability could be responsible for creating inadequate semantical patterns (hallucinations) in uniform surfaces, as they are less present in the target domain dataset.



Synthetic image Style image Baseline AdaIN AdaAttn

Figure 5.7: The gallbladder transferred by the AdaAttn-supported generator has a finer detailed texture that resembles the gallbladder in the style image.



Synthetic image Style image Baseline AdaIN AdaAttn

Figure 5.8: A synthetic image with a big uniform surface and little variety due to the organs shown. All three methods develop visual components that do not exist in the synthetic image. This is an example of the failure case of this approach.

6. Conclusion

The goal of this thesis was to increase the performance of a segmentation model, that was trained with generated images. The CUT architecture was modified to include AdaIN[10] and AdaAttn[11] as style transfer steps. Multiple evaluations were done to assess the quality of generated images by those three image generators.

Training segmentation models with generated images showed mixed results. No segmentation model, that was pretrained with generated images, could beat the unpretrained segmentation model. AdaIN and AdaAttn demonstrated some improvement over the baseline CUT model. As this test reflects our goal most accurately, it suggests that the modified generators are better at preserving information vital to segmentation but are not yet capable of enhancing overall segmentation performance.

Evaluating mask preservation capabilities demonstrated a decline in the ability of models, to preserve details important for segmentation. Mask Preservation refers to the possibility of synthetic masks being restored of a generated image. This decrease occurred at the beginning of the training, indicating a loss of focus on important details during the generator training process. This trend is contrary to the FID[12] scores over epochs, which increased at the beginning of training. As the label preservation task is more similar to the goal of segmentation, it is concluded, that these scores hold little relevance for the problem of image transfer in this thesis.

Examining the images qualitatively leads to formulating hypotheses about the above results. Instances of successful style transfer were seen, but are theoretically limited by the semantical overlap of tissue types in content and style image. Hallucinations, where details were mistakenly added to uniform surfaces, were very common. This might point to challenges in adapting to statistical imbalances.

Future Work

- Statistically adjusted datasets, with regards to the proportion of areas of different tissue types, could be used for training to explore if statistical imbalances lead to hallucinations in the generated images.
- The CUT[8] architecture was modified by Venkatesh et. al. [2], to meet a similar goal like within this thesis. Their method could be combined with the style adaptation mechanism, to evaluate if it can increase results.

- The problem of semantical mismatch in style and content image could be mitigated, by using multiple style image inputs, or matching tissue types in the content-style pairs of the adaptation mechanism. The latter one is impractical, as long as there is no automated way of matching tissue types.

List of Figures

Bibliography

[1] M. Pfeiffer, I. Funke, M. R. Robu, S. Bodenstedt, L. Strenger, S. Engelhardt, T. Roß, M. J. Clarkson, K. Gurusamy, B. R. Davidson, L. Maier-Hein, C. Riediger, T. Welsch, J. Weitz, and S. Speidel, "Generating large labeled data sets for laparoscopic image processing tasks using unpaired image-to-image translation," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham: Springer International Publishing, 2019, pp. 119–127.

[2] D. K. Venkatesh, D. Rivoir, M. Pfeiffer, F. Kolbinger, M. Distler, J. Weitz, and S. Speidel, "Exploring Semantic Consistency in Unpaired Image Translation to Generate Data for Surgical Applications," Sep. 2023, arXiv:2309.03048 [cs]. [Online]. Available: http://arxiv.org/abs/2309.03048

[3] J. Kaleta, D. Dall'Alba, S. Płotka, and P. Korzeniowski, "Minimal data requirement for realistic endoscopic image generation with Stable Diffusion," *International Journal of Computer Assisted Radiology and Surgery*, Nov. 2023. [Online]. Available: https://doi.org/10.1007/s11548-023-03030-w

[4] T. Dowrick, B. Davidson, K. Gurusamy, and M. J. Clarkson, "Large scale simulation of labeled intraoperative scenes in unity," *International Journal of Computer Assisted Radiology and Surgery*, vol. 17, no. 5, pp. 961–963, May 2022. [Online]. Available: https://doi.org/10.1007/s11548-022-02598-z

[5] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[7] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1857–1865. [Online]. Available: https://proceedings.mlr.press/v70/kim17a.html

[8] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 319–345.

[9] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, Nov. 2003, pp. 1398–1402 Vol.2. [Online]. Available: https://ieeexplore.ieee.org/document/1292216

[10] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 1510–1519. [Online]. Available: http://ieeexplore.ieee.org/document/8237429/

[11] S. Liu, T. Lin, D. He, F. Li, M. Wang, X. Li, Z. Sun, Q. Li, and E. Ding, "AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 6629–6638. [Online]. Available: https://ieeexplore.ieee.org/document/9710208/

[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," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[14] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 2414–2423. [Online]. Available: http://ieeexplore.ieee.org/document/7780634/

[15] B. Yu, L. Zhou, L. Wang, Y. Shi, J. Fripp, and P. Bourgeat, "Ea-GANs: Edge-Aware Generative Adversarial Networks for Cross-Modality MR Image Synthesis," *IEEE Transactions on Medical Imaging*, vol. 38, no. 7, pp. 1750–1762, Jul. 2019, conference Name: IEEE Transactions on Medical Imaging. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8629301

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022, arXiv:2112.10752 [cs]. [Online]. Available: http://arxiv.org/abs/2112.10752

[17] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation," Mar. 2023, arXiv:2208.12242 [cs]. [Online]. Available: http://arxiv.org/abs/2208.12242

[18] L. Zhang, A. Rao, and M. Agrawala, "Adding Conditional Control to Text-to-Image Diffusion Models," Nov. 2023, arXiv:2302.05543 [cs]. [Online]. Available: http://arxiv.org/abs/2302.05543

[19] D. Rivoir, M. Pfeiffer, R. Docea, F. Kolbinger, C. Riediger, J. Weitz, and S. Speidel, "Long-term temporally consistent unpaired video translation from simulated surgical 3d data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3343–3353.

[20] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred Neural Rendering: Image Synthesis using Neural Textures," Apr. 2019, arXiv:1904.12356 [cs]. [Online]. Available: http://arxiv.org/abs/1904.12356

[21] Y. Lu, M. Shen, H. Wang, X. Wang, C. van Rechem, and W. Wei, "Machine Learning for Synthetic Data Generation: A Review," Jan. 2024, arXiv:2302.04062 [cs]. [Online]. Available: http://arxiv.org/abs/2302.04062

[22] C. R. Brice and C. L. Fennema, "Scene analysis using regions," *Artificial Intelligence*, vol. 1, no. 3, pp. 205–226, 1970. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370270900081

[23] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," 2020.

[24] W.-Y. Hong, C.-L. Kao, Y.-H. Kuo, J.-R. Wang, W.-L. Chang, and C.-S. Shih, "CholecSeg8k: A Semantic Segmentation Dataset for Laparoscopic Cholecystectomy Based on Cholec80," Dec. 2020, arXiv:2012.12453 [cs]. [Online]. Available: http://arxiv.org/abs/2012.12453

[25] D. Y. Park and K. H. Lee, "Arbitrary Style Transfer With Style-Attentional Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 5873–5881. [Online]. Available: https://ieeexplore.ieee.org/document/8953228/

[26] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos," May 2016, arXiv:1602.03012 [cs]. [Online]. Available: http://arxiv.org/abs/1602.03012

[27] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 694–711.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. [Online]. Available: http://ieeexplore.ieee.org/document/7780459/

[29] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," Dec. 2017, arXiv:1706.05587 [cs]. [Online]. Available: http://arxiv.org/abs/1706.05587

[30] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, Jul. 2017, pp. 5967–5976. [Online]. Available: http://ieeexplore.ieee.org/document/8100115/

[31] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral Normalization for Generative Adversarial Networks," Feb. 2018, arXiv:1802.05957 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1802.05957

[32] C. Chu, A. Zhmoginov, and M. Sandler, "CycleGAN, a Master of Steganography," Dec. 2017, arXiv:1712.02950 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1712.02950

[33] H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang, and D. Tao, "Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping," Nov. 2018, arXiv:1809.05852 [cs]. [Online]. Available: http://arxiv.org/abs/1809.05852

[34] J. Yoon, S. Hong, S. Hong, J. Lee, S. Shin, B. Park, N. Sung, H. Yu, S. Kim, S. Park, W. J. Hyung, and M.-K. Choi, "Surgical Scene Segmentation Using Semantic Image

Synthesis with a Virtual Surgery Environment," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, ser. Lecture Notes in Computer Science, L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, Eds. Cham: Springer Nature Switzerland, 2022, pp. 551–561.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167