Fakultät für Ingenieurwesen
Facoltà di Ingegneria
Faculty of Engineering

Free University of Bozen-Bolzano
Faculty of Engineering
Master Degree in Computational Data Science

Improving Semantic Segmentation Models through Synthetic Data Generation via Diffusion Models

Author: Jonas Rabensteiner **ID:** 21196
Supervisor: Prof. Oswald Lanz
Co-supervisor: Ing. Cynthia I. Ugwu

14th September 2023

Abstract

Industrial datasets for semantic segmentation are typically scarce because of the time-consuming annotation procedure. However, modern deep learning model architectures for this task require a large amount of data and therefore perform below their potential capabilities when trained on small datasets. This makes semantic segmentation on industrial datasets a suitable candidate for the application of current advances with generative models. Particularly, recent Diffusion Models (DMs), which are mainly known for applications in the field of art, can be exploited to enhance existing datasets with synthetic data. In principle, DMs can be used to sample arbitrary amounts of data and thus to adapt the sparse existing datasets such that the models for semantic segmentation can be better trained on them. In the case of semantic segmentation, however, it is not enough just to generate the images, the images must also match with corresponding labels on a pixel level. In this thesis, two main approaches, namely semantic image synthesis and paired image-mask synthesis, are discussed in detail. Furthermore, two implementations for these approaches are proposed that allow training DMs on small-scale industrial semantic segmentation datasets and generating corresponding synthetic datasets. These implementations are referred to as "All-Latent" and "Improved Semantic Diffusion Model" (ISDM), named after the types of DMs on which they are built. Extensive experiments with three different semantic segmentation architectures are conducted to evaluate the impact of the synthetic datasets generated by All-Latent and ISDM in terms of the mean intersection-over-union (mIoU) score. They show that for both methods the generated data can contribute to improved performance of the downstream semantic segmentation models without modifying the models themselves. Especially when the models are trained on a mix of real and synthetic data, performance improvements with respect to the baseline model, which is trained on real data only, can be achieved in several experiments. This could be demonstrated for binary as well as for multi-class segmentation. For some experiments, it was even possible to achieve improvements when training the models only with synthetic data.

Contents

Abstract i

Contents iii

List of Figures v

List of Tables ix

Declaration xi

1 Introduction 1
- 1.1 Image Synthesis for Data Enhancement . . . 2
- 1.2 Current Trends in Generative Modelling . . . 3
- 1.3 Objectives . . . 3

2 Semantic Segmentation 5
- 2.1 U-Net . . . 5
- 2.2 U-Net++ . . . 5
- 2.3 Feature Pyramid Network (FPN) . . . 7
- 2.4 DeepLabv3+ . . . 7
- 2.5 Datasets for Semantic Segmentation . . . 7

3 Diffusion Models 9
- 3.1 Mathematical Foundations . . . 9
- 3.2 Implementation . . . 12
- 3.3 Related Work . . . 12
  - 3.3.1 Improved Diffusion . . . 13
  - 3.3.2 Denoising Diffusion Implicit Models . . . 14
  - 3.3.3 Guided Diffusion . . . 14
  - 3.3.4 Classifier-free Guidance . . . 15

4 Generating Synthetic Semantic Segmentation Datasets 17
- 4.1 GAN-based Approaches . . . 17
  - 4.1.1 SPADE . . . 17
  - 4.1.2 OASIS . . . 18
  - 4.1.3 DatasetGAN . . . 18
- 4.2 Diffusion-based Approaches . . . 18
  - 4.2.1 Latent Diffusion Model (LDM) . . . 18
  - 4.2.2 Semantic Diffusion Model (SDM) . . . 18
  - 4.2.3 DiffuseExpand . . . 19
  - 4.2.4 Mask-conditioned Latent Diffusion . . . 19

5 Dataset and Preprocessing 21
- 5.1 Lemons Quality Control Dataset . . . 21
- 5.2 Binary Lemon Datasets . . . 22
- 5.3 Multi-Class Lemon Datasets . . . 25

6 Methods **27**
6.1 All-Latent: Paired Image-Mask Synthesis Using LDMs Only . . . 27
6.1.1 Implementation Details . . . 27
6.1.2 Experiments . . . 30
6.2 ISDM: Improved Semantic Diffusion Model . . . 31
6.2.1 Implementation Details . . . 31
6.2.2 Experiments . . . 33

7 Results **37**
7.1 Validation . . . 37
7.1.1 Evaluation Metrics . . . 37
7.1.2 Method . . . 38
7.2 All-Latent Experiments . . . 40
7.2.1 Binary Blemish Dataset . . . 40
7.2.2 Binary Mould Dataset . . . 44
7.2.3 Binary Mould-Blemish-Gangrene Dataset . . . 44
7.3 ISDM Experiments . . . 46
7.3.1 Binary Blemish Dataset . . . 46
7.3.2 Binary Mould Dataset . . . 47
7.3.3 Binary Illness Dataset . . . 48
7.3.4 Binary Blemish-Illness-Mould Dataset . . . 51
7.3.5 Multi-Class Blemish-Illness-Mould Dataset . . . 52
7.3.6 Ablation Experiments . . . 57

8 Conclusion **63**

Bibliography **67**

List of Figures

1.1 Example for Semantic Segmentation. Figure adapted from Zhou et al. (2018a). 1

1.2 Paradigm Shift for the Training Process of (Discriminative) Models Through the Use of Generative Models. Given a (discriminative) task and a real dataset and model for this task, the common approach (top) in deep learning considers the data as fixed and fits the model to it to arrive at a functional approximator f. Usually, when f performs poorly, it is attempted to find a better model, but the data remains untouched. In the contrary, with generative models (bottom) the data is no longer considered fixed, but it is possible to enhance the data itself such that it fits the given model to arrive at a possibly better approximator f^* without the need to change the model. 2

1.3 DreamStudio's output for the text prompt "A student writing his Master's thesis on diffusion models" 3

2.1 Different Model Architectures for Semantic Segmentation. 6

3.1 Example for the Reverse Diffusion Process. Starting from a random input image $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ the noise is gradually reduced to arrive at an image $\mathbf{x}_0$ from the underlying data distribution. The forward process follows the opposite procedure. 10

5.1 Histograms of the Pixel Frequency for the Categories of the Lemons Dataset. The "background" category comprises the real image background as well as the healthy tissue of the lemons and is therefore extremely predominant in terms of pixel frequency (left). To represent the distribution of the various defects on the lemons, the "background" pixels need to be ignored (right). 22

5.2 Examples from the Lemons Dataset. The images (left) are shown alongside the corresponding semantic masks (right). Each category of the dataset is mapped to a specific colour that can be looked up on the legends. The ticks on the axes show that the original image resolution is 1056×1056. 23

5.3 Example Images from the Binary Blemish (Sub-)Dataset. Images (top) and corresponding semantic masks (bottom) are shown. The latter are represented in the required black-and-white format for binary datasets. 24

6.1 LDM Approach. Figure taken from Rombach et al. (2022). 28

6.2 Schema of the All-Latent Approach. The optional filtering step is not shown. $\mathbf{x}$ represents a real image, $\mathbf{y}$ a real semantic mask. The variables $\mathbf{z}$ and $\mathbf{u}$ are used for the respective latent space encodings and $\mathbf{c}$ is the preprocessed conditioning information. Variables with subscript T indicate Gaussian noise and variables with apostrophes are (re)constructions from the backward diffusion process. All the real training data is used during the training process (top). The unconditional mask-generating LDM (top solid-line box) only requires the real semantic masks for training, the conditional image-generating LDM (bottom solid-line box) needs both, real masks and images. Sampling with the trained models does not require any real data: semantic masks are sampled from pure Gaussian noise and the generated masks are then used for conditioning during the image sampling process. The real test set is only used during the validation, which requires all real and synthetic data. 29

6.3 Architecture and Sampling Procedure for SDMs. Figures taken from Wang et al. (2022). . 32

6.4 Schema of the ISDM Approach. $\mathbf{x}$ represents a real image, $\mathbf{y}$ a real semantic mask. Variables with subscript T indicate Gaussian noise, and variables with apostrophes are (re)constructions from the backward diffusion process. All the real training data is used during the training process. The sampling process requires the real semantic masks from the training set. During sampling and also during finetuning (if a drop rate greater than 0.0 is chosen), both $\mathbf{y}$ and the null label are used for conditioning. The real test set is only used during the validation, which requires all real and synthetic data. 34

7.1 Examples for Samples from the Semantic Image Synthesis Approach for All-Latent. "Good" and "bad" pairs of images (top) and semantic masks (bottom), according to filtering, are shown. 43
7.2 Examples for Samples from the Paired Image-Mask Synthesis Approach for All-Latent. "Good" and "bad" pairs of images (top) and semantic masks (bottom), according to filtering, are shown. 43
7.3 Comparison between the Semantic Image Synthesis ("SIS") and the Paired Image-Mask Synthesis Approaches ("PIMS") on the Binary Blemish Dataset. For the comparison, the models trained with all real and different numbers of synthetic data ("#S") are used. The percentages refer to the improvements/deterioration compared to the baseline. 44
7.4 Comparisons between the Semantic Image Synthesis ("SIS") and the Paired Image-Mask Synthesis Approaches ("PIMS") on the Binary Mould-Blemish-Gangrene Dataset. For the comparison, the models trained with all real and different numbers of synthetic data ("#S") are used. The percentages refer to the improvements/deterioration compared to the baseline. 45
7.5 Comparison between the All-Latent approach for Semantic Image Synthesis and the ISDM Method on the Binary Blemish Dataset. The label below each pair of bars indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. 47
7.6 Example for the Overfitting of the ISDM Method on the Binary Blemish Dataset. Although only the semantic mask and not the real image itself serves as input for conditioning, the trained SDM generates images that look very similar to the original. 48
7.7 Example for the Overfitting of the ISDM Method on the Binary Mould Dataset. Although only the semantic mask and not the real image itself serves as input for conditioning, the trained SDM generates images that show hardly any changes in shape and size compared to the original. 49
7.8 Results for the ISDM Method on the Binary Illness Dataset. The label below each bar indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline. 50
7.9 Example for Three Differently Sampled Images. All three images were sampled from the same SDM, but at different points in training, images (a) and (b), or with and without the rescaling of diffusion steps, images (b) and (c). 51
7.10 Example of Three Identically Sampled Images of Different Perceived Quality. The same SDM and the same semantic mask were used for generating all three samples. The semantic mask belongs to the real image shown in (d). 51
7.11 Example for Diverse Samples Generated from the Binary Blemish-Illness-Mould Dataset. The two synthetic images on the right were sampled using the same semantic mask for conditioning. They show no signs of overfitting and the defective region induced by the semantic mask is clearly visible on both. 52
7.12 Results for the ISDM Method on the Binary Blemish-Illness-Mould Dataset for a Finetuned and a Non-Finetuned SDM. The label below each pair of bars indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. 53
7.13 Results for the ISDM Method on the Multi-Class Blemish-Illness-Mould Dataset. The label below each pair of bars indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline. 55

7.14 Comparison between the Multi-class and the Binary Mode for the Classes “Blemish”, “Illness” and “Mould”. The labels on the x-axis stand for the different tested configurations: 1. no real data, but as many synthetic image-mask pairs as there are in the real dataset, 2. no real data and 1000 synthetic pairs, 3. all real data and the same amount of synthetic data, 4. all real data and 100 synthetic pairs, 5. all real data and 400 synthetic pairs, 6. all real data and 700 synthetic pairs, 7. all real data and 1000 synthetic pairs. The percentages refer to the improvements/deterioration compared to the respective baseline. 56
7.15 Comparison of Different Guidance Scales on the Binary Blemish-Illness-Mould Dataset. The label below each bar indicates the number of real (“#R”) and synthetic (“#S”) image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline. 57
7.16 Comparison of Different Numbers for the Sampling Steps (T) on the Binary Blemish Dataset. The label below each bar indicates the number of real (“#R”) and synthetic (“#S”) image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline. 58
7.17 Comparison of Different Amounts of Synthetic Data on the Binary Blemish-Illness-Mould Dataset. The label below each bar indicates the number of real (“#R”) and synthetic (“#S”) image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline. 59
7.18 Comparison of Synthetic Data to (Traditional) Data Augmentation. The label below each bar indicates the number of real (“#R”) and synthetic (“#S”) image-mask pairs used to train the downstream segmentation model. The results correspond to the ones for the Binary Blemish Dataset, but are compared to a baseline trained with data augmentation. The baseline trained without data augmentation is also reported. The percentages refer to the improvements/deterioration compared to the augmented baseline. 60

List of Tables

7.1 Results of the All-Latent Approach for Semantic Image Synthesis on the Binary Blemish Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. "#Filtered" indicates how many of the 2000 originally generated image-masks pairs were selected. If "#Filtered" is NONE, for the values of "#S" below 2000, a random subset of the 2000 generated image-mask pairs is chosen. 40

7.2 Results of the All-Latent Approach for Paired Image-Mask Synthesis on the Binary Blemish Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. "#Filtered" indicates how many of the 2000 originally generated image-masks pairs were selected. If "#Filtered" is NONE, for the values of "#S" below 2000, a random subset of the 2000 generated image-mask pairs is chosen. 41

7.3 Results of the ISDM method on the Binary Blemish Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. For the values of "#S" below 1000, a random subset of the 1000 generated image-mask pairs is chosen. 46

7.4 Results of the ISDM Method on the Binary Mould Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. For the values of "#S" below 1000, a random subset of the 1000 generated image-mask pairs is chosen. 49

Declaration

By my own signature I declare that I produced this work as the sole author, working independently, and that I did not use any sources and aids other than those referenced in the text. All passages borrowed from external sources, verbatim or by content, are explicitly identified as such.

Signed:..... Date: 14/09/2023

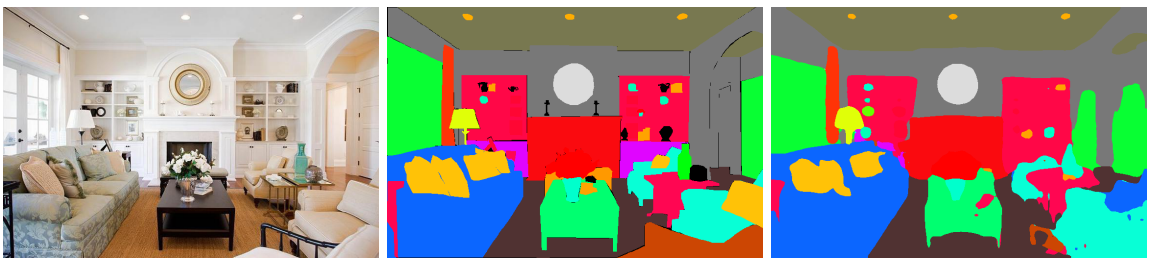
Chapter 1

Introduction

Semantic segmentation is a computer vision task in which each pixel of an image is categorized into a class. Accordingly, models addressing this task predict semantic masks, which assign a class value to each input pixel. Thus, the labels in semantic segmentation are typically grayscale images themselves, with each pixel containing the value of the corresponding class. Figure 1.1 demonstrates an example of the application of semantic segmentation for scene understanding. In particular, an image from the ADE20K (Zhou et al. 2017) dataset is shown together with the corresponding ground-truth label and a semantic mask predicted by a deep learning model. The two semantic masks in the figure are coloured for better visualisation. Each colour represents one of the classes from the dataset.

Besides scene understanding, semantic segmentation is also used in the field of visual anomaly/defect detection when it is required to detect anomalies with pixel accuracy. This application is particularly important for medicine (Khan et al. 2021) and industry (Qi et al. 2020). However, especially in medicine, there are major problems in creating and maintaining semantic segmentation datasets (Fernandez et al. 2022, Cechnicka et al. 2023, Shao et al. 2023, Macháček et al. 2023, Wu et al. 2023). This is because, as can be seen in Figure 1.1b, such datasets require meticulous labelling, which is time-consuming and costly because it often requires the help of domain experts. Thus, medical and industrial datasets are often scarce, which is a problem because, as pointed out by Wu et al. (2023), deep learning models for semantic segmentation tend to be data-hungry and therefore require sufficiently large datasets.

To address this issue, recent work has attempted to replace or extend datasets for semantic segmentation with synthetically generated datasets. Most of this work relies on generative adversarial networks (GANs) introduced by Goodfellow et al. (2014) to generate the synthetic data (Park et al. 2019, Sushko et al. 2022, Zhang et al. 2021). While this type of generative model works in general, researchers criticise image diversity (Wang et al. 2022), mismatched image-mask pairs (Wu et al. 2023) and training instabilities (Macháček et al. 2023). They propose approaches based on diffusion models (DMs) which, as shown by Dhariwal and Nichol (2021), have already outperformed GANs for image synthesis tasks. The goal of the approaches is to use the DMs to create enhanced datasets on which the data-hungry semantic segmentation models can be trained more effectively. Since most of these approaches are focused on the medical domain, this thesis expands on this line of research by applying similar approaches to industrial datasets.



(a) Image. (b) Ground-truth Semantic Mask. (c) Predicted Semantic Mask.

Figure 1.1: Example for Semantic Segmentation. Figure adapted from Zhou et al. (2018a).

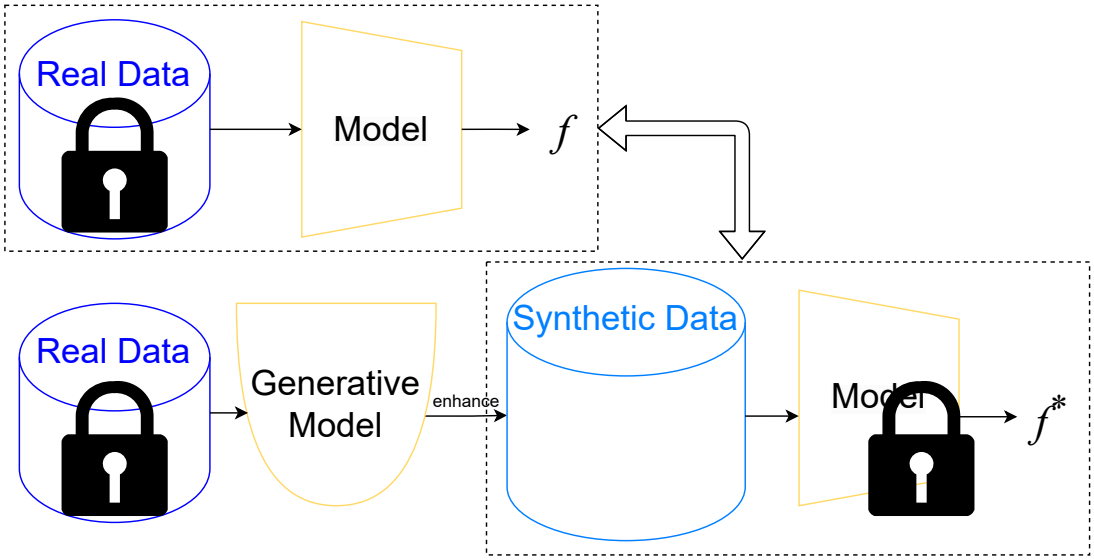


Figure 1.2: Paradigm Shift for the Training Process of (Discriminative) Models Through the Use of Generative Models. Given a (discriminative) task and a real dataset and model for this task, the common approach (top) in deep learning considers the data as fixed and fits the model to it to arrive at a functional approximator f. Usually, when f performs poorly, it is attempted to find a better model, but the data remains untouched. In the contrary, with generative models (bottom) the data is no longer considered fixed, but it is possible to enhance the data itself such that it fits the given model to arrive at a possibly better approximator f^* without the need to change the model.

1.1 Image Synthesis for Data Enhancement

Image synthesis, which is also referred to as image generation (both names are used interchangeably throughout this thesis), is a well-known task in the field of computer vision. As Croitoru et al. (2023) mentioned, the field essentially distinguishes between unconditional and conditional image synthesis.

Models for unconditional image generation are designed to sample images from pure random noise and thus do not require any input at inference time. These models are often benchmarked on large-scale image datasets in terms of the quality and diversity of the generated images. The most common metrics used in these benchmarks are the Inception Score (IS) proposed by Salimans et al. (2016), the Fréchet Inception Distance (FID) by Heusel et al. (2017) and the Learned Perceptual Image Patch Similarity (LPIPS) by Zhang et al. (2018). The IS, which was shown to correlate with human estimation, is used to measure the quality and diversity of generated images. Besides image quality, diversity is important as well, because the generative model should ideally capture the entire input distribution. Despite covering these two important properties the IS has the drawback that it cannot be used to evaluate the correspondence of the generated data with the input data. Therefore, FID was designed to consider both, generated and real images. It compares the distributions of the real and the generated data by leveraging the feature representations extracted by a neural network. The lower the FID, the better the two distributions correspond to each other. The approach behind the LPIPS metric differs from the IS and the FID, as it does not consider entire datasets of generated (and real) images, but it is used for human-like assessment of a pair of images.

Conditional image generation requires some kind of supervision signal, for example, a class label, to guide ("condition") the sampling process. This is particularly useful because it allows for a paradigm shift in computer vision that was also emphasised by Jahanian et al. (2021): traditionally, supervised learning has always been about considering the data as fixed and fitting models to that data to obtain a functional approximator f mapping from the input space to the desired output space. To this end, the computer vision community has put their focus on engineering better network architectures to improve the performance on a fixed dataset. However, due to the advances with models for conditional image synthesis, it is possible to control the generation of synthetic datasets in such a way that the datasets fit best for a given, fixed model. Therefore, in recent years there has been a shift in research towards

improving the data rather than designing complex network architectures. This paradigm shift is also illustrated in Figure 1.2. It assumes that there is a real dataset, albeit small and perhaps ill-suited for a model that can be trained on this real dataset to obtain the functional approximator f. According to the traditional approach, it was tried to change the model in order to arrive at an improved functional approximator f^*. The paradigm shift, however, foresees the use of a generative model, trained on the real dataset and used to enhance the data and create a more suitable synthetic dataset. This synthetic dataset is no longer fixed like its real counterpart since the generative model can basically generate an arbitrary amount of data and the data synthesis process can even be guided. Therefore, it is no longer necessary to change the given model to improve f, but the generative model can be used to fit the data to the given model to arrive at f^*.

1.2 Current Trends in Generative Modelling

OpenAI's ChatGPT is a recent and probably the best-known example of how generative artificial intelligence has already found its way into everyday life. Yet the language model behind the popular chatbot is by far not the first working generative model. Cao et al. (2023) date the beginnings of such models, which can quickly generate complex data and content, back to the 1950s. However, they also claim that the real breakthrough came with deep learning models. In computer vision, a breakthrough model for image synthesis is the GAN (Goodfellow et al. 2014). However, GANs are not the only choice for generating images, DMs, which were first introduced by Sohl-Dickstein et al. (2015), have proven their abilities for different image synthesis tasks as well. Rombach et al. (2022) argue that they feature some important advantages over GANs, such as a more stable training process which in addition is not prone to mode collapse. They also show that their Latent Diffusion Models (LDMs) can outperform GANs for some relevant image generation challenges. Additionally, as highlighted by Croitoru et al. (2023), DMs have recently attracted a lot of interest in academia and are also used in some important artificial imaging applications, such as Dall-E-2 (based on Ramesh et al. 2022) and DreamStudio (based on Rombach et al. 2022). Although these applications can create highly accurate images using only a textual description, as the example in Figure 1.3 demonstrates, when they were released there was no public response comparable to what ChatGPT has experienced. One of the reasons for this is certainly the usability of these different tools: while the ChatGPT chatbot can be a useful assistant for everyday life, the application area of Dall-E-2 and DreamStudio is mainly in the art domain and thus serves primarily for pleasure. Therefore, the aim of this master thesis is to show that DMs, such as those behind Dall-E-2 and DreamStudio, can also be used outside the field of art, for purposes that can improve applications in medicine and industry.

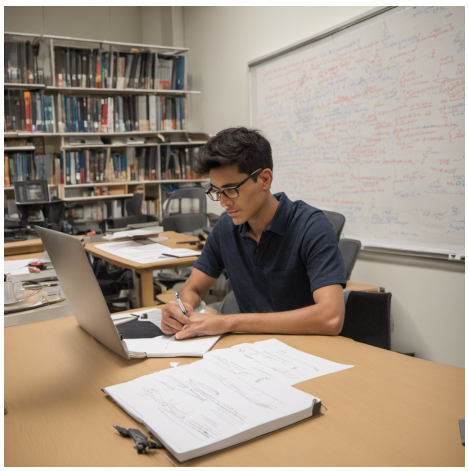


Figure 1.3: DreamStudio's output for the text prompt "A student writing his Master's thesis on diffusion models"

1.3 Objectives

Modern deep learning models for semantic segmentation require large amounts of training data to work optimally. However, corresponding datasets are often scarce and imbalanced, especially in industry, as labelling is time-consuming and tedious. Traditionally, attempts have been made to solve this problem through changes in model architecture and methodology. However, there are still few studies that attempt to change the datasets instead of the methodology. Therefore, this thesis addresses diffusion-based approaches for the generation of synthetic semantic segmentation datasets that are used to enhance the existing real datasets. Furthermore, the impact of the synthetic datasets is tested on different models for semantic segmentation through extensive experiments. In this process, the semantic segmentation models and their training and evaluation methods as well as the real data used for evaluation remain unchanged.

Only the dataset used for training the models is changed via DMs. The claim that this topic is of current scientific interest is strengthened by the fact that it was also discussed at the Computer Vision and Pattern Recognition Conference (CVPR) 2023, which is considered the most important conference in computer vision. In particular, the first workshop on Vision-based Industrial Inspection (VISION) hosted at CVPR 2023 issued a challenge with a very similar problem as the one addressed in this thesis. To summarise, the following contributions are proposed in this work:

1. Detailed review of DMs and methods to use them for the synthesis of semantic segmentation datasets.
2. Adaptation of two such diffusion-based methods for a small-scale industrial dataset.
3. Extensive experiments to investigate the applicability of the proposed methods for improving three different semantic segmentation models.

The ultimate goal is to understand if and how the different semantic segmentation models can be improved only by enhancing the training data with synthetic datasets generated by DMs. This goal aligns with the paradigm shift described in Section 1.1 and is pursued throughout this work. The thesis is structured as follows.

In Chapter 2 further aspects of semantic segmentation are highlighted. Chapter 3 discusses the mathematical foundations, practical implementations and important improvements of DMs. In Chapter 4 the use of the former for the tasks of paired image-mask and semantic image synthesis will be reviewed. The datasets used for this thesis and the required preprocessing are introduced in Chapter 5 and the proposed methods on this data in Chapter 6. Chapter 7 reports the results of the conducted experiments. Finally, the thesis is concluded in Chapter 8.

Chapter 2

Semantic Segmentation

The task of semantic segmentation was already mentioned in the introduction (Chapter 1) to this thesis. It is fundamental to highlight that the terminology for this task might sometimes appear confusing: "semantic (image) segmentation" is probably the most general name, but sometimes the names "image segmentation" or even only "segmentation" are used. If such models are used in the context of defect or anomaly detection, they may also be referred to as "defect segmentation" models. The literature is equally divided when it comes to the names for the labels: sometimes, instead of the term "semantic mask" used in this thesis, the terms "semantic map", "semantic layout" or "(semantic) segmentation map/mask" are used.

Despite the disagreement in the naming conventions, there are several well-known model architectures that have been designed for this task. The ones that are used in this work are introduced below. In addition, some well-known datasets for semantic segmentation are described, which typically serve to benchmark the models. Some of the standard metrics used for benchmarks, such as the mean intersection-over-union (mIoU), which corresponds to the ratio between the intersection and the union of the ground-truth and predicted semantic masks and the Dice score, which is computed similarly, are described in Section 7.1.2 about the validation methods used in this thesis.

2.1 U-Net

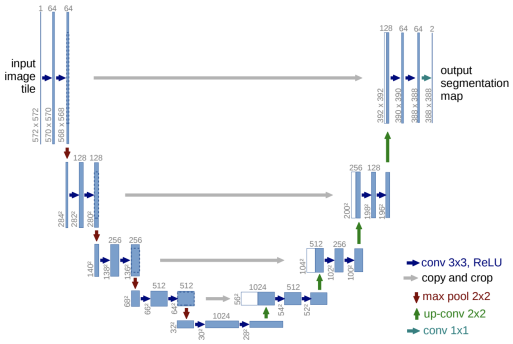
The U-Net is a widely used fully convolutional neural network (CNN) by Ronneberger et al. (2015) that was originally invented for biomedical image segmentation. The architecture of this model, which is shown in Figure 2.1a, features an encoder (downsampling) and a decoder (upsampling) part connected with skip connections. The decoder and the encoder are nearly symmetric, thus giving the model a U-shape.

Images passing through this model are first downsampled and encoded into latent representations, which are then upsampled and transformed into semantic masks by the decoder. The encoder consists of several blocks of 3×3 convolutional layers, ReLU activations and 2×2 max-pooling operations for downsampling. The decoder consists of similar components but uses upsampling instead of downsampling and each decoder block halves the number of feature maps, which were previously doubled in each encoder block, by means of deconvolution. The final layer is a 1×1 convolution that maps each vector in the last feature map of the decoder to the desired number of output classes.

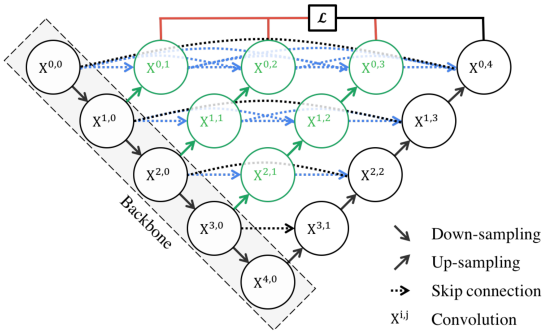
Another important aspect of the U-Net are the skip connections between the corresponding encoder and decoder blocks. They concatenate the respective feature maps and ensure that high-resolution features can be caught more precisely.

2.2 U-Net++

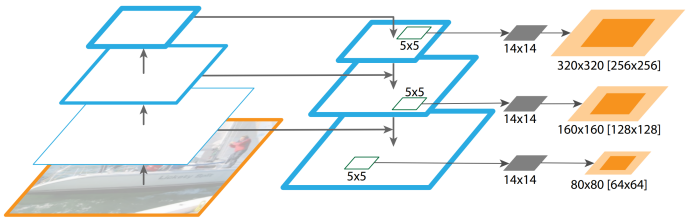
The U-Net++ by Zhou et al. (2018b) shown in Figure 2.1b uses the U-Net as a backbone and extends its skip mechanism: instead of directly concatenating the encoder and decoder feature maps, the U-Net++ introduces additional convolution blocks on the skip pathways. These transform the feature maps of the encoder to resemble the ones of the decoder, which should facilitate the work of the optimiser.



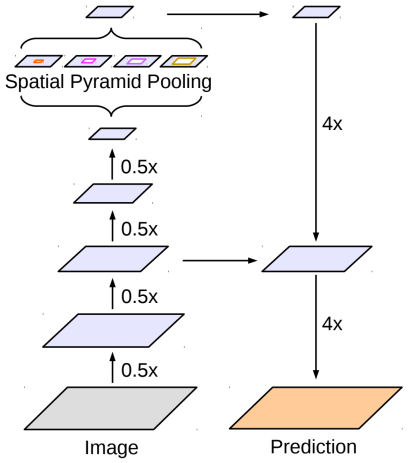
(a) U-Net Architecture. Figure taken from Ronneberger et al. (2015).



(b) U-Net++ Architecture. The parts in black correspond with the normal U-Net architecture, the parts in green and blue indicate the convolution blocks on the skip pathways and red represents deep supervision. Figure taken from Zhou et al. (2018b).



(c) FPN Architecture for Semantic Segmentation. An MLP is applied to 5×5 windows to generate semantic masks of size 14×14. Each level of the feature pyramid is used for prediction. This results in semantic masks of different sizes (light orange) for each level, which is intended for the recognition of objects of different sizes (dark orange). Figure taken from Lin et al. (2017).



(d) DeepLabv3+ Architecture. Figure taken from Chen et al. (2018)

Figure 2.1: Different Model Architectures for Semantic Segmentation.

Furthermore, through the extended skip pathways, the U-Net++ exhibits full-resolution feature maps along the pathways. Therefore, it allows for deep supervision, i.e. multiple loss heads that can be combined for more accurate results.

2.3 Feature Pyramid Network (FPN)

The FPN by Lin et al. (2017) does not follow a typical encoder-decoder architecture such as the U-Net, but it relies on a feature pyramid, a set of features that are extracted from different image resolutions. Such feature pyramids were much used before the era of CNNs when features were still hand-crafted. However, according to Lin et al. (2017) they are used at test time for modern approaches as well because they allow for the equal treatment of different resolution levels. Using image pyramids at training time is however infeasible regarding memory consumption. Nevertheless, the researchers came to the idea to use the feature maps of networks like the U-Net as multi-scale feature representation and to make independent predictions at all scales of the decoder path.

Originally, they developed the FPN for object detection, but they also showed that it is possible to use the network for semantic segmentation by using an additional multilayer perceptron at all resolutions. This procedure is depicted in Figure 2.1c.

2.4 DeepLabv3+

The DeepLabv3+ in Figure 2.1d is a model that was designed by Chen et al. (2018) with the aim of combining the benefits of two methods employed for semantic segmentation models: the first method corresponds to the use of encoder-decoder architectures, such as it is done for models like the U-Net. The second method is spatial pyramid pooling, which is an operation that analyses feature maps at different resolutions to encode multi-scale contexts.

The DeepLabv3+ builds upon the DeepLabv3, a model that employs spatial pyramid pooling, by adding a decoder that should ensure that object boundaries are detected more accurately. Both, the encoder of the DeepLabv3 and the proposed decoder are implemented using dilated convolutions. These generalise normal convolution operations as used for the U-Net by allowing to adjust the field-of-view.

2.5 Datasets for Semantic Segmentation

Some important datasets for semantic segmentation are Cityscapes (Cordts et al. 2016), ADE20K (Zhou et al. 2017), CelebAMask-HQ (Lee et al. 2020), COCO-Stuff (Caesar et al. 2018) and Pascal VOC 2012 (Everingham et al. 2012). In addition, there are some publicly available medical data sets, which are mostly aimed at segmenting anomalies. Unfortunately, there are not as many publicly available industrial datasets that target semantic segmentation for defect detection. This is a problem because synthetic data should have the greatest impact precisely for such datasets, which are usually significantly smaller than the ones mentioned above.

The cited datasets are large, publicly available datasets that do not concern anomaly detection, but the semantic segmentation of real-world scenes: Cityscapes consists of 25000 images from 50 German cities and targets the segmentation of urban street scenes. ADE20K is an even more extensive dataset, with over 27000 images where not only the objects but also parts of objects are labelled at a pixel-level. CelebAMask-HQ consists of 30000 images of human faces and the corresponding semantic masks featuring 19 different components of the human face. The COCO-Stuff dataset is a derivation from the well-known COCO dataset (Lin et al. 2014) with pictures of common objects in context. COCO-Stuff contains pixel-level labels of 172 classes for the entire COCO 2017 dataset. The Pascal VOC 2012 dataset consists of more than 9000 pictures of 20 different objects, including people, animals and vehicles, labelled at a pixel-level.

Chapter 3

Diffusion Models

Diffusion models (DMs) were introduced by Sohl-Dickstein et al. (2015) as a new type of generative deep learning model. DMs are inspired by non-equilibrium thermodynamics, which deals with the problem of irreversibility. They are designed to learn how to restore the structure in data, which they destroyed beforehand. Both, the forward diffusion process, which systematically adds noise to the input data, as well as the reverse (or backward) diffusion process, which learns to remove the added noise, operate as Markov chains which convert between two distributions in an iterative manner.

3.1 Mathematical Foundations

The algorithm of Sohl-Dickstein et al. (2015) which comprises the forward and backward processes is flexible in terms of data distributions that can be modelled, while it remains tractable to compute. Thus, it reduces the trade-off between flexibility and tractability of probabilistic models: highly flexible models that can fit arbitrary data distributions often work only in theory because they are not applicable in practice due to the high computational costs. The same is true for models that are easy to compute but do not feature the flexibility required for practical applications.

Assuming a data distribution $q(\mathbf{x}_0)$ the forward diffusion process with T steps is defined as:

$$q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3.1}$$

where $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the Markov diffusion kernel, which is typically chosen to be a Gaussian $\mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1}\sqrt{1-\beta_t}, \mathbf{I}\beta_t)$ with diffusion rates (variance schedule) $\beta_1, ..., \beta_T$. The reverse diffusion process, also referred to as the generative distribution, is defined analogously, but in reverse, starting from a standard normal distribution $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{3.2}$$

The assumption to start the reverse process from a randomly sampled, standard normally distributed $\mathbf{x}_T$ is only valid for a sufficiently large T, as the forward process needs to destroy the data completely within these T steps. Additionally, Sohl-Dickstein et al. (2015) pointed out that only if the forward process uses Gaussians in combination with small βs the reverse process has the same functional form as the forward process and can rely on Gaussians for the denoising process. This property is important because it reduces the learning procedure involved in the reverse process to a minimum: it is only required to learn the parameters (mean and variance) of a Gaussian distribution. More precisely the estimators for mean and variance that need to be trained for the reverse process can be defined as $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ respectively. Thus, the reverse diffusion kernel becomes $p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$. Figure 3.1 shows the conceptual idea behind the reverse (and forward) diffusion process.

Intuitively the training objective for the reverse process would be to maximize the probability $p(\mathbf{x}_0)$ which is assigned by the generative model p to the real input data $\mathbf{x}_0$. This is however intractable to compute, as it would involve an integration over all random variables $\mathbf{x}_1, ..., \mathbf{x}_T$. Therefore, applying

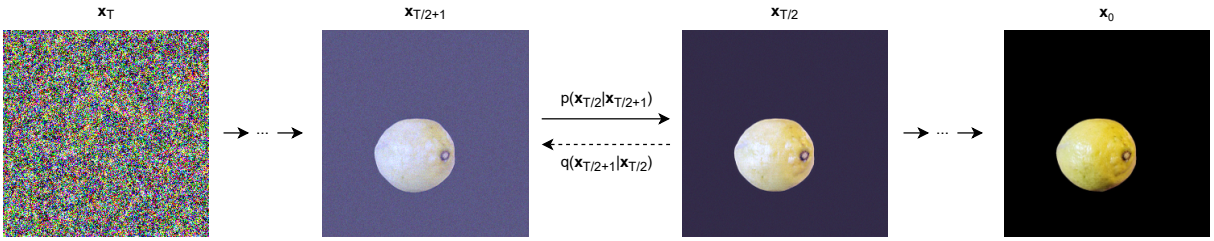


Figure 3.1: Example for the Reverse Diffusion Process. Starting from a random input image $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ the noise is gradually reduced to arrive at an image $\mathbf{x}_0$ from the underlying data distribution. The forward process follows the opposite procedure.

maximum likelihood optimisation directly is not feasible. However, it is possible to optimize the variational bound on the negative log-likelihood. This was already pointed out by Sohl-Dickstein et al. (2015) and is well-described in the work of Ho et al. (2020). In particular, they first defined the loss function L_{vlb} as the variational bound of the negative log-likelihood (NLL):

$$L_{vlb} := \underbrace{\mathbb{E}\left[-log\frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]}_{Variational\ Bound} \geq \underbrace{\mathbb{E}\left[-\log p(\mathbf{x}_0)\right]}_{NLL} \tag{3.3}$$

Then, they simplified L_{vlb} and rewrote it in terms of KL divergences:

$$\begin{aligned}
L_{vlb} &= \mathbb{E}\left[-\log\frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}\left[\log\frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p(\mathbf{x}_{0:T})}\right] \\
&= \mathbb{E}\left[\log\frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_T) * \prod_{t=1}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] && \text{(apply definition)} \\
&= \mathbb{E}\left[-\log p(\mathbf{x}_T) + \sum_{t=1}^{T}\log\frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] && \text{(apply log)} \\
&= \mathbb{E}\left[-\log p(\mathbf{x}_T) + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p(\mathbf{x}_0|\mathbf{x}_1)}\right] && \text{(restructure)} \\
&= \mathbb{E}\left[-\log p(\mathbf{x}_T) + \sum_{t=2}^{T}\log\left(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}\frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\right) + \log\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p(\mathbf{x}_0|\mathbf{x}_1)}\right] && \text{(extra-conditioning on } \mathbf{x}_0\text{)} \\
&= \mathbb{E}\left[-\log p(\mathbf{x}_T) + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \right. \\
&\qquad \left. + \log\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p(\mathbf{x}_0|\mathbf{x}_1)}\right] && \text{(apply log)} \\
&= \mathbb{E}\left[-\log p(\mathbf{x}_T) + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log q(\mathbf{x}_2|\mathbf{x}_0) + \sum_{t=2}^{T}\log q(\mathbf{x}_t|\mathbf{x}_0) - \right. \\
&\qquad \left. - \sum_{t=2}^{T}\log q(\mathbf{x}_{t-1}|\mathbf{x}_0) + \log\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p(\mathbf{x}_0|\mathbf{x}_1)}\right] && \text{(apply log)} \\
&= \mathbb{E}\left[-\log p(\mathbf{x}_T) + \log q(\mathbf{x}_T|\mathbf{x}_0) - \log q(\mathbf{x}_1|\mathbf{x}_0) + \log q(\mathbf{x}_1|\mathbf{x}_0) + \right. \\
&\qquad \left. + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p(\mathbf{x}_0|\mathbf{x}_1)\right] && \text{(restructure)} \\
&= \mathbb{E}\left[\log\frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p(\mathbf{x}_0|\mathbf{x}_1)\right]
\end{aligned}$$

$$= \mathbb{E}\left[\underbrace{D_{KL}\left(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)\right)}_{L_T} + \sum_{t=2}^{T}\underbrace{D_{KL}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)}_{L_{t-1}} \underbrace{-\log p(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\right] \tag{3.4}$$

The crucial step in this reformulation is the extra-conditioning on $\mathbf{x}_0$, which actually makes the final equation derivable and computable for known $\mathbf{x}_0$.

Additionally, Ho et al. (2020) pointed out that the forward process admits sampling $\mathbf{x}_t$ at an arbitrary time step t in closed form, which means it is not necessary to compute $\mathbf{x}_{t-1}$ before $\mathbf{x}_t$ can be computed. They introduced the auxiliary notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ to reparameterise the Gaussians and defined closed-form sampling for arbitrary time steps:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{3.5}$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \text{ with } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{3.6}$$

Due to this property, it is possible to not only sample $\mathbf{x}_0$ from the input data, but also to sample t from $[1, T]$ and apply a more fine-grained stochastic gradient descent training algorithm.

Furthermore, Equation 3.6 makes it easier to show that all L_{t-1} KL divergence terms of L_{vlb} have a closed-form solution because essentially they are comparisons between two Gaussians: $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is already known to be a Gaussian. $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ can be rewritten as a combination of three Gaussians using Bayes' rule: $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$, with:

1. $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right) = \frac{1}{\sqrt{2\pi\beta_t}} \exp\left(-\frac{1}{2}\frac{(x-\sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t}\right)$
2. $q(\mathbf{x}_{t-1}|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}\right) = \frac{1}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})}} \exp\left(-\frac{1}{2}\frac{(x-\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{(1-\bar{\alpha}_{t-1})}\right)$
3. $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) = \frac{1}{\sqrt{2\pi(1-\bar{\alpha}_t)}} \exp\left(-\frac{1}{2}\frac{(x-\sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{(1-\bar{\alpha}_t)}\right)$

The variance schedule $\tilde{\beta}_t$ of this new Gaussian can be obtained by just combining the first factors of each contributing Gaussian:

$$\frac{1}{\sqrt{2\pi\tilde{\beta}_t}} = \frac{1}{\sqrt{2\pi\beta_t}} \frac{\frac{1}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})}}}{\frac{1}{\sqrt{2\pi(1-\bar{\alpha}_t)}}} = \frac{1}{\sqrt{2\pi\beta_t}} \sqrt{\frac{(1-\bar{\alpha}_t)}{(1-\bar{\alpha}_{t-1})}} \Rightarrow \tilde{\beta}_t = \beta_t \frac{(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)} \tag{3.7}$$

Weng (2021) showed how to restructure the combined exponent of the three Gaussians to derive a formula for the mean $\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0)$ of the new Gaussian:

$$\begin{aligned}
&-\frac{1}{2}\left(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\right) = \\
&= -\frac{1}{2}\left(\frac{\mathbf{x}_t^2 - 2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1} + \alpha_t\mathbf{x}_{t-1}^2}{\beta_t} + \frac{\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_{t-1}\mathbf{x}_0 + \bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\right) = \\
&= -\frac{1}{2}\left(\frac{\mathbf{x}_{t-1}^2(\alpha_t(1-\bar{\alpha}_{t-1}) + \beta_t)}{\beta_t(1-\bar{\alpha}_{t-1})} - \frac{\mathbf{x}_{t-1}(2\sqrt{\alpha_t}\mathbf{x}_t(1-\bar{\alpha}_{t-1}) + 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\beta_t)}{\beta_t(1-\bar{\alpha}_{t-1})} + C(\mathbf{x}_t, \mathbf{x}_0)\right) = \\
&= -\frac{1}{2}\left(\frac{\mathbf{x}_{t-1}^2(1-\bar{\alpha}_t)}{\beta_t(1-\bar{\alpha}_{t-1})} - \frac{2\mathbf{x}_{t-1}(\sqrt{\alpha_t}\mathbf{x}_t(1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\beta_t)}{\beta_t(1-\bar{\alpha}_{t-1})} + C(\mathbf{x}_t, \mathbf{x}_0)\right) = \\
&= -\frac{1}{2}\left(\frac{\mathbf{x}_{t-1}^2}{\beta_t\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}} - \frac{2\mathbf{x}_{t-1}\frac{\sqrt{\alpha_t}\mathbf{x}_t(1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\beta_t}{1-\bar{\alpha}_t}}{\beta_t\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}} + C(\mathbf{x}_t, \mathbf{x}_0)\right) \\
&\Rightarrow \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}\mathbf{x}_t(1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\beta_t}{1-\bar{\alpha}_t}
\end{aligned} \tag{3.8}$$

where $C(\mathbf{x}_t, \mathbf{x}_0)$ is a function which does not depend on $\mathbf{x}_{t-1}$.

In summary, as for most supervised deep learning models, it is required to have a well-defined loss objective that is feasible to compute. DMs require a deep knowledge of probability theory to derive closed-form solutions such as Equation 3.4. However, the derivation alone is not enough to show that an objective is actually feasible to compute. Therefore, the Equations 3.7 and 3.8 show that the loss terms L_1 to L_{T-1} actually correspond to KL divergences between two Gaussians, which have a tractable closed-form solution.

3.2 Implementation

While the previous section focused on the mathematical foundations of DMs this section will shortly discuss the practical implementation of these mathematical results. In particular, the section will focus on some of the design choices by Ho et al. (2020), because their Denoising Diffusion Probabilistic Models (DDPMs) served as a baseline for many related approaches.

First and foremost it should be noted that the shown derivations only refer to the closed-form solutions for the loss terms L_1 to L_{T-1} of the L_{vlb} objective. The remaining loss terms L_T and L_0 were ignored until now because they are dependent on design choices. Ho et al. (2020) dealt with them in the following way: they decided to fix the variances as constants. Therefore, the variances were not learnable and hence the posterior $q(\mathbf{x}_T|\mathbf{x}_0)$ was not learnable as well, which made L_T a constant that could be ignored. To evaluate L_0 they even chose to model $p(\mathbf{x}_0|\mathbf{x}_1)$ independently by a discrete decoder model based on image color which made L_0 tractable as well.

Despite all these mathematical efforts to define L_{vlb} in closed-form, the researchers chose to reparameterise their neural network to predict the noise $\boldsymbol{\epsilon}$ from Equation 3.6 instead of the mean $\boldsymbol{\mu}$. They used the Equations 3.6 and 3.8 to rewrite $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ as:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) \tag{3.9}$$

This reparameterisation was used to define a simpler loss function based on the mean squared error of the estimated noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$:

$$L_{simple} := \mathbb{E}[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)||^2] \tag{3.10}$$

The authors conducted experiments with a U-Net-based architecture including group normalization and attention layers as well as parameters shared across time. They ablated their choices and found that an optimisation based on L_{simple} was both easier to implement and improved sampling quality as well (when predicting $\boldsymbol{\epsilon}$). Using this setup they were the first to generate high-quality image samples with DMs. Additional design choices for their DDPM include setting T to 1000 for training and sampling, using a linear schedule of constant βs (from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$) and an exponential moving average (EMA) with 0.9999 decay over the model parameters. The pseudo codes for their training and sampling algorithms are shown in Algorithm 1 and Algorithm 2 respectively. The fresh variable $\boldsymbol{\sigma}_t$ in Algorithm 2 represents the fixed standard deviations used by Ho et al. (2020), which is a simplification that will be discussed with more detail in the next section.

$\theta = initialise\_parameters()$;
while *not converged* **do**
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$; `/* sample data */`
  $t \sim unif(1, T)$; `/* sample time */`
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; `/* sample noise */`
  $\theta = GD\_step(\nabla_\theta ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)||^2)$
end

Algorithm 1: DDPM Training

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; `/* sample noise */`
for $i \leftarrow T$ **to** 1 **do**
  if $t > 1$ **then**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  else
    $\mathbf{z} = \mathbf{0}$
  end
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \boldsymbol{\sigma}_t \mathbf{z}$
end

Algorithm 2: DDPM Sampling

3.3 Related Work

In this section some related work that built upon DDPMs and marked further milestones in the development of DMs is reviewed.

3.3.1 Improved Diffusion

Nichol and Dhariwal (2021) based their work, which is often simply referred to as "Improved Diffusion", on the DDPMs by Ho et al. (2020) and aimed at improving the log-likelihood metric. They found the log-likelihood of DDPMs to be sub-optimal compared to other generative models and stressed on the importance of exploring this weakness as the log-likelihood is an important metric for generative models. A poor performance on this metric might indicate problems, such as mode collapse. They addressed this issue by identifying and proposing solutions for several problems of DDPMs:

- Increasing T: The first and easiest suggestion for better log-likelihoods was to increase T from 1000 to 4000, which comes however at the cost of slower sampling speeds.
- Learning $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$: Instead of learning the variance, Ho et al. (2020) chose to set it to $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \boldsymbol{\sigma}_t^2\mathbf{I}$, with fixed $\boldsymbol{\sigma}_t$. Nichol and Dhariwal (2021) found this to be a reasonable choice for sample quality, but not regarding log-likelihood. Therefore, they decided to learn the variance, which is however difficult due to training instabilities, as already pointed out by Ho et al. (2020). Thus, to keep the values for $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ in a small, acceptable range, they did not model it directly, but a vector was learned that was used to interpolate between β_t and $\tilde{\beta}_t$ (which were also the two extremes Ho et al. (2020) tested for $\boldsymbol{\sigma}_t^2$). Furthermore, they needed to adapt the loss function, as L_{simple} is not dependent on the variance. They defined a new hybrid loss as:

  $$L_{hybrid} := L_{simple} + \lambda L_{vlb} \tag{3.11}$$

  with $\lambda = 0.001$ as the optimisation of $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ should not influence too much the one of $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$.
- Improving the Noise Schedule: They showed that a linear noise schedule destroys structure too quickly and proposed a cosine-based schedule instead. It is designed to have small changes at the extremes of the diffusion process at $t = 0$ and $t = T$ and a nearly linear behaviour in the middle. As additional improvements, they used a small offset, because the model struggled with too small β_ts at the beginning, and they clipped β_t to a maximum of 0.999.
- Reducing Gradient Noise: Besides their hybrid loss, they wanted to optimise their models on L_{vlb} alone as well, which proved to be difficult. An analysis of the gradient noise showed that L_{vlb} was much noisier than L_{hybrid}. To overcome this issue they employed a dynamic importance sampling for t instead of sampling it uniformly in the range $[1, T]$. This modification in combination with L_{vlb} achieved the best log-likelihood, but it did not help to improve the log-likelihood when optimising L_{hybrid}.

Nichol and Dhariwal (2021) ablated their design decisions and claimed their hybrid loss and cosine schedule to be the best combination for improving the log-likelihood while still maintaining a FID score similar to the baseline by Ho et al. (2020). Furthermore, it was discovered that models with learned variance trained on L_{hybrid} and generate high-quality samples in terms of FID when using as few as 100 diffusion steps during sampling compared to the 4000 steps used for training. They could not reproduce these results with fixed-variance models trained using L_{simple}. This change in behaviour is of interest because, as specifically pointed out by Song et al. (2020), the long, computationally intensive sampling times are one of the major drawbacks of DMs. As the forward and the reverse diffusion processes are traditionally defined as counterparts with the same number of time steps T, they needed to rescale the time steps: An arbitrary subsequence S of the values $[1, ..., T]$ was used and the sampling noise schedule $\bar{\alpha}_{S_t}$ was defined using the given training noise schedule $\bar{\alpha}_t$. The sampling variances were simply rewritten as β_{S_t} and $\tilde{\beta}_{S_t}$ using $\bar{\alpha}_{S_t}$. Like this, the learned variance was automatically mapped to the range $[\beta_{S_t}, \tilde{\beta}_{S_t}]$, which is a rescaling for the shorter reverse diffusion process. To obtain S they selected equidistant values from the sequence $[1, ..., T]$. They conducted their experiments with a model based on the DDPM but changing the attention layers to multi-head attention which is applied at two different resolutions. Furthermore, they introduced changes for the time step embedding and set $T = 4000$ during training. The EMA procedure over model parameters of the DDPM implementation was preserved. They compared their Improved Diffusion implementation to the standard DDPMs and showed that it is possible to reach better log-likelihoods as well as a reduction in compute for sampling while keeping comparable FIDs. In addition, they also compared to the approach of Song et al. (2020), which is discussed in the next section and could outperform it with their L_{hybrid} models from 50 time steps onward.

3.3.2 Denoising Diffusion Implicit Models

Unlike the Improved Diffusion approach presented previously, Song et al. (2020) specifically targeted the problem of the long sampling times of DDPMs compared to GANs. They proposed Denoising Diffusion Implicit Models (DDIMs), which rely on the same training algorithm, but are more efficient during sampling. DDIMs generalize DDPMs by weakening the Markov property. This was achieved by introducing a T-dimensional vector $\boldsymbol{\sigma}$ to control the stochasticity of the forward process. Depending on $\boldsymbol{\sigma}$, $\mathbf{x}_t$ can depend on both $\mathbf{x}_{t-1}$ and $\mathbf{x}_0$, which breaks the Markov property. For $\boldsymbol{\sigma} = \mathbf{0}$ the forward process is deterministic for any given $\mathbf{x}_{t-1}$ and $\mathbf{x}_0$. It is also possible to define $\boldsymbol{\sigma}$ such that the DDIM becomes a DDPM. Despite this conceptual change, Song et al. (2020) could show that L_{simple} can still be used as loss objective if the model parameters are not shared across time, which, according to them, implied that they could reuse pretrained DDPMs and only needed to change the sampling procedure. Note that there is a discrepancy, because Ho et al. (2020) actually stated specifically that they shared their model parameters across time. Nevertheless, Song et al. (2020) managed to exploit pretrained DDPMs and DDIM sampling to generate samples that could compete with the ones of DDPMs in terms of quality, with a $10\times$ to $50\times$ faster sampling procedure. DDIM sampling essentially involves controlling the time step subsequence S and the variance hyperparameter $\boldsymbol{\sigma}$. To obtain S they either used the same technique as Nichol and Dhariwal (2021) (referred as linear selection) or they selected time steps at a linearly growing distance (quadratic selection), but always such that the last element is close to T.

3.3.3 Guided Diffusion

The papers by Nichol and Dhariwal (2021) and Song et al. (2020) mainly addressed problems of DDPMs for unconditional image synthesis and proposed several improvements for this task. Dhariwal and Nichol (2021) found that, despite these notable improvements, DMs were still behind GAN-based approaches and therefore they introduced further architectural changes as well as classifier guidance, a technique to trade off diversity for fidelity, meaning that the generated images become more similar to the real ones. Classifier guidance uses an extra classifier to condition the DM on a class label during sampling. In particular, the gradients of the classifier are included in the denoising process when sampling from the DM. Thus, by tuning a single parameter, the guidance scale s, Dhariwal and Nichol (2021) could trade off diversity for fidelity (the fidelity increases with s) of the generated images and set a new state-of-the-art for class-conditional image synthesis tasks. Furthermore, they showed that the technique could also be applied to unconditional DMs, for which it could improve the sample quality as well. However, they already reached a new state-of-the-art FID for such tasks with their architectural changes alone, which involved:

1. The use of multiple attention heads with 64 channels per head
2. Attention at several resolutions (32x32, 16x16, 8x8)
3. The use of BigGAN residual blocks for up- and downsampling
4. Adaptive Group Normalization (AdaGN) for embedding timestep and class into residual blocks

While they tested and justified their choices for 1.-3. in ablation studies included in the paper, AdaGN layers were already used in previous work (Nichol and Dhariwal 2021). They published the implementation for their follow-up work under the name "Guided Diffusion" as they considered their extension for class-conditioning their major contribution. Besides classifier guidance during sampling, AdaGN layers are the second method with which the researchers made use of class information. Essentially it uses the time step and class embeddings to scale and shift the output of a group normalization layer inside the residual blocks. While classifier guidance is only applied during sampling and can also be used with DMs that were trained without conditioning, AdaGN layers represent an architectural change to the model.

The inspiration for the classifier guidance scheme was taken from class-conditioned GANs as well as from previous work on DMs, for instance from Sohl-Dickstein et al. (2015), which had already shown that pretrained DMs can be conditioned with the gradients of a classifier. The main idea is to train a classifier $p_\phi(y|\mathbf{x}_t, t)$ on the noisy images $\mathbf{x}_t$ and use the gradients $\nabla_{\mathbf{x}_t} \log p_\phi(y|\mathbf{x}_t, t)$ to guide the sampling process towards the class label y. Dhariwal and Nichol (2021) derived equations for both, DDPM and DDIM conditional sampling procedures, which could both outperform state-of-the-art GAN-based approaches in terms of FID.

They tested s in the range $[0.0, 10.0]$. For their tests on ImageNet 256×256 setting $s = 10.0$ worked best for their unconditionally trained model and $s = 1.0$ for the one trained with conditioning on the class labels. Higher values for s result in higher fidelity, but as FID depends on fidelity and diversity it is necessary to find a reasonable trade-off for s that does not deteriorate one of them too much.

3.3.4 Classifier-free Guidance

Although Dhariwal and Nichol (2021) showed that their approach can reach state-of-the-art performance, the downside was that it required training an extra classifier. Therefore, inspired by classifier guidance, Ho and Salimans (2022) investigated if guidance could be performed without a separate classifier and designed an approach called Classifier-Free Diffusion Guidance. The idea behind this approach is to jointly train a class-conditional ($\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y)$) and an unconditional ($\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$) DM. They used a single class-conditional neural network to parameterise both these models and simply set the class label to a predefined null label ($y = \varnothing$) to simulate the unconditional case. During training, they randomly chose between $\varnothing$ and the true class label with probability p_{uncond}. The sampling process is a simple linear combination of $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y)$ and $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$:

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, y) = (1 + s)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - s\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \tag{3.12}$$

where s is the guidance scale that was already introduced by Dhariwal and Nichol (2021) with which Ho and Salimans (2022) managed to compete in their experiments. They tested s in the interval $[0.0, 4.0]$ and obtained their best FID results with a small amount of guidance ($s = 0.1$ and $s = 0.3$). Additionally, they experimented with p_{uncond} and found that during training the unconditional model needs only to be used occasionally $p_{uncond} \in \{0.1, 0.2\}$. Note that the above equation indicates that sampling with classifier-free guidance involves evaluating the model twice in each step, once with and once without conditioning. Thus, they showed that it is possible to achieve guidance without the need for an extra classifier, but just by DMs alone, which comes however at the cost of increased sampling time as evaluating a classifier is usually computationally cheaper than evaluating a DM twice.

Chapter 4

Generating Synthetic Semantic Segmentation Datasets

The focus of this thesis is on two approaches for creating synthetic semantic segmentation datasets. These are semantic image synthesis and paired image-mask synthesis. As pointed out by the papers of Sushko et al. (2022) and Wang et al. (2022), semantic image synthesis can best be understood as the opposite task of semantic segmentation. It targets the generation of realistic images that correspond with semantic masks provided as inputs. Semantic image synthesis is far less explored than its counterpart, especially when DMs are involved. In fact, Croitoru et al. (2023) categorised more than 100 papers on vision-related DMs, but only one of them, namely the work by Wang et al. (2022), belonged to this task.

While semantic image synthesis is only about generating synthetic images by leveraging information from real semantic masks, some work goes even further and aims at generating both images and corresponding semantic masks. The papers by Wu et al. (2023), Cechnicka et al. (2023), Shao et al. (2023) Macháček et al. (2023), Han et al. (2023), Fernandez et al. (2022) and Zhang et al. (2021) are examples for such work. These papers are mostly very recent such that there is not even a unique name for this task yet. Throughout this master thesis, this task will be referred to as "paired image-mask synthesis", which seemed to be the most self-explanatory name.

Both, semantic image synthesis and paired image mask synthesis, can address the issue of data-hungry semantic segmentation models by extending or replacing existing manually labelled datasets with synthetically generated data. In the following Section 4.1, some GAN-based approaches for these tasks will be reviewed, as they served as inspiration for diffusion-based approaches, which will be discussed in Section 4.2.

4.1 GAN-based Approaches

4.1.1 SPADE

Park et al. (2019) claimed that approaches for semantic image synthesis often worked sub-optimal, because of the use of standard normalisation layers, such as batch normalisation and group normalisation, which vanish a lot of the information contained in the semantic masks. Thus, they addressed this problem by equipping GANs with their proposed conditional normalisation technique called SPADE. SPADE is an acronym for "spatially-adaptive (de)normalisation" layers. These layers learn spatial modulation parameters (called scale and bias) which are computed from the semantic masks and combined elementwise with the normalised activations of the input image.

Park et al. (2019) showed that these layers can better preserve the semantic information in the mask and forward it through the network than previous methods for conditioning because they can benefit from normalisation, but without losing information.

Using SPADE layers they managed to quantitatively outperform the state-of-the-art models for semantic image synthesis in terms of mIoU and FID scores. Furthermore, they also conducted a user study where voters had to select which of the generated images fitted best to a given mask. The voters preferred their results on all the tested datasets.

4.1.2 OASIS

Sushko et al. (2022) built their OASIS model on top of the implementation by Park et al. (2019). They argued that the state-of-the-art GAN-based approaches for semantic image synthesis would show problems with training instability and image quality, because of the typical adversarial training process of GANs. Although these problems were usually diminished by making use of an additional perception network, Sushko et al. (2022) demanded for a computationally more lightweight solution. They proposed an approach in which they refused the additional perception network but changed the architecture of the discriminator to a U-Net-like semantic segmentation model. The feedback of this discriminator was no longer binary as for a conventional discriminator, but on a pixel level: for each pixel, the discriminator either decided on one of the real semantic classes of the dataset or on the "fake class", which was added in addition. They used the model by Park et al. (2019) as a baseline for their OASIS model, but improved on the architecture, especially on the redesigned discriminator, to gain a more efficient and simpler solution.

They showed that their OASIS model could outperform the state-of-the-art in terms of image quality (FID) on ADE20K, COCO-Stuff and Cityscapes datasets. Furthermore, they also stressed on the fact that class imbalance is a hard problem for semantic segmentation models, but has not yet been researched together with semantic image synthesis. Therefore, they extended their evaluation by making use of an imbalanced dataset. They showed that by assigning higher weights to pixels of underrepresented classes OASIS could help to drastically outperform the baseline model in terms of mIoU.

4.1.3 DatasetGAN

Zhang et al. (2021) addressed the issue of the demanding curation of semantic segmentation datasets with their DatasetGAN approach. They showed that it can reduce the required time for labelling to a minimum because it can synthesize large datasets from only a handful of manually annotated images. DatasetGAN relies on the expressive latent representations of pretrained, state-of-the-art GANs as it decodes the latents into semantic masks. Furthermore, Zhang et al. (2021) achieved this by training only a shallow decoder on top of the latent representation, which already generalized well when provided with only a few manual annotations. In particular, they used StyleGAN as their baseline. They sampled a few images from StyleGAN while also recording the corresponding latent codes, afterwards, they labelled these images manually and trained MLPs as "style interpreters" on the latent codes to approximate the labels. Additionally, filtering was applied during postprocessing to remove the most uncertain images.

The generated image-mask pairs were evaluated using the Deeplabv3 semantic segmentation model. However, they did not conduct in-domain experiments, meaning that they did not train and test on different splits of the same dataset, but they trained StyleGAN on a different dataset than the one they tested it on. The generated data (between 3000 and 10000 image-mask pairs) helped to outperform other techniques for improving semantic segmentation on scarce datasets, such as transfer learning.

4.2 Diffusion-based Approaches

4.2.1 Latent Diffusion Model (LDM)

The Latent Diffusion Model (LDM) by Rombach et al. (2022) marked a break-through for DMs as it achieved competitive if not state-of-the-art results for various computer vision tasks, such as image inpainting, class-conditional and unconditional image synthesis and text-to-image synthesis. In contrast to conventional DMs, such as DDPMs, which use the original image resolution for forward and reverse diffusion processes, LDMs operate in the latent space. Thus Rombach et al. (2022) could perform all the listed tasks by even reducing the computational costs compared to pixel-based DMs. LDMs are highly flexible regarding conditioning because the researchers integrated cross-attention layers into the latent space which can be exploited for conditioning on various different inputs such as text or semantic masks.

They tested their LDMs for various tasks with different conditioning formats, which is probably why semantic image synthesis was only addressed marginally. Nevertheless, they demonstrated that LDMs could also be used for semantic image synthesis on a landscape dataset.

4.2.2 Semantic Diffusion Model (SDM)

Wang et al. (2022) stated to be the first to use DMs for the task of semantic image synthesis and stressed

on the fact that all the other recent work on this task would mainly be GAN-based and come with the usual trade-off between high fidelity and diversity of the generated images. Therefore, they proposed to overcome these issues with DDPMs. In particular, they relied on the DDPMs from the Guided Diffusion implementation, but rather than focusing on single-class conditioning, they extended the conditioning mechanism such that they could condition the DDPMs on semantic masks. Thus, they named their models Semantic Diffusion Models (SDM). One of the major contributions of the SDM implementation is that it processes the noisy input image and the semantic mask independently. The noisy image serves as input for the encoder of the network, whereas the semantic mask, which is used to condition the diffusion process, is only fed into the decoder. Furthermore, the mask is forwarded through the spatially-adaptive normalization (SPADE) layers introduced by Park et al. (2019). SPADE layers help to generate images with high semantic relevance. In addition, they applied classifier-free guidance during sampling to further enhance the correspondence between the real semantic masks that were used for conditioning and the generated images. Specifically, during sampling they conditioned the DM with both, the real semantic mask and the null label. The null label for this case is a semantic mask of zero values only, which corresponds to no conditioning. The final output of the inference process, namely the estimated noise, was computed as a linear combination of the conditional and unconditional noise, hyper-parameterized by the guidance scale (s). Furthermore, Wang et al. (2022) implemented an optional procedure to finetune the DMs by randomly replacing the mask by the null label during the training process.

They evaluated their approach on four different datasets (Cityscapes, ADE20K, CelebAMask-HQ, Coco-Stuff) and computed the FID metric for image quality and the LPIPS metric for image diversity. Additionally, they tested the usefulness of the generated samples for semantic segmentation with different standard models for this task on the basis of the mIoU. For some datasets, their generated data was able to outperform real data in terms of mIoU. All these metrics were compared with recent GAN-based approaches. They managed to outperform them in terms of fidelity and diversity and for some datasets (CelebAMask-HQ and Cityscapes) also for the downstream task of semantic segmentation. Furthermore, a paired user study between their SDMs and some of the GAN-based approaches was conducted and won by the SDM approach.

4.2.3 DiffuseExpand

Shao et al. (2023) wanted to address the problems for semantic segmentation models that arise due to the scarcity of data in the medical domain. They proposed DiffuseExpand, as a paired image-mask synthesis method based on DMs. DiffuseExpand uses DMs to first generate semantic masks and then to sample corresponding images by conditioning on these masks. Furthermore, for refinement, not all of the generated image-mask pairs are considered, but only high-quality samples are selected by means of an additional neural network. Actually, DiffuseExpand was proposed as a four-phase algorithm: In the first phase they fine-tuned a pre-trained DM for the mask generation. For the second phase a segmentation model was trained for the purpose of classifier guidance. These first two phases were needed for the conditional image synthesis in phase three. For the final phase a neural network was used which, given a generated image-mask pair, estimated whether to keep it or not.

Shao et al. (2023) used two different X-ray datasets to evaluate DiffuseExpand. They tested the different phases of their algorithm and compared the performance to related approaches. The IS was used to measure the quality of the generated images and the FID for the diversity. Furthermore, they evaluated if their approach could improve the performance of a U-Net by means of the Dice score. The experiments showed that it was worth running all four phases of the algorithm, especially for the Dice score. The comparison with other approaches revealed that DiffuseExpand could outperform them in terms of the Dice score, but not the FID.

4.2.4 Mask-conditioned Latent Diffusion

Macháček et al. (2023) drew attention to a special task within the medical domain, namely the segmentation of gastrointestinal (GI) polyp images. The labelled data for this task is rare, as collecting it is time-consuming and costly and requires taking into account privacy issues and expert assessment. Therefore they proposed an approach for paired image-mask synthesis on a binary semantic segmentation dataset for GI polyp images, which could help to build a system for computer-aided diagnosis. The approach uses the Improved Diffusion implementation as well as conditional LDMs: the Improved Diffusion

model is used to generate semantic masks of polyps from pure random noise. The LDM is conditioned on the generated masks to synthesise corresponding images.

The authors tested their approach in terms of image quality, using the FID. However, they did not compare their results with others but only tested their own DMs across different epochs and iterations. Moreover, they evaluated the effect of their generated data by means of three different semantic segmentation models: the UNet++, the FPN and the DeepLabv3+. For all three models, they estimated different metrics on the HyperKvasir dataset, which was not used for training. The results showed that the generated data could improve the performance of segmentation models, but the amount of improvement depended on the model architecture, which is why they claimed that it is necessary to test approaches for semantic image and paired image-mask synthesis with different semantic segmentation models.

Chapter 5

Dataset and Preprocessing

Some well-known datasets for semantic segmentation are mentioned in Chapter 2. However, most of these datasets consist of a large number of different real-world pictures and do not aim at anomaly detection, but at understanding the objects in the depicted scene. For these reasons, the datasets differ significantly from medical datasets and industrial datasets.

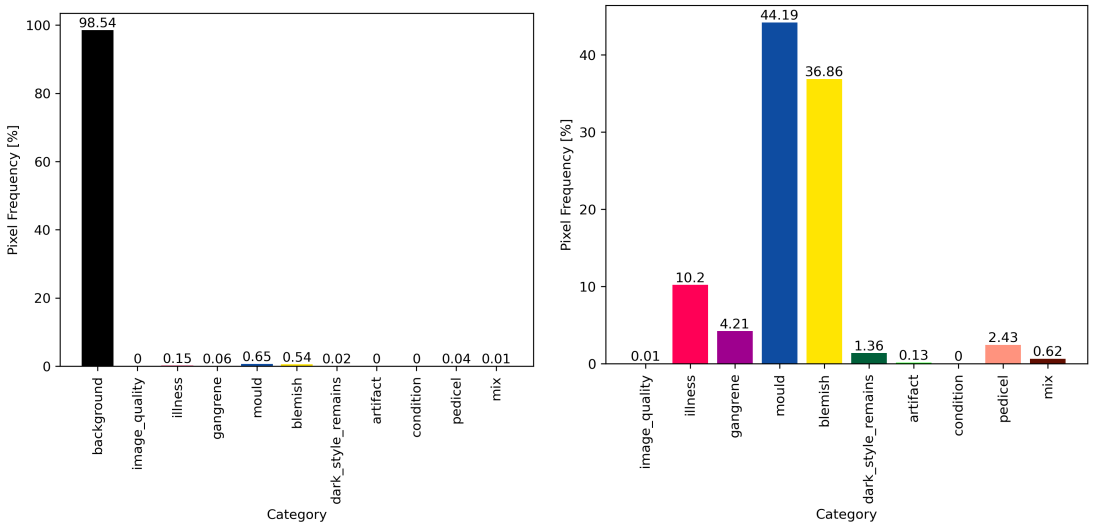
Datasets from these two domains are usually much smaller than the reviewed benchmark datasets. Furthermore, they usually consist of images that are taken under constant conditions and show only a single or a small number of objects, which should be analysed for anomalies. Many approaches for semantic image synthesis and paired image-mask synthesis address the medical domain mainly because labelling medical images is even harder as it requires domain expertise from highly educated professionals such as physicians (Fernandez et al. 2022, Cechnicka et al. 2023, Shao et al. 2023, Macháček et al. 2023, Wu et al. 2023). Moreover, synthetic data is even more important for the medical domain as privacy issues might prohibit the use of real data. Another reason why the literature on the application of these approaches to industrial datasets is scarce is that most industrial datasets are not published.

One of the best-known public datasets for industrial inspection is the MVTec anomaly detection dataset from Bergmann et al. (2021), which is however designed for unsupervised anomaly detection and contains only anomaly-free images in the training set. Therefore, it is not suitable for training conditional DMs, as the conditioning can not be trained. Thus, it was necessary to use less explored datasets, which is also acceptable, because basically, the approaches should work for any supervised, vision-based anomaly detection dataset with a sufficient amount of data to train the involved DMs.

For all the datasets presented in this chapter, it is expected that the datasets are split into a training and a test set to ensure that a proper real subset is kept apart for the final validation. It makes sense to enforce training and test sets since after using the approaches presented in this paper it is necessary to validate if the generated data actually improves the downstream semantic segmentation models or not.

5.1 Lemons Quality Control Dataset

The Lemons Quality Control Dataset by Adamiak (2020) is one of the few publicly available datasets addressing semantic segmentation for anomaly detection in the industrial domain. As the name of the dataset, which will be abbreviated as Lemons Dataset, suggests, it consists of 2690 images of lemons with defects labelled on a pixel level for the purpose of fruit quality control. The dataset features nine different defect categories, seven of which concern actual product quality issues. The two other categories, “image_quality” and “artifact”, describe problems with the images themselves, but these are very rare defects. The healthy tissue of the lemons is not labelled, therefore the background of the image and the healthy parts share a single label (with category ID 0) and will simply be referred to as the “background” category. Figure 5.1 represents the distribution of the different defect categories with and without considering the “background” category. Figure 5.2 shows some example images and their corresponding ground-truth labels (semantic masks). These figures demonstrate that most of the pixels in the dataset belong to the “background” category. Only approximately 1.5 % of the pixels are defective. Also, there is a considerable discrepancy between the defect categories themselves: the two most frequent defects “mould” and “blemish” comprise more than 80 % of all defective pixels. Other anomalies such as “image_quality”, “artifact”, “condition” and “mix” have relative frequencies below 1 %. Only for the



(a) Distribution with “Background”.

(b) Distribution without “Background”.

Figure 5.1: Histograms of the Pixel Frequency for the Categories of the Lemons Dataset. The “background” category comprises the real image background as well as the healthy tissue of the lemons and is therefore extremely predominant in terms of pixel frequency (left). To represent the distribution of the various defects on the lemons, the “background” pixels need to be ignored (right).

“mix” category the low frequency compared to other categories is actually favourable, because “mix” indicates two or more overlapping defects, which is a scenario with which semantic segmentation models normally can not deal, because they are designed to predict one class per pixel. Datasets with imbalanced and underrepresented categories generally decrease the performance of semantic segmentation models, which is why the following sections will discuss several subsets of the Lemons Dataset, that were examined during the experimental part of this thesis.

5.2 Binary Lemon Datasets

One possible way to model the data distribution of a semantic segmentation dataset with DMs is to train a separate DM for each class of the dataset. In this case, it is required to split the dataset according to the classes on the semantic masks such that the splits become binary datasets featuring only a single defect category and the “background”. Within the semantic masks the “background” should be encoded with the class value 0 and the defect class with 255, thus black and white images are obtained for the semantic masks, on which the defects are easily recognisable.

The limitation of this approach is that, due to the scarcity of some classes of the Lemons Dataset, it is impossible to train a DM for these classes, because there are too few images remaining for the DM to learn the data distribution. Therefore, the focus will be on the most frequent classes and on the combination of such. The binary datasets that are used for the experiments are discussed shortly in the following:

- Binary Blemish: “Blemish” is a collective class for several issues on lemons. In total, there are 200 images in the dataset, which are only featuring this defect class. Some examples of images and corresponding masks are shown in Figure 5.3.
- Binary Mould: Although “mould” is actually the most common defect according to the number of pixels, the Binary Mould Dataset is very small because mould rarely occurs as the only defect on lemons. Thus, the dataset consists of only 39 images. Nevertheless, this dataset is used for some experiments, on the one hand, to test the limits of the DMs, and on the other hand because it is well suited for debugging, since “mould” defects can be recognised easily.
- Binary Blemish-Illness-Mould: To overcome the issue of too few available images for single classes,

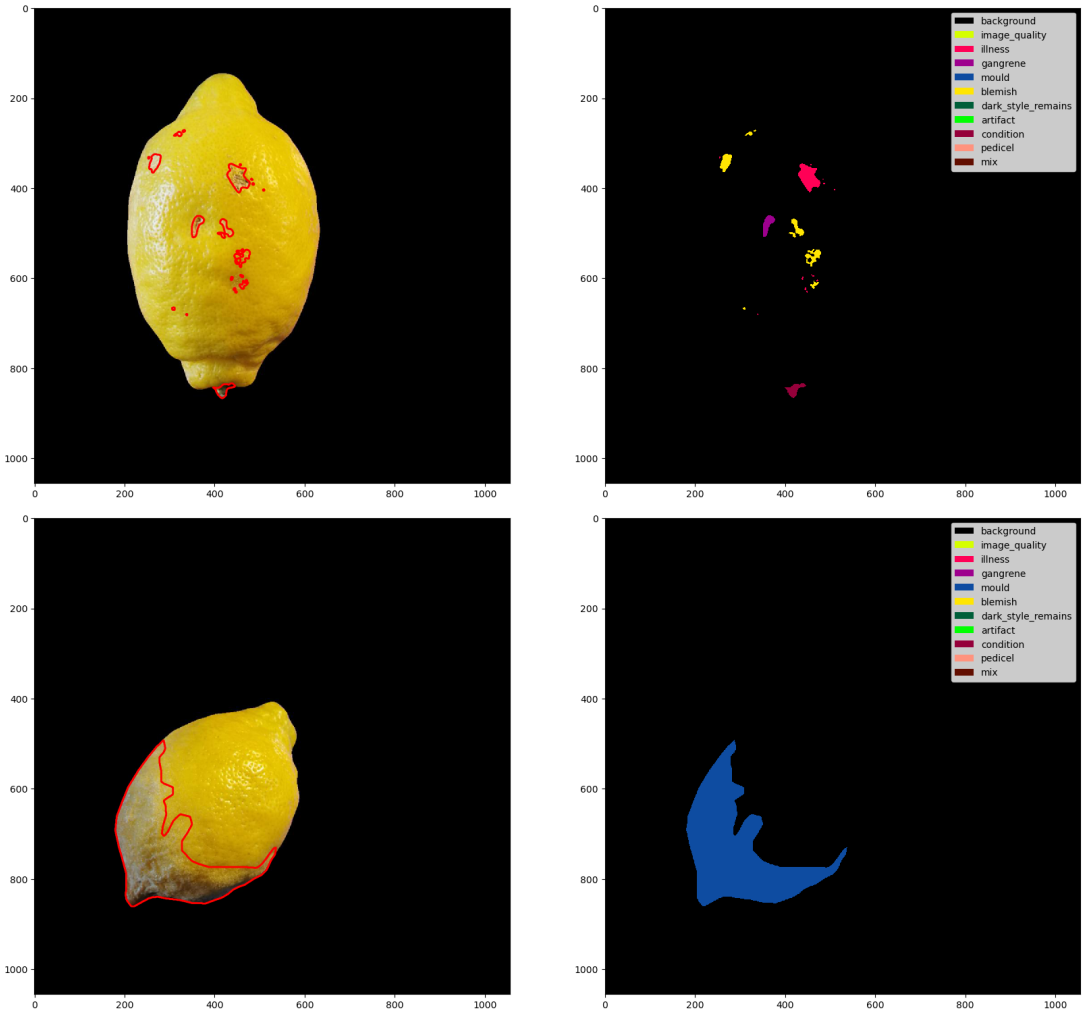


Figure 5.2: Examples from the Lemons Dataset. The images (left) are shown alongside the corresponding semantic masks (right). Each category of the dataset is mapped to a specific colour that can be looked up on the legends. The ticks on the axes show that the original image resolution is 1056×1056.

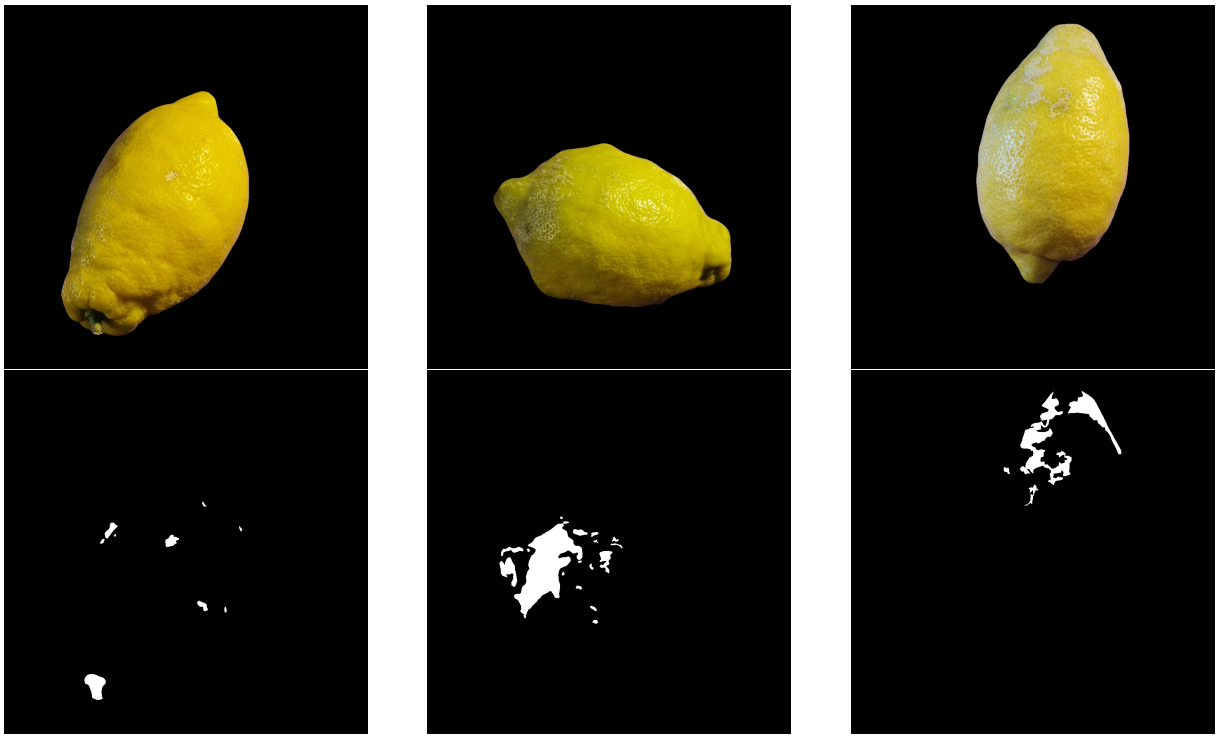


Figure 5.3: Example Images from the Binary Blemish (Sub-)Dataset. Images (top) and corresponding semantic masks (bottom) are shown. The latter are represented in the required black-and-white format for binary datasets.

the idea was to take the three most representative classes of the dataset ("blemish", "illness" and "mould") and to treat them as a single class. The resulting dataset consists of 790 images. Obviously, it is not possible to control which of the three classes the DMs will generate on the lemons during sampling, but tests with this dataset help to understand the behaviour of the DMs for the binary case when it is provided with much more images than there are available for any single class.

- Binary Mould-Blemish-Gangrene: This sub-dataset was created following the idea of the Binary Blemish-Illness-Mould Dataset, except that not the three most frequent classes were chosen, but the three classes that are the easiest to recognise based on visual inspection. Another benefit, besides the well-recognisable defects, is that this dataset (315 image-mask pairs) is only slightly bigger than the Binary Blemish Dataset, so ideally, training times and results on this dataset should be comparable.

- Binary Illness: This sub-dataset was created in order to have a separate dataset for the last class of the Binary Blemish-Illness-Mould Dataset. This was mainly done, to have an additional source for result comparison. The dataset consists of 106 images and masks and thus lies between the Binary Mould and Binary Blemish Datasets in terms of size.

Although, as described above, the entire Lemons Dataset could be modelled using the binary subsets, the usage of this technique would be rather impractical for most real-world applications. This is because it would mean training a DM for each individual class, which is not only time-consuming but not even possible for classes with little data. Nevertheless, it has advantages to consider binary subsets, because the DMs are much faster to train and debug due to the reduced size of the datasets. Additionally, having only a single defect class prevents the risk of confusion between classes.

In order to have a more practical example, a multi-class split of the Lemons Dataset is also considered. It is presented in the following section.

5.3 Multi-Class Lemon Datasets

Instead of treating each class separately, the entire Lemons Dataset with all its classes can be used to train DMs as well. However, since the class imbalance for the Lemons Dataset is quite high, it is appropriate to consider only a subset of the classes and thus simplify the problem. As with the binary datasets, the focus will lie on datasets consisting of the most frequent classes. For the multi-class case, the categories need to be numbered in ascending order without gaps starting from the class value 0 for the "background". The resulting pictures for the semantic masks can therefore appear completely black, since grayscale values in the range $[1, 10]$, which is the maximum range for the defect classes when using the entire Lemons Dataset, can only be distinguished from black when looking closely.

The only multi-class subset created for the Lemons Dataset is referred to as "Multi-Class Blemish-Illness-Mould Dataset". It is composed of the three most frequent defect classes. Actually, this dataset even consists of five classes, because the "background" class and in this case the "mix" class must be included, as there are small overlaps between the individual defects. However, the mix class should have little influence on the models due to its low frequency. It is only important to know about it because the models to be trained must be configured for five and not four classes. An alternative would be to remove the few images in which "mix" occurs and train the models with four classes. If no images are removed, the total number of image-mask pairs for this dataset is 790, the same as for the Binary Blemish-Illness-Mould Dataset.

Chapter 6

Methods

The methods proposed in this chapter aim to explore the capability of recent DMs for paired image-mask and semantic image synthesis. The main goal is not to generate images that look as realistic as possible, but to find an approach to improve semantic segmentation models for vision-based anomaly detection with the generated data.

6.1 All-Latent: Paired Image-Mask Synthesis Using LDMs Only

Inspired by the approach by Macháček et al. (2023), who implemented paired image-mask synthesis based only on DMs for a medical binary segmentation dataset, the first choice was to adapt their work to the binary lemon datasets. However, instead of using two different types of DMs, the decision was to rely only on Latent Diffusion models, which is why the approach is called "All-Latent". In contrast, Macháček et al. (2023) used an unconditional Improved Diffusion model to synthesise semantic masks and a LDM to generate images using the semantic masks for conditioning. The decision to use LDMs for both the unconditional mask generation and the conditional image generation is based on the fact that LDMs are indirectly a further development of Improved Diffusion models, making it unnecessary to use both implementations.

6.1.1 Implementation Details

The LDM approach, illustrated in Figure 6.1, is characterised by the application of the forward and reverse diffusion processes in the latent space. To map images into the latent space and (after the reverse diffusion process) back into the image space, the encoders and decoders of pre-trained autoencoder models are used. In the case of All-Latent, a pre-trained VQ-F4 autoencoder model was used, which is also listed in the official Latent Diffusion repository by Rombach et al. (2022). This autoencoder was used for the image-generating LDM as well as for the LDM for generating semantic masks. For the former, the use of the conditioning mechanism of LDMs is required to condition on semantic masks. As indicated in Figure 6.1, this mechanism is designed for supervision signals of different formats, such as text and other images, but also the semantic masks, which are used for the proposed method. In order to work with these different supervision signals, it requires a domain-specific encoder (τ_μ) that maps the conditioning information into the latent space. Since the original paper on LDMs tested the approach for semantic image synthesis, the LDM implementation already has such a domain-specific encoder for semantic masks, which, like all components of the LDM implementation, can be set via a configuration file. Thus, the configuration used for the conditional LDM in All-latent, used a domain-specific encoder called "trainable spatial rescaler" that consists of bi-linear interpolation with a convolution layer on top to downscale and transform the semantic mask for the conditioning process. The so-called "learned-conditioning" is performed by concatenating the encoded input image and the rescaled semantic mask. Alternatively, if the configurations for the conditioning process are chosen differently, the LDM approach would also allow using a non-trainable rescaler, such as pure bi-linear interpolation, as well as cross-attention instead of concatenation for the conditioning process. The decision was not to change these parameters, but to use the provided default configuration for semantic image synthesis with LDMs.

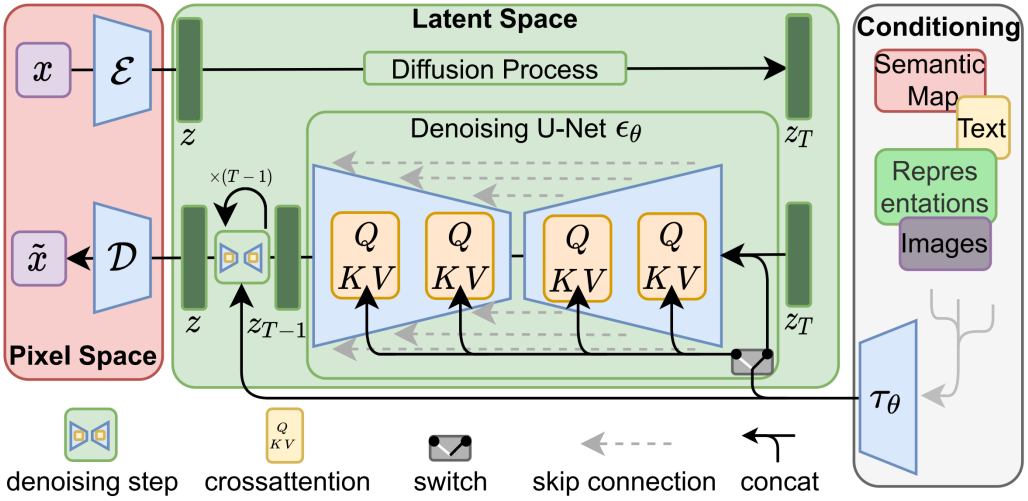


Figure 6.1: LDM Approach. Figure taken from Rombach et al. (2022).

The configuration used for the mask-generating LDM is almost identical, only the conditioning was disabled. Thus, both LDMs used the mentioned autoencoder, the U-Net architecture by Dhariwal and Nichol (2021) for the reverse diffusion, 1000 diffusion steps during training, a loss function similar to L_{simple}, but using the L1-norm instead of the L2-norm and a linear variance schedule. Note that this configuration was actually meant for RGB images, therefore the semantic masks needed to be transformed from grayscale to RGB, which is for sure sub-optimal, but Rombach et al. (2022) did not provide a configuration file nor a pretrained autoencoder for grayscale images. Nevertheless, it is possible to use the two LDMs as described above for paired image-mask synthesis.

The proposed All-Latent approach, which is also depicted in Figure 6.2, comprises the following steps to fulfil this task:

1. Train an unconditional LDM on the real semantic masks.
2. Train a conditional LDM on the real images conditioned on the real masks. This step can be done in parallel to the first step.
3. Sample masks from the first LDM.
4. Sample images from the second LDM while using the generated masks from the third step for conditioning.
5. Filter the best image-mask pairs based on the mIoUs or the cross-entropy loss of a U-Net.
6. Validate the generated data

The approach following the steps above allows for sampling an arbitrary amount of independent image-mask pairs because it is designed such that images and masks are created from pure random noise: the images are created from random noise and conditional semantic masks, which are created from random noise alone.

The idea of filtering the generated data during post-processing is nothing new. For example, Shao et al. (2023) even chose to design a separate neural network just for filtering in order to deal with the issue that DMs sometimes generate bad data. Since the goal of this thesis is to improve semantic segmentation models, the idea was to use such a model for filtering as well. Thus, filtering consists of training a lightweight semantic segmentation model, namely the U-Net model proposed by Ronneberger et al. (2015) and implemented by Iakubovskii (2019) on the real data and evaluating it individually on each image-mask pair of the synthetic data. The top-K image-mask pairs are selected either on their mIoU (higher is better) or cross-entropy loss (lower is better). These metrics are computed by comparing the predictions of the U-Net on the generated images with the generated masks, which are considered ground-truth for this case.

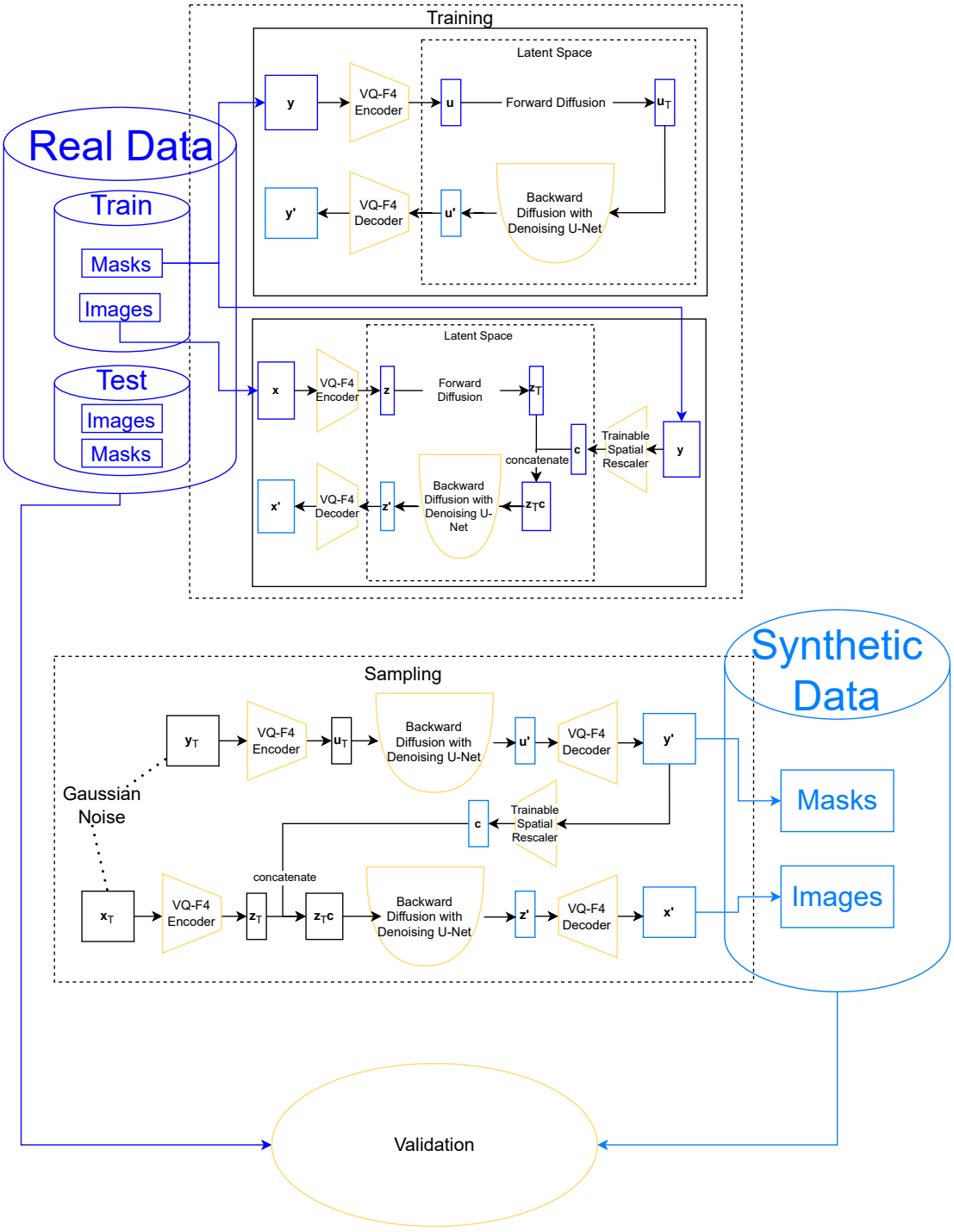


Figure 6.2: Schema of the All-Latent Approach. The optional filtering step is not shown. $\mathbf{x}$ represents a real image, $\mathbf{y}$ a real semantic mask. The variables $\mathbf{z}$ and $\mathbf{u}$ are used for the respective latent space encodings and $\mathbf{c}$ is the preprocessed conditioning information. Variables with subscript T indicate Gaussian noise and variables with apostrophes are (re)constructions from the backward diffusion process. All the real training data is used during the training process (top). The unconditional mask-generating LDM (top solid-line box) only requires the real semantic masks for training, the conditional image-generating LDM (bottom solid-line box) needs both, real masks and images. Sampling with the trained models does not require any real data: semantic masks are sampled from pure Gaussian noise and the generated masks are then used for conditioning during the image sampling process. The real test set is only used during the validation, which requires all real and synthetic data.

In addition to the approach for paired image-mask synthesis described above the same procedure was implemented as a semantic image synthesis approach as well. In this case, it is only required to train the conditional image-generating LDM. During the sampling process the real semantic masks of the train set are either used repeatedly for conditioning or data augmentation is used to avoid reusing identical masks. The augmentations that are applied are random affine transformations (rotation, translation, scaling) as well as random horizontal and vertical flips. These transformations do not change the classes of the defects on the semantic masks.

The last step, after generating (and filtering) images and masks is to evaluate their impact on semantic segmentation models. Since the validation method itself requires training and evaluating semantic segmentation models, it is described in the next chapter. Whereas, the experiments conducted with All-Latent are described in the following section.

6.1.2 Experiments

The usefulness of All-Latent for improving semantic segmentation models was evaluated on the Binary Blemish Dataset for three different test scenarios:

1. The impact of different amounts of generated data.
2. The impact of replacing the mask-generating LDM with a semantic image synthesis approach.
3. The impact of the filtering mechanism.

The two LDMs for mask and image generation were configured as described in the sections above and trained for 1000 epochs each, on an NVIDIA Tesla V100 GPU. The best models were obtained at epochs 800 and 650 respectively, which was estimated by monitoring a validation set corresponding to a 20 % split from the real training set. The trained models were used to sample 2000 image-mask pairs with the DDIM sampling procedure using 250 time steps.

After this single run through the All-Latent procedure, which might take several days depending on the available computing resources, the conditional LDM was reused to sample another 2000 images to test the second scenario. In this case the semantic masks, however, were not sampled from the LDM, but by following a semantic image synthesis approach with the real, but augmented masks. In addition to the comparison of both approaches to a common baseline, which made no use of any synthetic data, the two approaches for All-Latent were also validated against each other. Furthermore, postprocessing was applied to the two datasets generated from the two approaches to create filtered versions for testing the last scenario.

Besides the Binary Blemish Dataset, the first and second scenarios were also tested on the smaller Binary Mould Dataset and the slightly larger Binary Mould-Blemish-Gangrene Dataset. All-Latent was not tested on multi-class versions of the Lemons Dataset, as this would have required several adjustments in the implementation, which was not worthwhile, as had been discovered in the experiments.

6.2 ISDM: Improved Semantic Diffusion Model

The second method to be tested for the problem posed is based on the implementation of SDMs by Wang et al. (2022). This implementation represents a pure semantic image synthesis approach which was adopted largely unchanged in this work. Two small changes to the training and sampling strategy resulted in some improvements for the tested datasets, which is why the proposed method is referred to as Improved SDM (ISDM). This term is not meant to be comparative or even degrading with respect to the original approach but serves primarily to differentiate between them. The term ISDM is used for the overall method presented below, while the term SDM is still used to describe the model architecture, which remained unchanged.

6.2.1 Implementation Details

The semantic image synthesis approach proposed by Wang et al. (2022) features some differences compared to the one attempted with All-Latent. These differences mainly concern the conditioning strategy:

- There are no explicit encoders that transform images and semantic masks into a latent space before conditioning is performed.
- The semantic mask used for conditioning is not directly concatenated with the input image. Instead, the information is brought together only in the decoder layers of the SDM by means of spatially adaptive (de-)normalisation (SPADE). For this special conditioning process, the network architecture of the Guided Diffusion model, which serves as a basis, was changed by Wang et al. (2022).
- Finetuning is performed by resuming the training process including a random drop of the conditioning information. This is done by using a semantic mask of zero values only (null/empty label) instead of the real semantic mask.
- Classifier-free guidance is used for the sampling process, which allows for more precise control of the conditioning. In particular, classifier-free guidance computes the output of the model by repeating each sampling step twice, once conditioned on the real semantic mask and once conditioned on the null label. The two predictions are then combined into a single one using the guidance scale. Such a sampling step is depicted in Figure 6.3b.

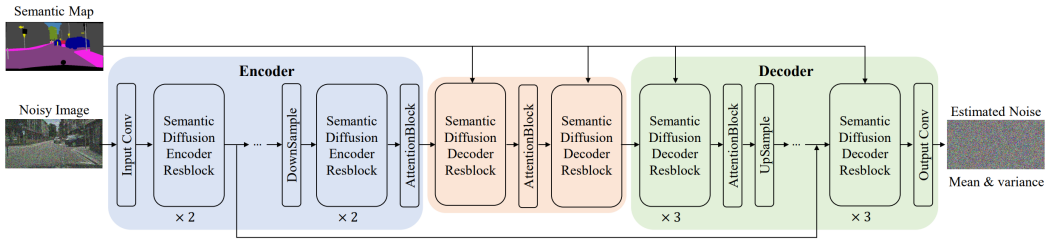
Some of these changes are also depicted in Figure 6.3, which demonstrates the model architecture (6.3a) and the sampling procedure (6.3b) for the SDM. Another difference between the approaches is that for ISDM, the real semantic masks are simply reused, while the proposed semantic image synthesis approach for All-Latent allows to apply data augmentation to the semantic masks.

Wang et al. (2022) tested their approach on several of the semantic segmentation datasets for scene understanding presented in Section 2 and achieved state-of-the-art results for the fidelity (FID) and diversity (LPIPS) of the generated images. Furthermore, they achieved state-of-the-art mIoU scores on two of these datasets.

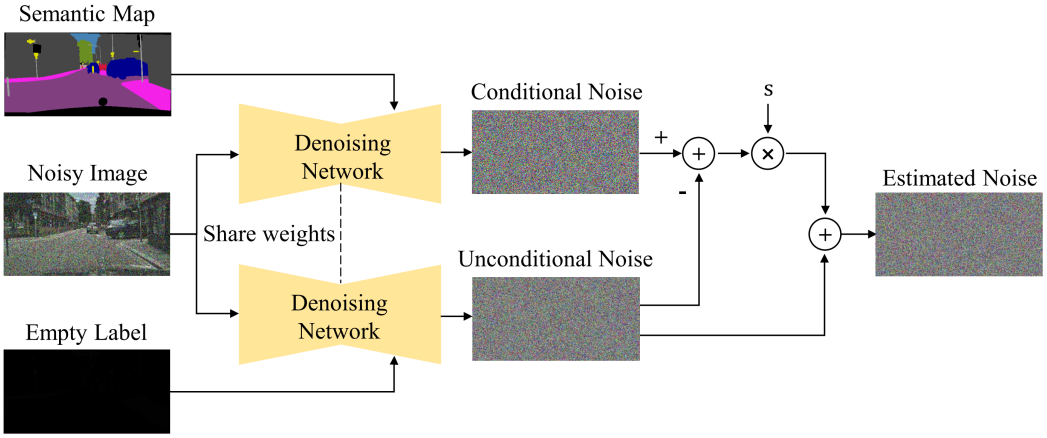
Therefore, the decision was to mainly stay with the default configurations for the SDM. The default configuration for training includes the use of $T = 1000$ diffusion steps, a linear variance schedule and the prediction of the mean as well as the standard deviation of the random noise, which is why L_{hybrid} is used as a loss function. For finetuning the same configuration was used, with an additional drop rate parameter, which indicates the probability with which the null label is used instead of the real semantic mask. For sampling a guidance scale of $s = 1.5$ was applied.

Besides the use of these default configurations, the proposed ISDM approach essentially made two adjustments to the implementation of Wang et al. (2022). One for the training process and a smaller one for the sampling process. These changes did not lead to better results but increased usability and efficiency on the tested datasets.

One shortcoming of the original implementation was that it did not use the notion of epochs nor any methods to monitor the training process, which are traditional design patterns in deep learning. Instead, only the optimiser steps, i.e. the number of batches processed, are counted. Every 10000 steps, without regard to any metrics, a model checkpoint is stored that occupies several GB of disk space. In the experiments, this did not prove to be a suitable strategy to train reproducible models for the Lemons



(a) SDM Architecture.



(b) SDM Sampling Step Including Classifier-free Guidance.

Figure 6.3: Architecture and Sampling Procedure for SDMs. Figures taken from Wang et al. (2022).

(Sub-)Datasets, since the number of steps required to train reasonable models depended significantly on the size of the data set and on the batch size. Also, this method proved to be extremely wasteful in terms of disk space, as the first saved checkpoints were mostly unusable and also because the implementation did not use a maximum number of steps, but kept training until it was manually aborted. To avoid these problems, the implementation was changed so that epochs are tracked in addition to the optimiser steps and the training process is automatically terminated after a maximum number of epochs (20000).

Additionally, it was attempted to implement a monitoring criterion based on a validation split of the training set and to save checkpoints only when the validation loss improved, but this had a negative effect on the performance of the SDMs. This was probably due to the fact that the datasets used for the experiments were already small and the 20 % validation split was therefore simply too small. Thus, instead of using a separate validation set, the training loss per epoch was used as a monitoring criterion. Although this cannot avoid possible overfitting, which is also discussed in the experiments in Chapter 7, consistently usable models could be trained that delivered reasonable results even with steps below the 30000 mark, which was not the case before with arbitrary checkpoints after any 10000 steps.

Furthermore, with the ISDM method a checkpoint was only saved when the monitoring criterion (training loss) for an entire epoch improved. The previous checkpoint was deleted, when a new checkpoint was saved, which had an advantageous effect on memory utilisation. In addition, this reduced the time needed to select the best model after training to a minimum. However, for larger datasets and to avoid overfitting it might be beneficial to use validation set monitoring and keep more than one checkpoint.

The second change to the original SDM implementation brought significant advantages in terms of runtime as well. One of the disadvantages of the SDM implementation was that for sampling, as for training, 1000 diffusion steps were used as default. This led to long sampling times and high resource consumption . However, it was noted that the implementation, by being based on Guided Diffusion, would actually allow the use of some techniques to improve sampling speed. These include DDIM sampling and timestep rescaling in combination with the use of L_{hybrid}. These two techniques were introduced by Song et al. (2020) and Nichol and Dhariwal (2021) respectively, who showed that they could be used to drastically reduce the required number of diffusion steps during sampling. For the experiments conducted

in this work, the former technique did not work well, but this had already been noted in an issue on the official SDM GitHub repository. The latter technique, on the other hand, proved to be effective.

These two changes allowed effective training of SDMs and subsequent generation of synthetic datasets using the ISDM method shown in Figure 6.4 which has a similar data and control flow to the All-Latent method. However, the ISDM method was implemented as a pure semantic image synthesis approach, which is why no DM is used to generate semantic masks. Therefore, the real masks are also reused for the synthetic dataset. Since the SDM models use classifier-free guidance during sampling, each sampling step consists of two parts: for the first the real semantic mask ($\mathbf{y}$) is used for conditioning, for the second the null label ($\varnothing$) is used. The outputs of both procedures are combined under the control of the guidance scale (s) into one output, which finally ends up in the dataset as the image corresponding to $\mathbf{y}$. It should be noted that the illustration is a simplified representation, because backward diffusion does not consist of a single model prediction, but is an iterative process. So, the Denoising U-Net in Figure 6.4 actually represents the entire process. Similarly, classifier-free guidance is not only done once at the end of the backward diffusion process as shown in the figure, but in every single step. In reality, it does not directly predict and combine images from $\mathbf{y}$ and $\varnothing$, but actually, it predicts and combines the noise on the images that should be removed in each step, as it is depicted in Figure 6.3b.

6.2.2 Experiments

Like the All-Latent method, the ISDM method was tested on different sub-sets of the Lemons Dataset. In contrast to All-Latent, the ISDM method can also be used on multi-class datasets without major changes. Therefore tests on a multi-class dataset were included as well. The various experiments that were conducted are briefly described below. They are grouped according to the datasets on which they were performed.

Binary Blemish Dataset

The experiments on the Binary Blemish Dataset were mainly used to analyse the general impact of the data generated by SDMs for semantic segmentation, as well as for comparison with the results of All-Latent. For this purpose, an SDM model was trained on the dataset for three days on an NVIDIA Tesla V100 GPU. The best checkpoint was reached after 67480 optimisation steps, which demonstrated the increased efficiency of the modified checkpoint procedure because, without it, most models below 70000 to 80000 optimisation steps were unusable. No further fine-tuning was carried out for this SDM. The model was used to sample 1000 images.

In addition to the general impact of the synthetic data for semantic segmentation models and the comparison to All-Latent, a discussion regarding some observed issues for the ISDM method was added to the results for this model in Section 7.3.1.

Binary Mould Dataset

The experiments on the Binary Mould Dataset were actually conducted to understand whether, in contrast to the All-Latent method, it is possible to generate useful results from SDMs trained on such a small dataset. In fact, it turned out that this was only possible by using the ISDM method with the proposed, modified checkpointing. Neither with the SDM implementation alone nor with checkpoints based on the validation loss, was it possible to train SDMs that were able to remove enough noise and sample recognisable lemons. This indicates that a training set of about 30 elements, such as the Binary Mould Dataset, seems to be the lower limit with which the SDMs can be trained to produce more than just random noise. The SDM was trained for three days on an NVIDIA Tesla V100 GPU and the best checkpoint was found after 101895 steps. No further finetuning was performed. The model was used to sample 1000 images.

The results for the experiments on this dataset were reported in Section 7.3.2. They consist of a discussion about the observed overfitting and the analysis of the impact of the generated data for semantic segmentation.

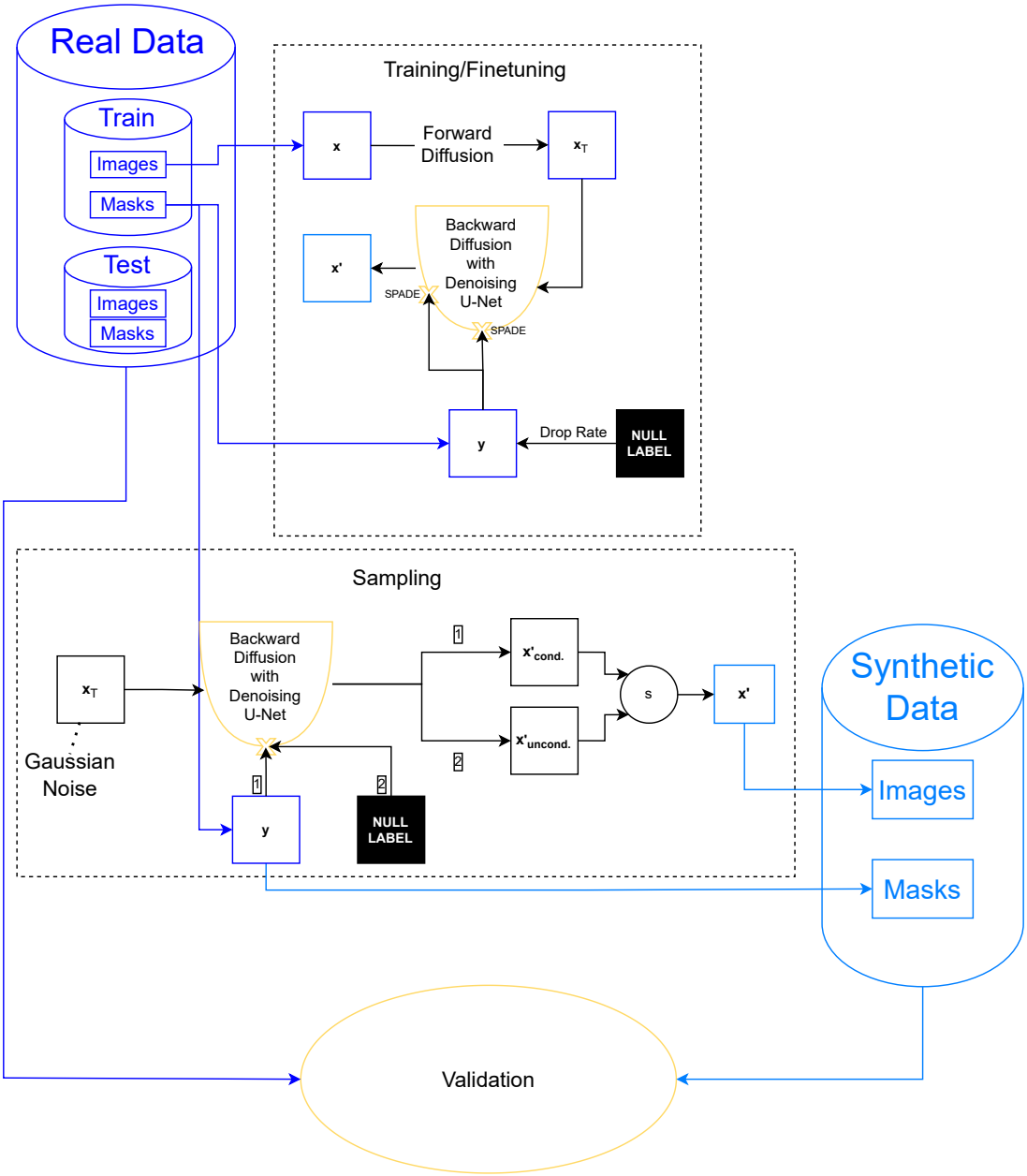


Figure 6.4: Schema of the ISDM Approach. $\mathbf{x}$ represents a real image, $\mathbf{y}$ a real semantic mask. Variables with subscript T indicate Gaussian noise, and variables with apostrophes are (re)constructions from the backward diffusion process. All the real training data is used during the training process. The sampling process requires the real semantic masks from the training set. During sampling and also during finetuning (if a drop rate greater than 0.0 is chosen), both $\mathbf{y}$ and the null label are used for conditioning. The real test set is only used during the validation, which requires all real and synthetic data.

Binary Illness Dataset

The SDM for the Binary Illness Dataset was trained twice (resumed after the first training) for two days on an NVIDIA Tesla V100 GPU, as the sampled images still appeared noisy after the first training. The best checkpoint overall was at 53664 steps. No further fine-tuning was performed, but 1000 images were sampled directly with this model.

The experiments on this dataset should serve as a confirmation of the results for the first two datasets. Furthermore, the observed changes in the appearance of the sampled images during training, as well as other observed anomalies, were discussed with the help of examples.

Binary Blemish-Illness-Mould Dataset

The Binary Blemish-Illness-Mould Dataset is larger compared to the other two binary datasets above. The aim of the experiments was to see how the size of the dataset affects the ISDM method. For this purpose, the SDM was trained on the dataset for three days, whereby the best checkpoint was found after 59124 optimisation steps. In addition, the model was subsequently finetuned for two days, resulting in a new best checkpoint after 72436 steps (training and finetuning steps summed up). Both SDMs (finetuned and non-finetuned) were used to generate two synthetic datasets of 1000 image-mask pairs each.

Thus, in addition to the usual discussion, the results for this dataset also include a comparison between the results for the finetuned and the non-finetuned SDM.

Multi-Class Blemish-Illness-Mould Dataset

The Multi-Class Blemish-Illness-Mould Dataset, which consists of the three most frequent classes in the Lemons Dataset, is the only multi-class dataset on which the ISDM method was tested. The reason for this was not difficulties in switching from binary to multi-class, as was the case for the All-Latent method. In fact, the SDM implementation allowed for a relatively easy switch from binary to multi-class mode, only the number of input classes needed to be set accordingly. However, the focus of the experiments was initially on obtaining consistent results for the easier binary problems before switching to multi-class.

The main goal for the experiments on the Multi-Class Blemish-Illness-Mould Dataset was to see if the results obtained for the binary datasets could also be applied to the multi-class case. The multi-class SDM was trained for three days and reached the best checkpoint after 24440 optimiser steps. Due to time constraints, no further fine-tuning was carried out. The SDM was used to sample 1000 images.

The discussion of the results in Section 7.3.5, describes the general impact of the synthetic data for semantic segmentation and compares it to the results for the respective binary datasets.

Chapter 7

Results

The goal of this thesis is to investigate whether semantic segmentation models for anomaly detection tasks can be improved with synthetic data from DMs. Thus, the validation process itself requires to train semantic segmentation models by using the generated data. In this chapter, first, the general validation process will be explained and then the results of the various experiments that were conducted throughout this thesis will be presented.

7.1 Validation

This section first discusses some well-known validation metrics for semantic segmentation. Then the exact validation method is described in more detail.

7.1.1 Evaluation Metrics

The validation of semantic segmentation approaches is non-trivial. It is not possible to simply compare the ground-truth class with the predicted class as in a classification task, because the output is not a single class per input image, but one class per pixel. In order to do that, a comparison must be carried out pixel by pixel, but the evaluation of a single pixel does not give necessary information about objects. Moreover, metrics such as accuracy, i.e. the sum of correctly assigned pixels divided by the sum of all pixels, are often misleading, because for some tasks, especially for defect segmentation, the background class makes up a large majority of all pixels. Thus, even an algorithm that classifies everything as background achieves a high accuracy. For this reason, other metrics than accuracy are proposed for semantic segmentation. Müller et al. (2022) mentioned some of them in their guideline for evaluation metrics in medical image segmentation. They reported the following names, definitions and formulas, which consider binary semantic segmentation problems:

- Accuracy (Acc), also known as pixel accuracy, is simply the number of correctly classified pixels divided by the number of all pixels.

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \tag{7.1}$$

- Recall, also known as sensitivity or true positive rate:

$$Recall = \frac{TP}{TP + FN} \tag{7.2}$$

- Precision, also known as specificity or true negative rate:

$$Precision = \frac{TN}{TN + FP} \tag{7.3}$$

- Intersection over Union (IoU), also known as Jaccard Index, or Jaccard Similarity Coefficient, is a widely used metric for semantic segmentation. It calculates the ratio between the intersection and the union of the ground-truth and predicted semantic masks.

$$IoU = \frac{TP}{TP + FP + FN} \tag{7.4}$$

- Dice Similarity Coefficient (DSC), also known as Dice score, F1-score or Sørensen-Dice index, is computed similarly to the IoU and is widely used as well.

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{7.5}$$

For all the formulas above, it is assumed that TP, FP, TN and FN are the numbers of true positive, false positive, true negative and false negative predicted pixels respectively.

Additionally, Müller et al. (2022) claimed that the aforementioned class imbalance between the background class and the foreground classes, which applies particularly often to medical datasets, affects the entire approach from preprocessing, to model design and postprocessing as well as validation methods. They stressed that accuracy, or in general metrics that consider true negatives, are a bad choice for class-imbalanced datasets and recommended to use metrics based on IoU and DSC. Furthermore, they pointed out that class imbalance affects multi-class evaluation even more than binary evaluation. They considered a distinct evaluation of each class based on DSC to be the best choice, but stated that besides DSC also IoU, Recall and Precision should be reported for comparison. If it is necessary to reduce the obtained results for the individual classes into a single value, the use of macro- and micro-averaging is proposed. Macro-averaging just averages over the final scores for the individual classes. Micro-averaging computes a single score by aggregating true positive, false positive, false negative and true negative pixels over all images and classes. However, one should be aware of the bias introduced by the background class (or any other highly frequent class) when averaging. The implementation by Iakubovskii (2019), which was used in this thesis, includes “macro-imagewise” and “micro-imagewise” reductions as well. These reductions compute the macro (or micro) reduction independently for each image and then average the obtained image scores over the entire dataset. It should also be noted that the aggregations of the IoU score over images and classes are often simply referred to as mean IoU (mIoU).

7.1.2 Method

The best-case scenario, of course, would be to generate data that can improve any model for semantic segmentation. However, this is a nearly impossible task because there are far too many different approaches towards semantic segmentation to find a general-purpose solution. Furthermore, Macháček et al. (2023), who tested their approach with three different well-known semantic segmentation models, found that their gained improvements correlated with the model architecture. Therefore, they claimed that such an approach needs to be tested with multiple architectures to have a proper estimation of the improvements and/or deterioration. Thus, the decision was to follow their approach and to mainly reuse their validation process. The three main changes to their implementation for validation are:

1. The adaptation for the multi-class case, as the original implementation was only intended for binary datasets.

2. The deactivation of data augmentation during training, as this induces randomness into the training process, which is not desirable for a validation. In addition, the possibility that any improvements or deteriorations come from the data augmentation can be excluded.

3. The calculation of per-class metrics for multi-class semantic segmentation datasets, as recommended by Müller et al. (2022).

The three semantic segmentation models that were used are the U-Net++, the FPN and the DeepLabv3+, which were already introduced in Chapter 2. In particular, the implementations of these models from Iakubovskii (2019) were considered.

In general, for any given dataset, the proposed validation process first trains and evaluates a model on the real data only. This model serves as baseline to which the other models are compared. These

other models are trained with a mix between real and synthetic data or with synthetic data only but are always evaluated on the real test set. The following configurations are the ones that are typically validated, but depending on the experiment and the dataset this may also differ.

- No real data, but all synthetic image-mask pairs (usually 1000, but sometimes also more).
- No real data, but the same number of synthetic pairs as there are existing real pairs in the training dataset. This configuration was used to directly compare the "quality" of real and generated data.
- All real training pairs and the same amount of synthetic data.
- All real training pairs and 100 synthetic pairs.
- All real training pairs and 400 synthetic pairs.
- All real training pairs and 700 synthetic pairs.
- All real training pairs and 1000 synthetic pairs.

The reasoning behind testing these different configurations is that adding too much synthetic data can also deteriorate the model performance. This was already pointed out by Zhang et al. (2021). The incremental configurations have been added in order to control more precisely when the addition of further synthetic data has disadvantages for the model performance. The choice to create synthetic datasets to 1000 image-mask pairs has several reasons: first, the number 1000 is in the same order of magnitude as the number of images in the Lemons Dataset from Section 5.1. Since experiments usually only look at subsets of this dataset, 1000 is even higher than the cardinality of the real data. Secondly, the empirical results showed that there are usually no performance advantages for the datasets used if synthetic datasets with a cardinality above 1000 are used. Therefore, such, larger data sets are only used where it also makes sense to test this. Furthermore, it must be taken into account that depending on the approach, sampling can take up to several days. This is a well-known disadvantage of diffusion models, which has already been addressed by researchers such as Nichol and Dhariwal (2021) and Song et al. (2020).

All the models for semantic segmentation are trained for 50 epochs on the different dataset configurations, which was usually sufficient for the models to converge. One drawback of all the approaches presented in this thesis is that the semantic segmentation models can not be trained at the original image resolution, but 256×256 was chosen as a default resolution. The reason for this is that most of the DM implementations do not support arbitrary image resolutions, but 256×256 seems to be an unofficial standard that is natively supported by all DMs that were used. Therefore, it would be conceivable that only by using the real data, with original resolution, one could train the same semantic segmentation models to outperform all the models presented in this work. However, the primary goal of validation is not to set a new state-of-the-art for the datasets used, but to compare them with the baseline model trained on real data with the same resolution as the synthetic data. The training of the semantic segmentation models is monitored using a validation split of 20 % from the real training set. For the configurations that make use of the same amount of synthetic samples as there are real image-mask pairs, the real training set after performing the validation split is considered. The best model checkpoint is saved and updated based on the mIoU metric of the validation set. All the semantic segmentation models use a “resnet34” encoder with pretrained Imagenet weights. The implementation of Iakubovskii (2019) allows for an automatic preprocessing according to these settings. This is the only preprocessing besides resizing which is just required for the real data. As already mentioned, no data augmentation is applied during training. The real and the synthetic dataset are simply concatenated and shuffled for the training process. Of course, this also induces randomness into the validation process, but the experiments have shown that the results remain quite stable.

After training the models are evaluated on the real test set. This is the first and only occasion where the real test set is used. In particular, the mIoU will be used as main metric for model comparison. The reduction method to compute the mIoU may however change depending on what makes most sense for the given dataset.

7.2 All-Latent Experiments

The experiments with the All-Latent approach served as an estimation for the extent to which LDMs can improve the three semantic segmentation model architectures. All-Latent was tested as a paired image-mask synthesis approach with two LDMs as well as a semantic image synthesis approach using the real semantic masks and the image-generating LDM. The following sections describe the results of these two approaches on some of the binary datasets discussed in Section 5.2. The macro image-wise IoU was chosen as the evaluation metric, because among the possible reduction techniques for the IoU this is the one which is least affected by the pre-dominant background class and thus the scores are in general lower, but with higher differences between them.

7.2.1 Binary Blemish Dataset

The Binary Blemish Dataset is the dataset on which All-Latent was tested most extensively. Besides the comparisons between the semantic image and the paired image-mask synthesis approach, the effect of filtering during postprocessing is tested as well. The results of the experiments with semantic image synthesis are shown in Table 7.1, the ones of paired-image mask synthesis in Table 7.2.

Table 7.1: Results of the All-Latent Approach for Semantic Image Synthesis on the Binary Blemish Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. "#Filtered" indicates how many of the 2000 originally generated image-masks pairs were selected. If "#Filtered" is NONE, for the values of "#S" below 2000, a random subset of the 2000 generated image-mask pairs is chosen.

				U-Net++		**FPN**		**DeepLabv3+**	
ID	**#Filtered**	**#R**	**#S**	**mIoU**	**Diff.[%]**	**mIoU**	**Diff.[%]**	**mIoU**	**Diff.[%]**
Baseline	0	128	0	0.608	0.0	0.565	0.0	0.579	-
1	100	0	100	0.495	-18.5	0.495	-12.4	0.495	-14.4
2		128	100	**0.613**	+0.9	**0.589**	+4.3	**0.589**	+1.8
3	128	0	128	0.495	-18.5	0.495	-12.4	0.495	-14.4
4		128	128	***0.627***	+3.2	**0.598**	+5.9	***0.611***	+5.6
5	400	0	400	0.495	-18.5	0.495	-12.4	0.495	-14.5
6		128	400	**0.611**	+0.5	0.495	-12.4	**0.601**	+3.8
7	700	0	700	0.495	-18.5	0.495	-12.4	0.498	-13.9
8		128	700	**0.625**	+2.9	***0.604***	+6.9	**0.602**	+4.1
9	NONE	0	128	0.571	-6.0	0.542	-4.1	0.532	-8.0
10			2000	0.572	-5.9	**0.574**	+1.5	0.567	-2.0
11		128	100	0.598	-1.6	**0.57**	+0.9	**0.582**	+0.6
12			128	0.571	-6.1	0.542	-4.1	**0.58**	+0.2
13			400	0.599	-1.3	0.495	-12.4	**0.583**	+0.7
14			700	0.598	-1.6	**0.577**	+2.1	0.575	-0.6
15			2000	0.584	-3.8	**0.572**	+1.3	0.579	0.0

Table 7.2: Results of the All-Latent Approach for Paired Image-Mask Synthesis on the Binary Blemish Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. "#Filtered" indicates how many of the 2000 originally generated image-masks pairs were selected. If "#Filtered" is NONE, for the values of "#S" below 2000, a random subset of the 2000 generated image-mask pairs is chosen.

				U-Net++		**FPN**		**DeepLabv3+**	
ID	**#Filtered**	**#R**	**#S**	**mIoU**	**Diff.**[%]	**mIoU**	**Diff.**[%]	**mIoU**	**Diff.**[%]
Baseline	0	128	0	0.608	0.0	0.565	0.0	0.579	-
1	100	0	100	0.527	-13.2	0.497	-12.1	0.527	-8.9
2		128	100	0.605	-0.4	***0.584***	+3.3	0.574	-0.8
3	128	0	128	0.538	-11.4	0.512	-9.4	0.532	-8.1
4		128	128	0.586	-3.5	**0.582**	+3.0	***0.585***	+1.1
5	400	0	400	0.558	-8.1	0.541	-4.2	0.546	-5.6
6		128	400	**0.611**	+0.5	**0.569**	+0.7	0.579	+0.1
7	700	0	700	0.559	-8.0	**0.57**	+0.9	0.563	-2.8
8		128	700	**0.613**	+0.8	**0.57**	+0.9	0.574	-0.8
9	NONE	0	128	0.55	-9.5	0.518	-8.4	0.534	-7.8
10			2000	0.566	-6.9	0.565	-0.1	0.559	-3.3
11		128	100	***0.614***	+1.0	**0.575**	+1.7	0.563	-2.6
12			128	0.604	-0.7	**0.568**	+0.5	0.576	-0.5
13			400	0.597	-1.7	**0.575**	+1.7	**0.580**	+0.3
14			700	0.592	-2.6	**0.581**	+2.8	0.579	+0.1
15			2000	0.598	-1.5	**0.567**	+0.3	**0.580**	+0.2

General Impact of Synthetic Data

The Tables 7.1 and 7.2 show that for both approaches only a few of the tested configurations could outperform the baseline and even those only by a small margin. It also emerges that semantic image synthesis worked better in general than paired image-mask synthesis for all model architectures.

Also, the tables show that the U-Net++ was the best model for the dataset. It had the best baseline performance and also the best overall performance. However, it was also the model that could be improved the least in relative terms. With semantic image synthesis, it improved by a maximum of 3.2 %. While the FPN and the DeepLabv3+ models improved by 6.9 and 5.6 % respectively. The U-Net++ and the DeepLabv3+ improved the most when all the real training data, but only the 128 best synthetic image-mask pairs were used. The FPN benefited from more filtered synthetic image-mask pairs (700) and it was also the only model that could benefit from the non-filtered data for most of the corresponding experiments.

It is odd that the FPN seemed to be absolutely unable to deal with 400 synthetic image-mask pairs, because whether the 400 images and masks were filtered or not (experiments 6 and 13 respectively), the model only achieved an IoU of 0.495, which corresponded to the case when the model always predicts "background" and never a defect. Another notable case among the experiments with the FPN is experiment 10, which was the only case for the semantic image synthesis approach where a model trained only on synthetic data could outperform the baseline.

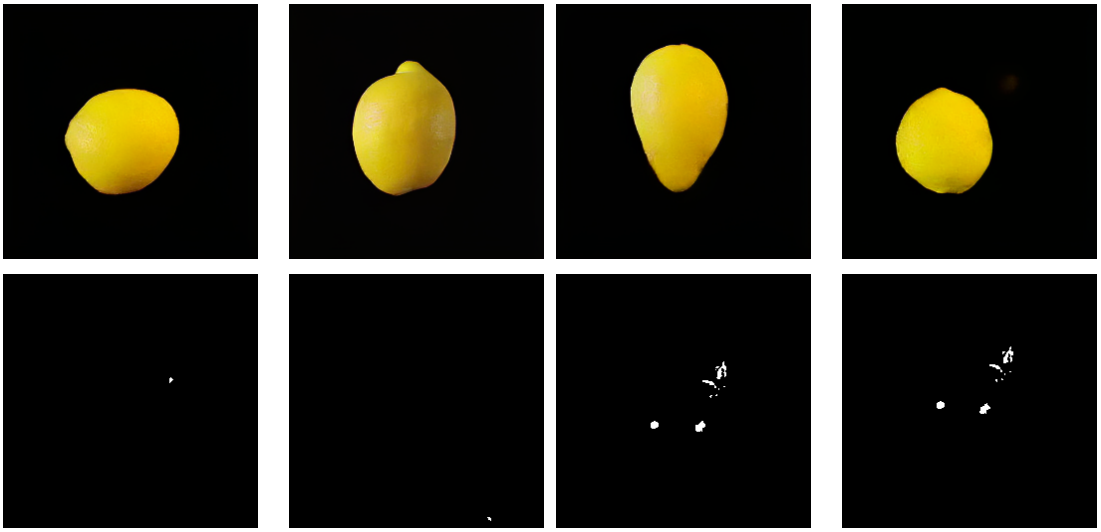
Likewise, the FPN was also the model that benefited the most from the paired image-mask synthesis approach in Table 7.2. While this approach worked bad in general, it is worth mentioning the result of experiment 7 for the FPN model, as this model could compete with the baseline, although it was only trained on synthetic data.

Impact of Filtering in Postprocessing

The Tables 7.1 and 7.2 also reveal the influence of filtering for the semantic image synthesis and paired image-mask synthesis approaches. While filtering almost always brought benefits for the former approach (with the exception for the outliers of the FPN in experiments 6 and 13), this is not so clear for the latter approach. Nevertheless, in four out of six cases (across both tables), the best models turned out to be those for which the synthetic data was heavily filtered ($\#\text{Filtered} \in \{100, 128\}$). These models almost always achieved a higher mIoU than that of the baseline. On the contrary, the use of much more synthetic data (128 image-mask pairs in this case) than real data was available often had no advantages. This suggests that while the models can be improved by adding synthetic data, it is the quality of the added data that matters more than the quantity.

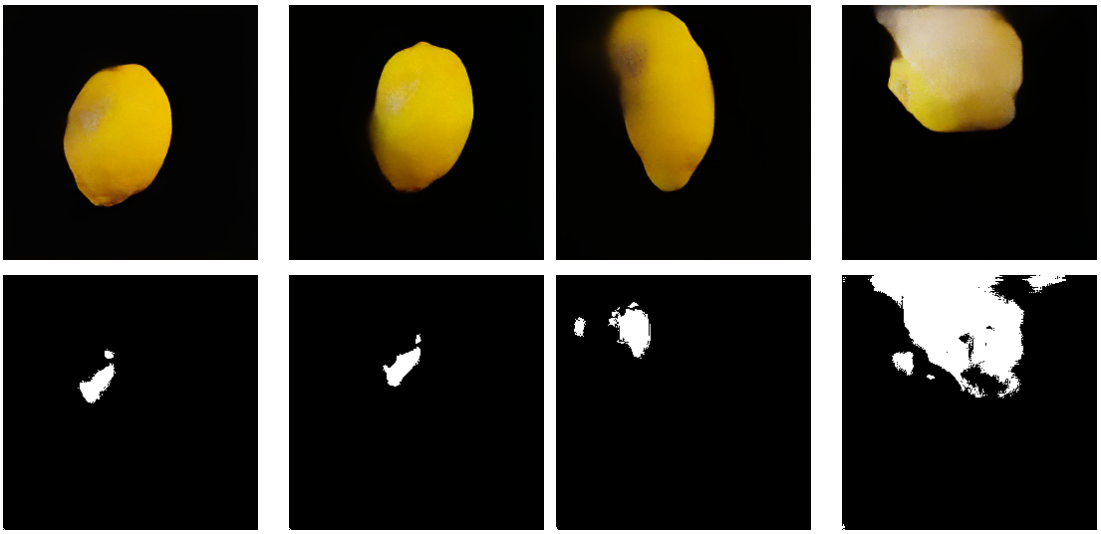
However, a look at the filtered data reveals a problem with the filtering process itself: in Figure 7.1 and in Figure 7.2, it can be seen that the preferred data for each approach presented very similar semantic masks. This was even more noticeable for semantic image synthesis (Figure 7.1), as the selected masks often had only very small defects and these were sometimes not even within the lemon. While this is a sign that the conditioning of the image-generating LDM worked poorly, it is not only negative, because it also shows that the models could be improved just by adding "healthy" lemons. In the case of paired image-mask synthesis (Figure 7.2), it is less clear why some pairs were selected and others not. However, there seems to be a dependency on the shape of the defect region, as masks with similar-looking defects were selected. This is most likely a problem induced by the U-Net used for filtering.

Furthermore, it can also be seen in the figure that both, the generated semantic masks and the images could be of poor quality, which indicates general problems with the mask-generating LDM and with the conditioning for the image-generating LDM. The issues with the former most likely come from the fact that the LDM was trained as if the semantic masks were "natural" RGB images. However, using an LDM approach with grayscale images would have required training an autoencoder specifically for these images, which would have complicated the overall approach further. The problems with the image-generating LDM are probably due to the conditioning process, as the Figure 7.1 shows that the LDM was capable of producing realistic-looking lemons when it was only slightly conditioned. This would also match with the results of Park et al. (2019) and Wang et al. (2022), which showed that conditioning based only on the concatenation of image and semantic mask without further precautions is not optimal. In addition, with LDMs, conditioning takes place in the latent space, which means that necessary spatial information could get lost.



(a) Examples from the best 100 pairs. (b) Examples from the worst 1000 pairs.

Figure 7.1: Examples for Samples from the Semantic Image Synthesis Approach for All-Latent. “Good” and “bad” pairs of images (top) and semantic masks (bottom), according to filtering, are shown.



(a) Examples from the best 100 pairs. (b) Examples from the worst 1000 pairs.

Figure 7.2: Examples for Samples from the Paired Image-Mask Synthesis Approach for All-Latent. “Good” and “bad” pairs of images (top) and semantic masks (bottom), according to filtering, are shown.

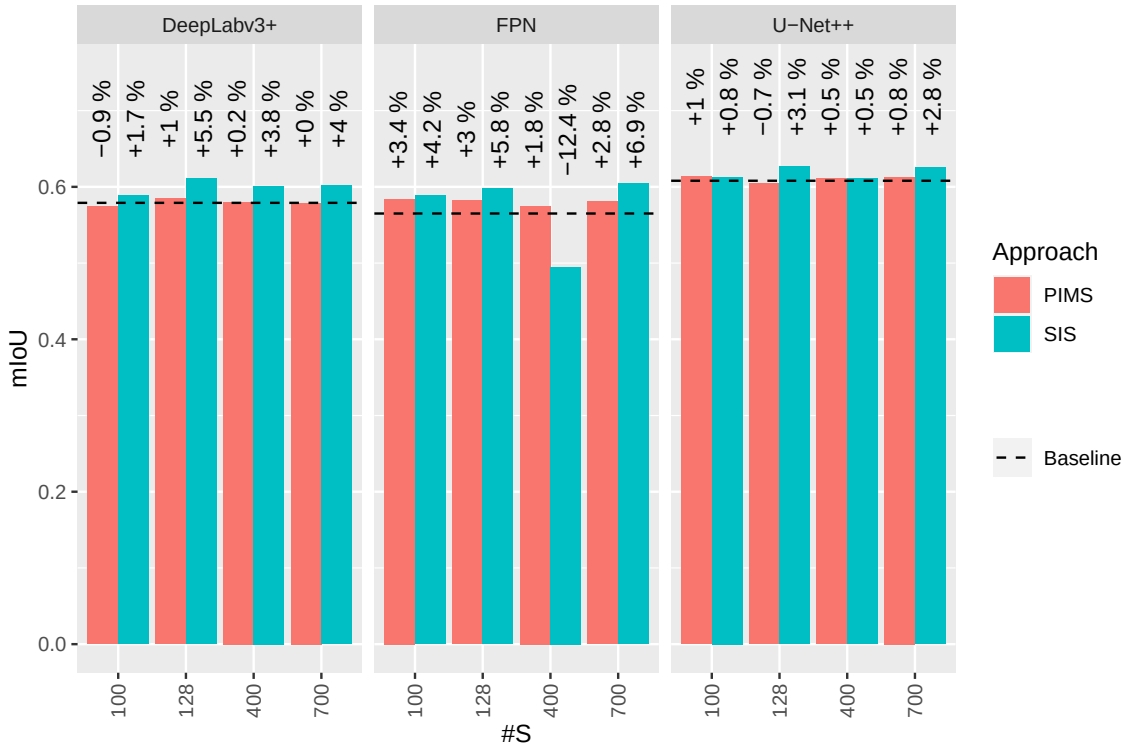


Figure 7.3: Comparison between the Semantic Image Synthesis ("SIS") and the Paired Image-Mask Synthesis Approaches ("PIMS") on the Binary Blemish Dataset. For the comparison, the models trained with all real and different numbers of synthetic data ("#S") are used. The percentages refer to the improvements/deterioration compared to the baseline.

Comparison between Semantic Image Synthesis and Paired Image-Mask Synthesis

The comparison between the semantic image synthesis and the paired image-mask synthesis approaches in Figure 7.3 shows that the first could outperform the latter for almost all amounts of synthetic data and model architectures. The only exceptions were the U-Net++ models for 100 and 400 synthetic samples, where the two approaches showed nearly identical performance and the FPN model for 400 samples. The latter however corresponds to the outlier case mentioned above.

7.2.2 Binary Mould Dataset

Due to the modest results of the Binary Blemish Dataset, the idea was to test the approaches on a smaller and simpler dataset for debugging reasons. The Binary Mould Dataset would actually be perfect for this, as the defect regions for this dataset are mostly large and easily recognisable. The problem that can be seen in Figure 7.1 could therefore not occur at all, as there are no such small defect regions. Unfortunately, the binary mould dataset proved to be too small to train the LDMs sufficiently. The models were not able to remove the noise, even after completing the training process.

7.2.3 Binary Mould-Blemish-Gangrene Dataset

Since it has been shown that smaller datasets than the Binary Blemish Dataset do not consist of sufficient data to train the LDMs, the All-Latent approach should be additionally tested on a larger binary dataset. In order to maintain comparability with the Binary Blemish Dataset, the only slightly larger Binary Mould-Blemish-Gangrene Dataset was chosen for a second series of tests with the All-Latent approach. Again, both the semantic image synthesis and the paired image-mask synthesis methods were tested. Further tests with the filter methodology were omitted because although it could improve the results significantly, especially for semantic image synthesis, it did not show the desired behaviour: while it

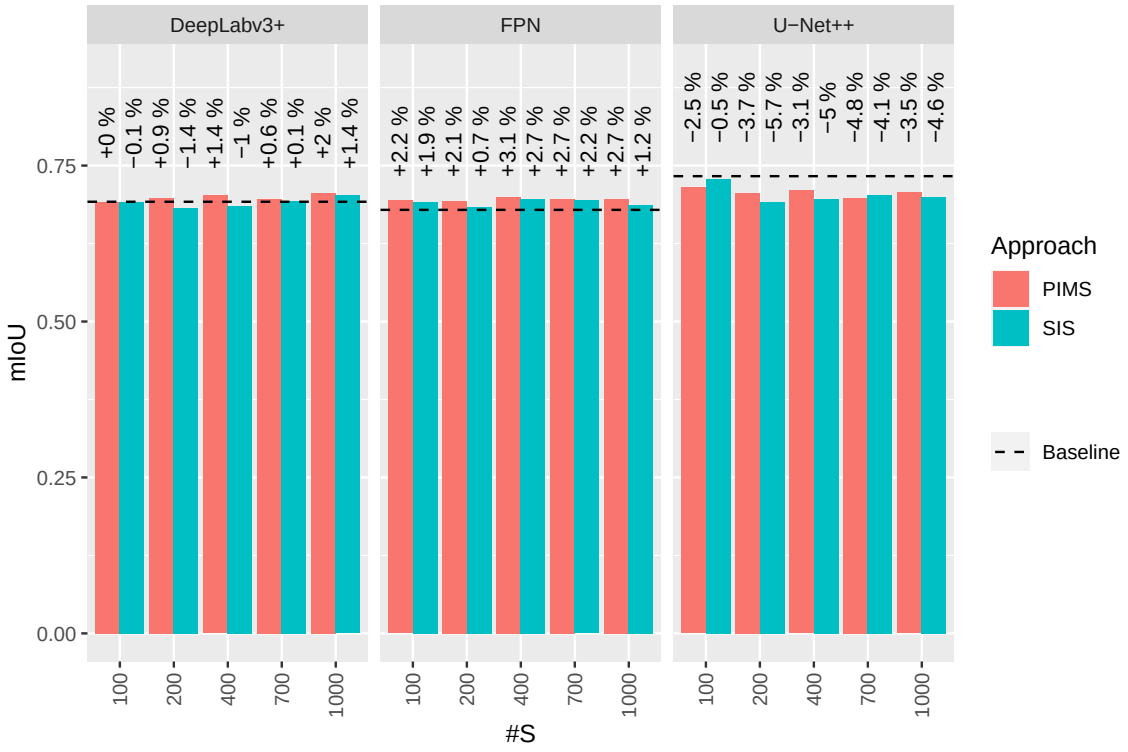


Figure 7.4: Comparisons between the Semantic Image Synthesis ("SIS") and the Paired Image-Mask Synthesis Approaches ("PIMS") on the Binary Mould-Blemish-Gangrene Dataset. For the comparison, the models trained with all real and different numbers of synthetic data ("#S") are used. The percentages refer to the improvements/deterioration compared to the baseline.

is a notable result that binary semantic segmentation datasets can be "improved" by adding (almost) only anomaly-free synthetic data, it does not really align with the goal of enhancing the datasets in a controlled way.

The results for the semantic image synthesis and the paired image-mask synthesis approaches on the Binary Mould-Blemish-Gangrene Dataset are compared in Figure 7.4. They differed depending on the model architecture. The U-Net++ model only deteriorated for all configurations in comparison to the baseline. The FPN, as was the case for the Binary Blemish Dataset, was able to benefit most from the synthetic data: for both approaches, small but constant improvements were achieved, ranging from 0.7 to 2.7 % for semantic image synthesis and from 2.1 to 3.1 % for paired-image-mask synthesis. For the DeepLabv3+, maximum improvements of 2.0 % could be achieved. These results are consistent with the ones for the Binary Blemish Dataset, where, likewise without filtering during post-processing, little advantage could be gained from the approaches.

7.3 ISDM Experiments

Due to the modest results of the All-Latent approach and also due to its complexity - after all, for the complete paired image-mask synthesis approach it is necessary to train two diffusion models and an additional semantic segmentation model for filtering - the decision was to opt for a simpler approach to achieve the objectives. In particular, the semantic image synthesis implementation of Wang et al. (2022) should be adapted and tested for the given problem, because All-Latent with semantic image synthesis (and filtering) proved to be more efficient and since it also seemed to be the more well-known approach in the literature. Therefore, the ISDM method was developed. The following sections report the results of this method on several binary, but also one multi-class dataset discussed in Section 5.2. For the binary datasets, the macro image-wise mIoU aggregated over classes and images served as the main evaluation metric, whereas for the multi-class case the image-wise mIoU was calculated individually for each class.

7.3.1 Binary Blemish Dataset

General Impact of Synthetic Data

The results for the experiments with the ISDM method on the Binary Blemish Dataset are reported in Table 7.3. It emerged that nearly all of the tested configurations could outperform the baseline, even if sometimes only by a small margin. From 400 generated image-mask pairs onward, the results improved significantly and were consistently above 4.1 % for all models. For the U-Net++ and the DeepLabv3+, the models seemed to improve even further as more synthetic data was added, although the differences between the configurations with #S=700 and #S=1000 were marginal. Another interesting result is that for all models it was possible to train them on synthetic data only and achieve mIoU scores comparable to the baseline. However, this was only possible if many more synthetic image-mask pairs (1000) were used than the 128 real pairs with which the baseline was trained. If the same number of synthetic data was used, the mIoU scores were significantly lower than those of the baseline, which is an indication that the generated data is not "as good" as the real data.

Table 7.3: Results of the ISDM method on the Binary Blemish Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. For the values of "#S" below 1000, a random subset of the 1000 generated image-mask pairs is chosen.

			U-Net++		**FPN**		**DeepLabv3+**	
ID	**#R**	**#S**	**mIoU**	**Diff.**[%]	**mIoU**	**Diff.**[%]	**mIoU**	**Diff.**[%]
Baseline	128	0	0.608	-	0.565	-	0.579	-
1	0	128	0.539	-11.4	0.532	-5.9	0.537	-7.2
2		1000	**0.615**	+1.1	**0.578**	+2.3	**0.580**	+0.2
3	128	100	0.607	-0.2	**0.575**	+1.8	0.572	-1.2
4		128	**0.614**	+1.0	**0.577**	+2.0	0.578	-0.2
5		400	**0.638**	+4.9	**0.593**	+4.9	**0.611**	+5.5
6		700	**0.641**	+5.5	***0.596***	+5.5	**0.615**	+6.2
7		1000	***0.642***	+5.7	**0.588**	+4.1	***0.616***	+6.4

Comparison to the All-Latent Method

The results stated above are an improvement over the All-Latent method, which is clarified by the comparison in Figure 7.5. It reveals that usually from 400 generated samples onward, the ISDM method could consistently outperform the baseline and also the All-Latent approach for all model architectures.

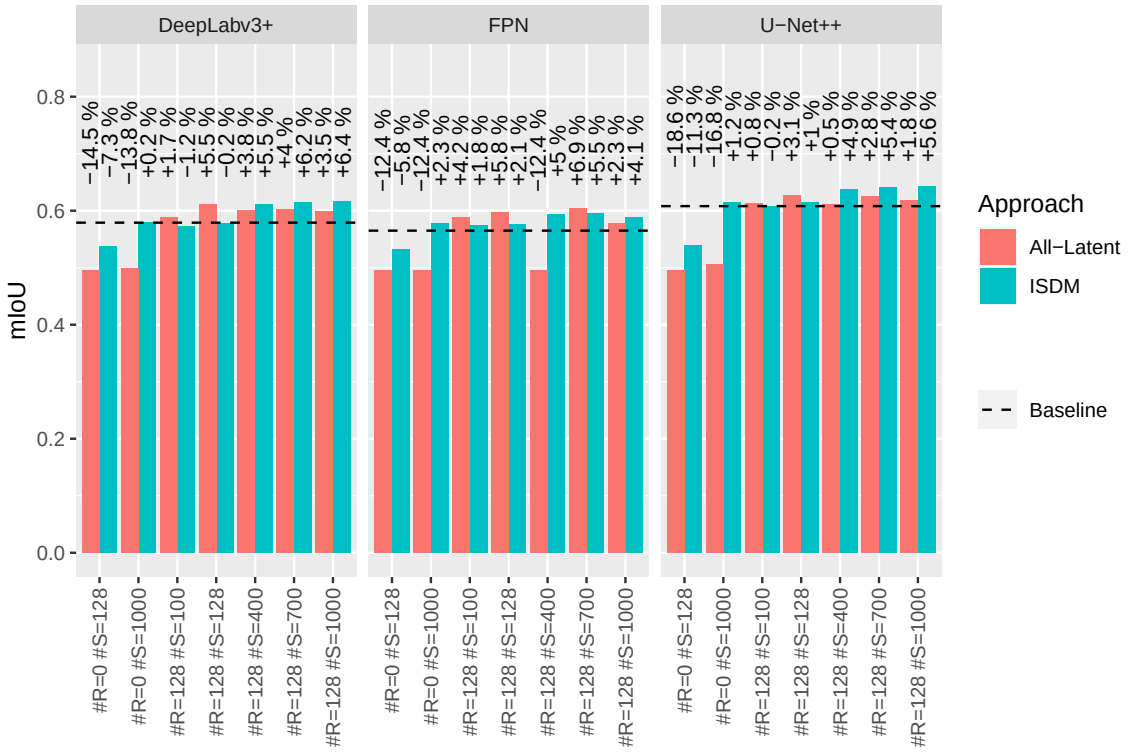


Figure 7.5: Comparison between the All-Latent approach for Semantic Image Synthesis and the ISDM Method on the Binary Blemish Dataset. The label below each pair of bars indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model.

These results were achieved without any filtering in post-processing. Furthermore, ISDM outperformed All-Latent by a large margin for the models that are trained only with synthetic data.

Discussion of Observed Sampling Issues

Although the ISDM method proved to be significantly more efficient than All-Latent, there were also some disadvantages compared to the latter. Figure 7.6 demonstrates that the model trained on the Binary Blemish Dataset was not able to completely remove the noise even after three days of training. In comparison with Figure 7.1 for the All-Latent method, it is obvious that All-Latent removed the noise in the background but also on the lemon in a cleaner way. The background in the latter image is completely black, as in the original images, which could not be achieved with ISDM.

Another unpleasant side effect of the ISDM method is most probably provoked by the strong conditioning: Figure 7.6 reveals that the SDM models tend to overfit for the Binary Blemish Dataset. The positions and orientations of the generated lemons are identical to the real image. This overfitting was observed to be more pronounced the larger the defect regions. On the positive side, at least the shape of the lemons seemed to vary slightly and the conditioning worked so well that the defects were even better recognisable than on the original.

7.3.2 Binary Mould Dataset

Discussion of Overfitting

In contrast to the unusable results for the All-Latent method with the small Binary Mould Dataset (see Section 7.2.2), the ISDM method was able to sample usable images for this dataset as well. However, the experiments with such a small dataset resulted in a clear overfitting with respect to the input dataset, which is illustrated in Figure 7.7. Despite the clear overfitting, it seemed reasonable to validate the generated data and include the results in this work, because the objective to improve upon the baseline

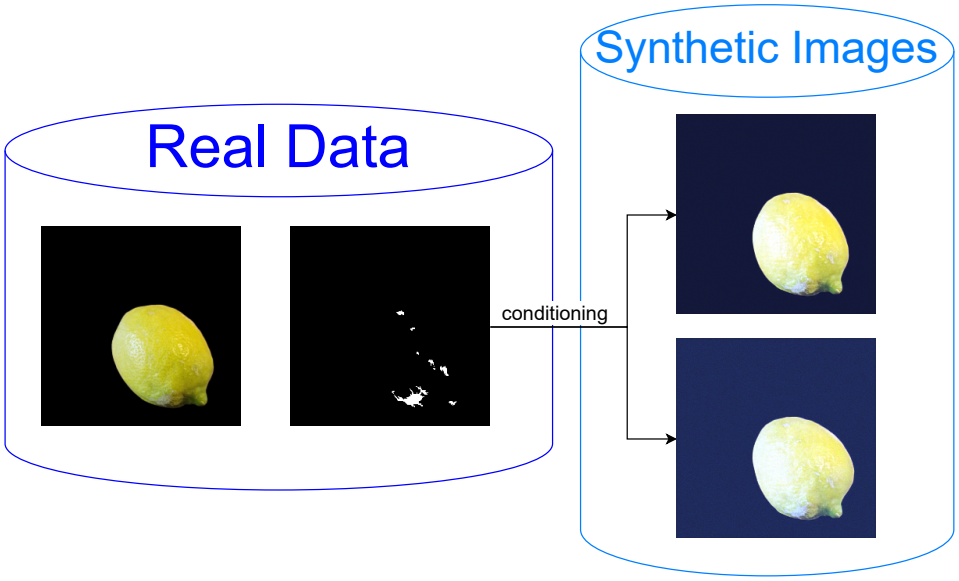


Figure 7.6: Example for the Overfitting of the ISDM Method on the Binary Blemish Dataset. Although only the semantic mask and not the real image itself serves as input for conditioning, the trained SDM generates images that look very similar to the original.

for the semantic segmentation models could still be achieved.

General Impact of Synthetic Data

The results for the ISDM method on the Binary Mould Dataset are reported in Table 7.4. The U-Net++, which as for the Binary Blemish Dataset, had best baseline performance, could be improved only marginally for two of the tested configurations. The FPN and the DeepLabv3+ could be improved for almost all configurations. This is a notable result for this dataset since the baseline mIoUs were already close to the optimum (1.0) and thus in a range in which it is often difficult to improve models.

As was the case for the Binary Blemish Dataset, the DeepLabv3+ was able to benefit the most from the data generated with the ISDM method, but the FPN was the only model that could keep up with the baseline even when trained on synthetic data alone. It is also interesting to note that the FPN and the DeepLabv3+ performed both poorly for the configuration used in experiment 1, while the U-Net++ only lost 9.4 % for this experiment, in which the real data was replaced one-to-one with synthetic data.

7.3.3 Binary Illness Dataset

General Impact of Synthetic Data

The experiments on the Binary Illness Dataset performed poorly (see Figure 7.8). Only a few of the tested configurations were able to achieve improvements compared to the baseline, which is approximately 0.57 for all three models. The FPN and the U-Net++ benefited the most, for the last configuration with #R=66 #S=1000. The DeepLabv3+ could not benefit significantly from any of the configurations. This shows that the improvements that can be achieved for a semantic segmentation model also depend on the input dataset, because for the experiments on the first two datasets DeepLabv3+ was able to achieve the highest improvements among all models.

Discussion of Sampling Observations

Two observations made while sampling with SDMs trained on the Binary Illness Dataset and other datasets should be briefly highlighted in the following. In addition, it should be shown that with time step rescaling 300 diffusion steps were enough to sample images that were comparable to the ones sampled with 1000 steps without rescaling.

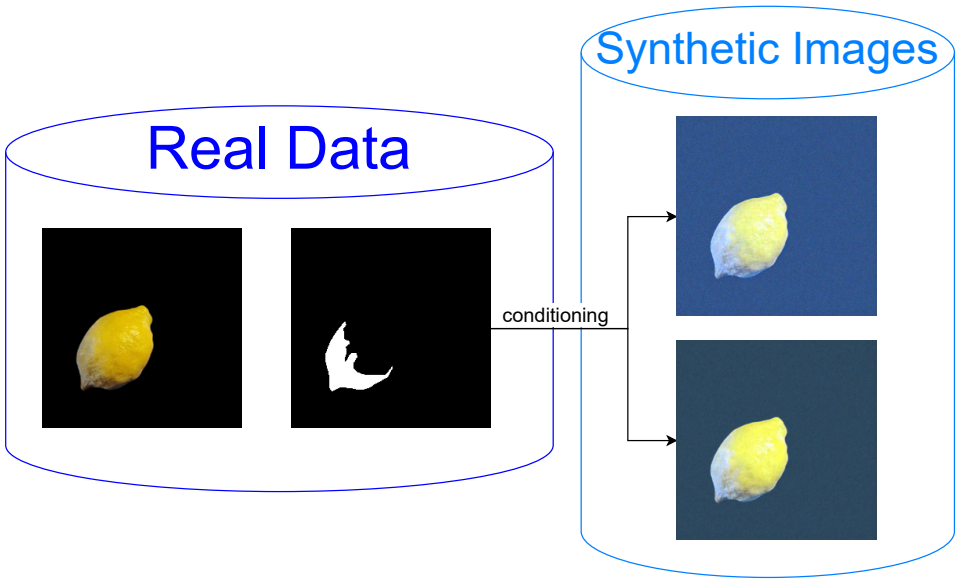


Figure 7.7: Example for the Overfitting of the ISDM Method on the Binary Mould Dataset. Although only the semantic mask and not the real image itself serves as input for conditioning, the trained SDM generates images that show hardly any changes in shape and size compared to the original.

Table 7.4: Results of the ISDM Method on the Binary Mould Dataset. The first row represents the baseline trained without synthetic data. All the other rows are compared to this baseline in terms of relative difference ("Diff." columns) between the mIoUs. The mIoU scores that are higher than the baseline are written in bold and the maximum value is written in bold italics. "#R" is the number of real pairs and "#S" is the number of synthetic pairs that were used to train the models. For the values of "#S" below 1000, a random subset of the 1000 generated image-mask pairs is chosen.

			U-Net++		**FPN**		**DeepLabv3+**	
ID	**#R**	**#S**	**mIoU**	**Diff.[%]**	**mIoU**	**Diff.[%]**	**mIoU**	**Diff.[%]**
Baseline	24	0	0.936	-	0.921	-	0.915	-
1	0	24	0.848	-9.4	0.471	-48.8	0.474	-48.2
2	0	1000	0.889	-5.1	**0.922**	+0.1	0.886	-3.2
3	24	24	***0.946***	+1.0	**0.939**	+2.0	**0.942**	+2.9
4	24	100	0.934	-0.2	**0.934**	+1.4	**0.940**	+2.8
5	24	400	0.932	-0.4	**0.937**	+1.7	**0.936**	+2.3
6	24	700	0.930	-0.6	**0.942**	+2.3	***0.949***	+3.7
7	24	1000	**0.936**	+0.1	***0.945***	+2.6	**0.942**	+3.0

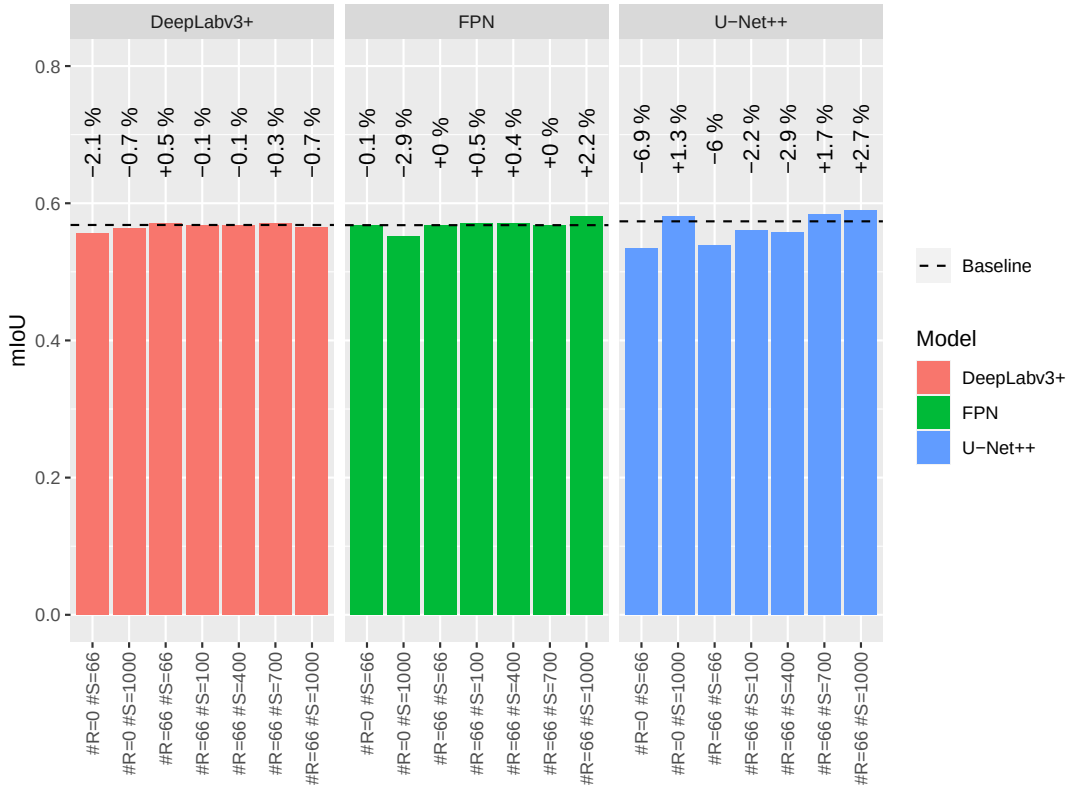


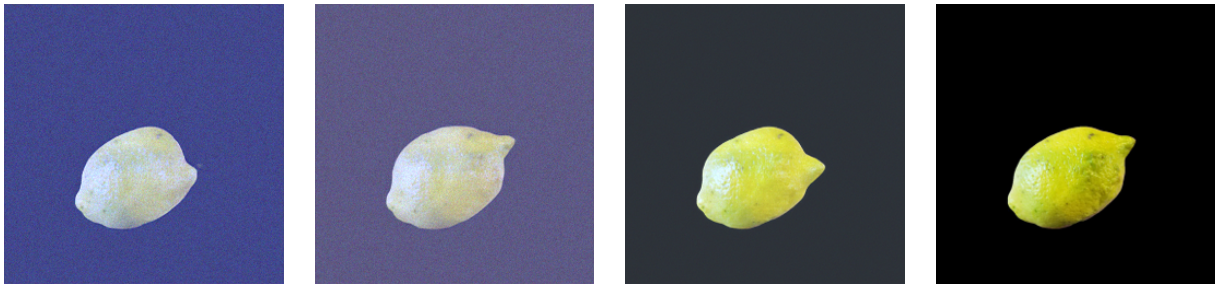
Figure 7.8: Results for the ISDM Method on the Binary Illness Dataset. The label below each bar indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline.

(a) Sample from SDM Trained for 29142 Steps.

(b) Sample from SDM Trained for 53664 Steps.

(c) Sample from SDM Trained for 53664 Steps without Rescaling.

Figure 7.9: Example for Three Differently Sampled Images. All three images were sampled from the same SDM, but at different points in training, images (a) and (b), or with and without the rescaling of diffusion steps, images (b) and (c).



(a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Real Image.

Figure 7.10: Example of Three Identically Sampled Images of Different Perceived Quality. The same SDM and the same semantic mask were used for generating all three samples. The semantic mask belongs to the real image shown in (d).

An example of the effect of longer training and time step rescaling is shown in Figure 7.9. All three images in this figure were samples with the same semantic mask for conditioning. The first image, which appears the noisiest among all three, was sampled from an SDM trained for 29142 steps on the Binary Illness Dataset and sampled with 300 diffusion steps and time step rescaling. The second image was sampled with the same procedure from a later checkpoint (53664 optimisation steps) of the same model, it shows some noise as well but seems the best among all three. The last image was sampled from the same model, but using 1000 diffusion steps for sampling without rescaling. It seems better than the first, but worse than the second, although its sampling process lasted more than three times as long.

The second observation was that the same SDM when conditioned with the same semantic mask, could generate samples of vastly different quality. This is illustrated in Figure 7.10 and is a reason why researchers, such as Shao et al. (2023), chose to design filtering mechanisms to exclude bad samples. Besides the problems with the different quality, there is also an overfitting in the samples, but this was to be expected, as it was also observed for the other two datasets discussed above.

7.3.4 Binary Blemish-Illness-Mould Dataset

Discussion of Observations

Since the ISDM method on the Binary Blemish Dataset tended to overfit, the behaviour should be tested on a larger dataset. Therefore, the Binary Blemish Illness Mould dataset was chosen, which combines the three most common defect classes into a single one. Figure 7.11 shows that the problem of overfitting reduced for this larger dataset: for the identical conditioning information, the SDM could sample two

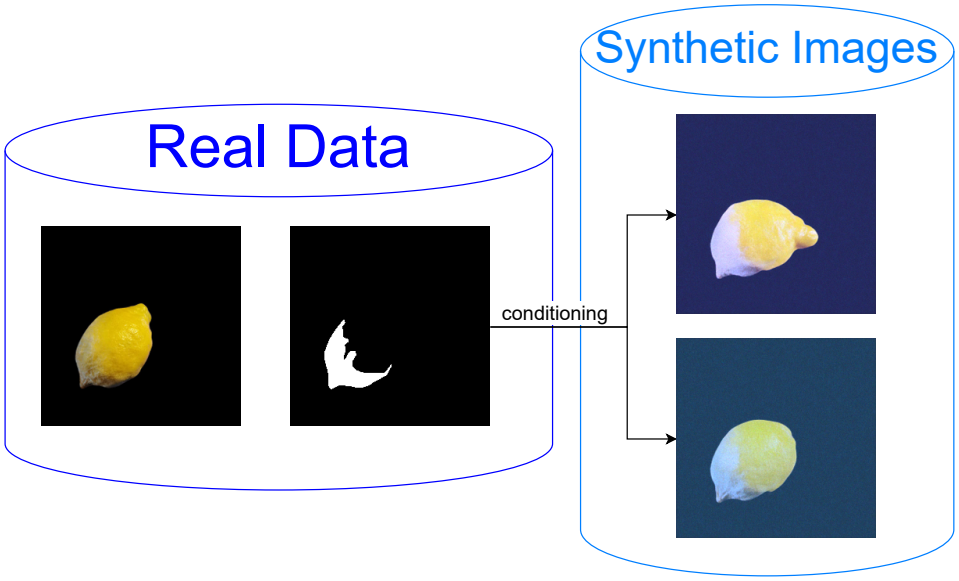


Figure 7.11: Example for Diverse Samples Generated from the Binary Blemish-Illness-Mould Dataset. The two synthetic images on the right were sampled using the same semantic mask for conditioning. They show no signs of overfitting and the defective region induced by the semantic mask is clearly visible on both.

lemons that differ from each other in shape, colour and certain features (see the top end of the lemons). They also differed significantly more from the real counterpart than the examples in Figure 7.7 for which the same real image and semantic mask served as reference.

General Impact of Synthetic Data

Although getting rid of overfitting is a desirable result, it is not enough to automatically improve the downstream semantic segmentation task. This can be seen in Figure 7.12, as the generated data could hardly achieve improvements compared to the baseline regardless of whether finetuning was used or not. However, for all three models, the rightmost configuration (#R=504 #S=1000), which used all available real and synthetic data, performed best. This could be a sign that the 1000 generated image-mask pairs per model might not be sufficient to achieve improvements. Therefore, this choice was tested again in the ablation experiments in Section 7.3.6.

Impact of Finetuning

The effect of the finetuning stage of the ISDM method is represented in Figure 7.12 as well. The data generated with the finetuned SDM usually achieved higher improvements than the data generated with the non-finetuned model. However, for most of the tested configurations, the differences were marginal. The largest differences could be observed when the segmentation models were trained on synthetic data only. These differences amounted to up to 6.4 % for the U-Net++, up to 9.4 % for the FPN and up to 7.7 % for the DeepLabv3+ when compared to the baseline. So, if the aim is to completely replace the real dataset with a synthetic one, finetuning should be considered.

7.3.5 Multi-Class Blemish-Illness-Mould Dataset

General Impact of Synthetic Data

The results for the multi-class blemish-illness-mould dataset are shown in Figure 7.13. For all three models and for all three classes, improvements could be achieved through the synthetic data. The extent of these improvements differed depending on the model and class, but was generally low. The DeepLabv3+ is the model that could be improved the least. It achieved the greatest improvement for “blemish” (7.0 %)

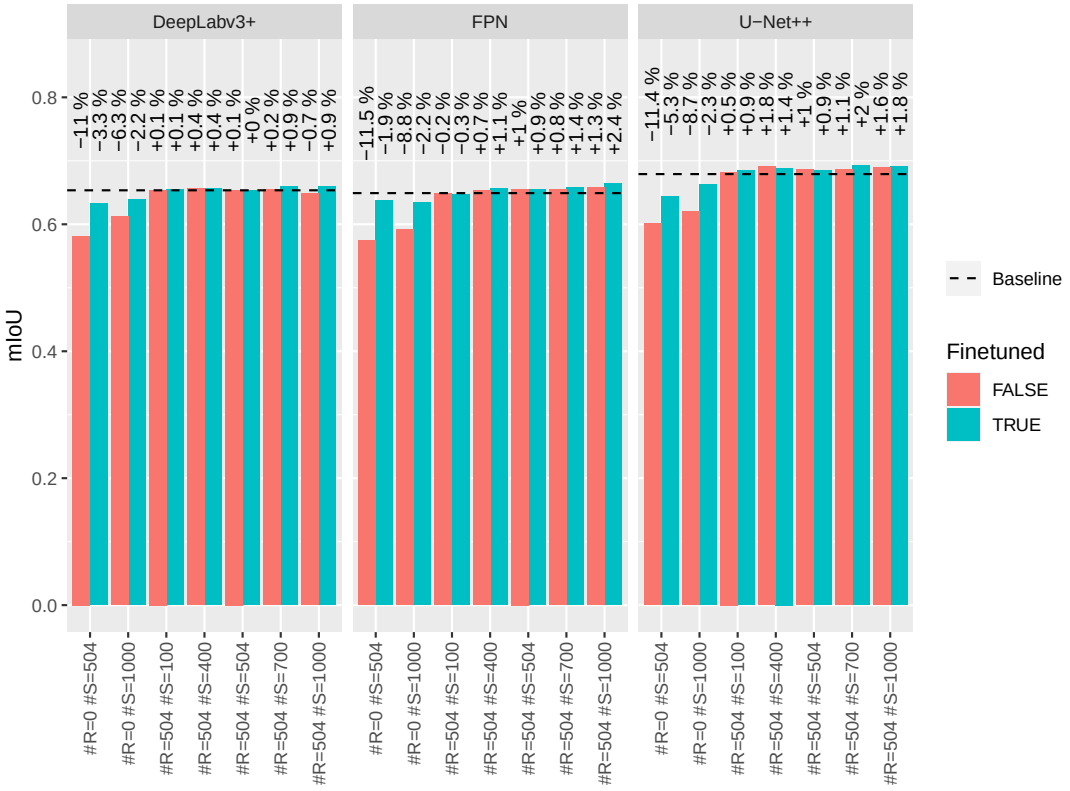


Figure 7.12: Results for the ISDM Method on the Binary Blemish-Illness-Mould Dataset for a Finetuned and a Non-Finetuned SDM. The label below each pair of bars indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model.

when the configuration #R=504 #S=700 was used. With this configuration, however, there was a minus of 1.8 % for the "illness" class. "Illness" is also the class for which the DeepLabv3+ could not achieve significant improvements at all. The FPN and the U-Net++, on the other hand, could actually improve the most for this class, with a 7.2 % gain compared to the baseline.

Overall, the FPN was also the model that could achieve the greatest improvements for the classes "blemish" and "mould" with 5.9 and 3.1 % respectively. However, the model reached its best improvements for each class with different configurations: for "blemish" and "illness", the use of many synthetic image-mask pairs (700 or 1000) proved to be particularly good. "Mould", on the other hand, improved most when only 100 synthetic images were added to the real dataset. A similar configuration-dependent behaviour was also observed for the other models.

However, there were also configurations for each model with which all three classes improved, or at least did not deteriorate, compared to the baseline. Therefore, one of the configurations #R=504 #S=700 or #R=504 #S=1000 should be chosen for the FPN, as these gave the best average improvements for this model. A similar situation arose for the U-Net++, where the three classes each benefited most from different dataset configurations. For this model, the configuration #R=504 #S=504 could offer the best compromise. For the DeepLabv3+, the configurations #R=504 #S=400 and #R=504 #S=700 were the only ones where adding the synthetic data did not degrade any of the classes. None of the models achieved results that could compete with the baselines when trained on synthetic data alone.

Comparison to the Binary Datasets

Figure 7.14 shows a comparison between the results for the multi-class dataset and the individual datasets for the classes "blemish", "illness" and "mould". For the multi-class case the baseline and all other results were significantly higher than for the binary cases. This is probably due to the fact that the Multi-class Blemish-Illness-Mould Dataset is much larger than the other three datasets. This behaviour was also observed for the experiments with the Binary Blemish-Illness-Mould Dataset which featured higher mIoUs than the smaller Binary Blemish and Binary Illness datasets.

The mIoU scores for the "mould" class were very high in general. This indicates that it was easier for the semantic segmentation models to learn "mould" than the other two classes, which was also expected, because among all defects in the Lemons Dataset, "mould" is also the easiest defect for humans to recognise. Accordingly, the differences in the mIoU scores between binary and multi-class cases are smaller for "mould" than for the other defects, because the segmentation models reached already reasonable results for the small Binary Mould Dataset. It is therefore surprising that the FPN and the DeepLabv3+ could still improve significantly for the "mould" class in the binary case. For the U-Net++, on the other hand, and for the multi-class case, the improvements are smaller.

For the other two binary datasets, their scarcity is most probably the reason why the classes can be improved significantly more by synthetic data than in the multi-class case, which can be seen in Figure 7.14 by looking at the percentages of improvement/deterioration compared to the respecitve baseline. However, the amount of improvement was not the only difference between the results for the binary cases and the multi-class case. The configurations for which improvements could be achieved differed too.

Furthermore, the extent of improvements seemed to depend on classes and models. For example, the FPN could be improved with several different configurations for both binary and multi-class cases. The U-Net++ and DeepLabv3+ were able to achieve improvements for "blemish" with different configurations, however, they did not improve on "illness" with most of the configurations. For the "mould" class, the U-Net++ could hardly be improved, neither in multi-class nor in binary mode. The DeepLabv3+, on the other hand, could achieve notable improvements for four configurations in binary mode. In multi-class mode, however, the results for this model were not as good as for the FPN.

The ISDM method thus showed to be most effective for small binary segmentation datasets. It could also achieve improvements in the multi-class case, but these varied depending on the class and model. However, as long as the synthetic data was used together with the real data, almost none of the experiments showed deterioration, so using the method at least did no harm to the semantic segmentation models. In some cases, improvements could even be achieved with synthetic data alone.

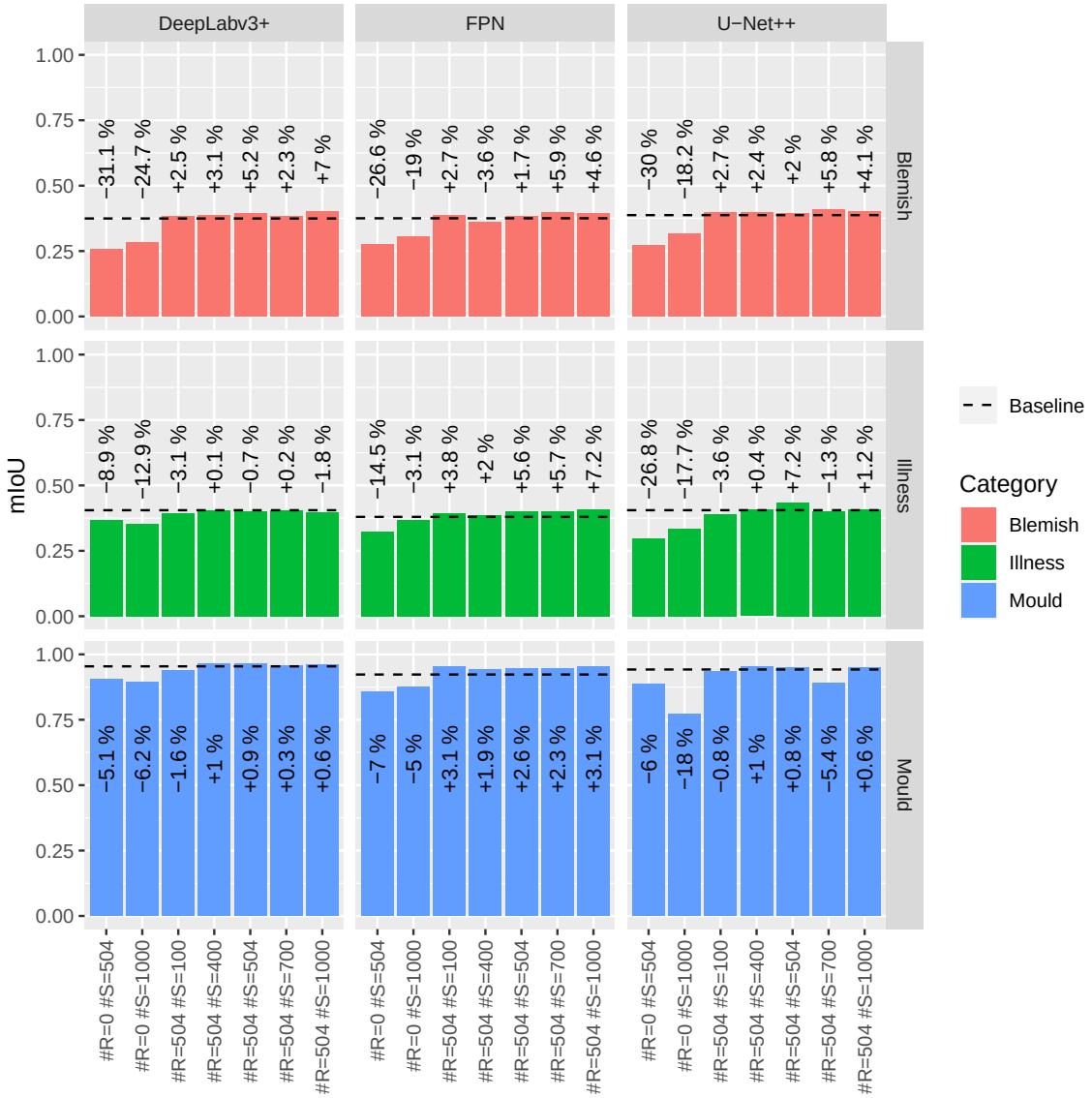


Figure 7.13: Results for the ISDM Method on the Multi-Class Blemish-Illness-Mould Dataset. The label below each pair of bars indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline.

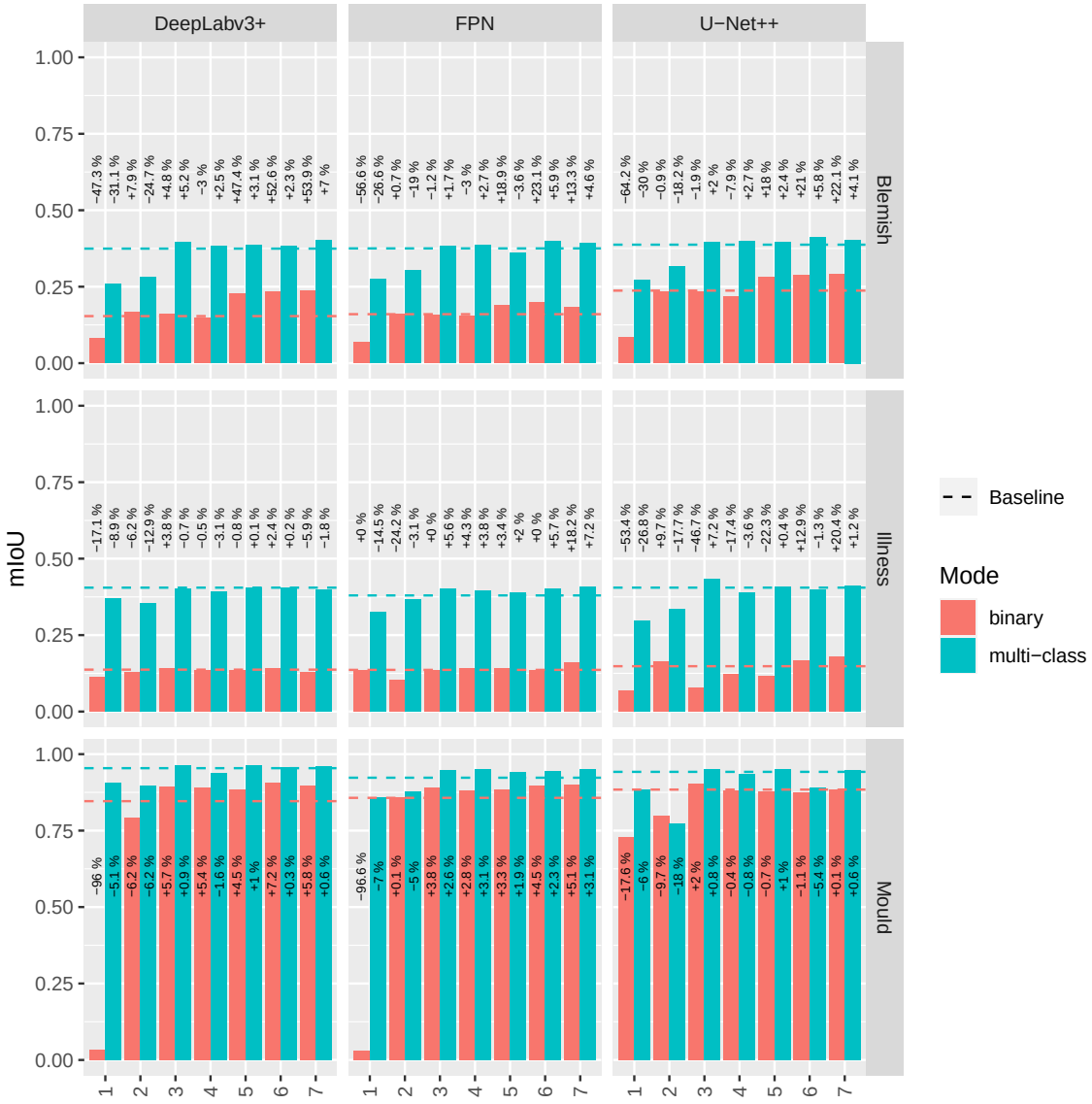


Figure 7.14: Comparison between the Multi-class and the Binary Mode for the Classes “Blemish”, “Illness” and “Mould”. The labels on the x-axis stand for the different tested configurations: 1. no real data, but as many synthetic image-mask pairs as there are in the real dataset, 2. no real data and 1000 synthetic pairs, 3. all real data and the same amount of synthetic data, 4. all real data and 100 synthetic pairs, 5. all real data and 400 synthetic pairs, 6. all real data and 700 synthetic pairs, 7. all real data and 1000 synthetic pairs. The percentages refer to the improvements/deterioration compared to the respective baseline.

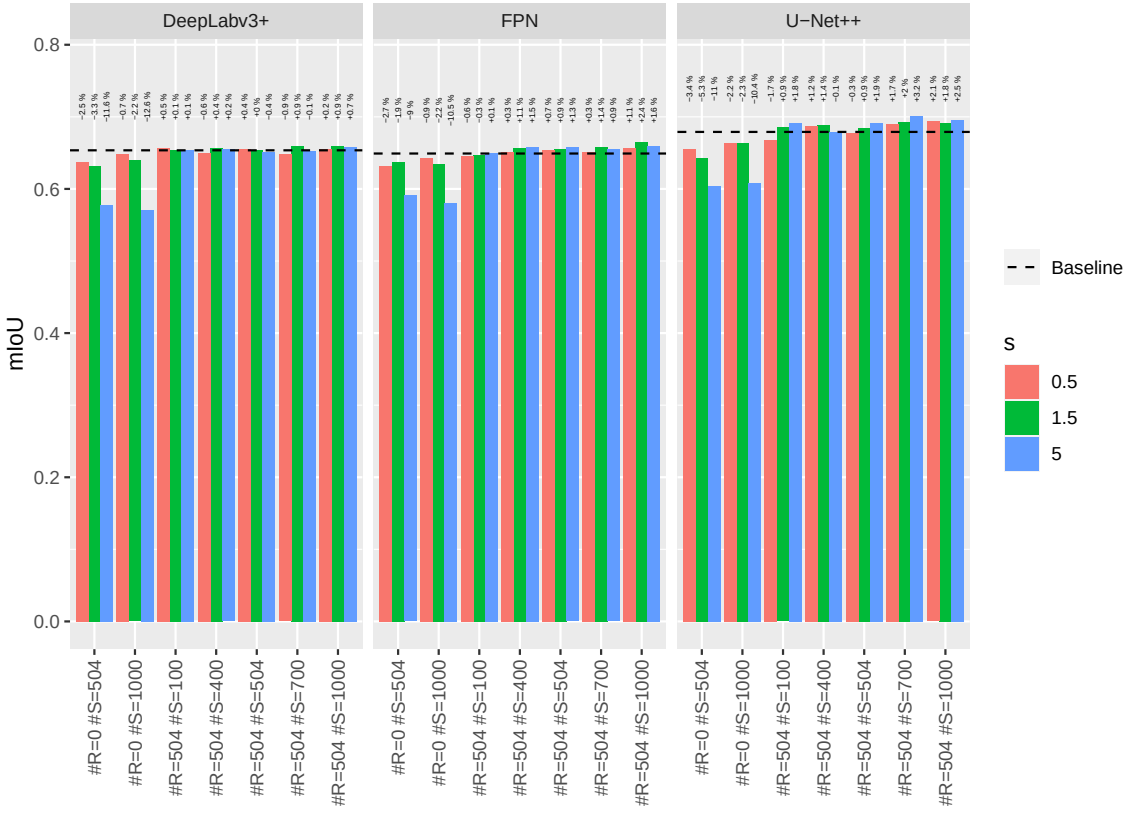


Figure 7.15: Comparison of Different Guidance Scales on the Binary Blemish-Illness-Mould Dataset. The label below each bar indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline.

7.3.6 Ablation Experiments

In this section, some of the design choices for the ISDM method are ablated.

Guidance Scale

One of the design decisions for the ISDM method was to stay with the guidance scale $s = 1.5$ that was also used by Wang et al. (2022) for most of the datasets. To ablate this choice additional experiments were conducted with guidance scales $s = 0.5$ and $s = 5$ for the Binary Blemish-Illness-Mould Dataset.

The results of these experiments are reported in Figure 7.15. They show that a lower guidance scale of $s = 0.5$ worked worse than the two higher guidance sales for most of the configurations tested. However, it worked best in 5 out of 6 cases across all model architectures where only synthetic data was used. For these configurations, the results seemed to deteriorate for higher values of s, because for all models, the highest guidance scale ($s = 5$) proved to be significantly worse than the other two. On the other hand, for the configurations using real and synthetic data, setting $s = 5$ could also bring advantages, especially for the U-Net++. Overall, the guidance scale of $s = 1.5$ used for all other experiments proved to be a suitable trade-off.

Diffusion Steps During Sampling

One of the changes in the ISDM method compared to the original implementation of Wang et al. (2022) was the use of $T = 300$ instead of $T = 1000$ diffusion steps during sampling. This choice was ablated on the Binary Blemish Dataset. Figure 7.16 shows the comparison between using $T = 300$ with time step rescaling and $T = 1000$ without time step rescaling.

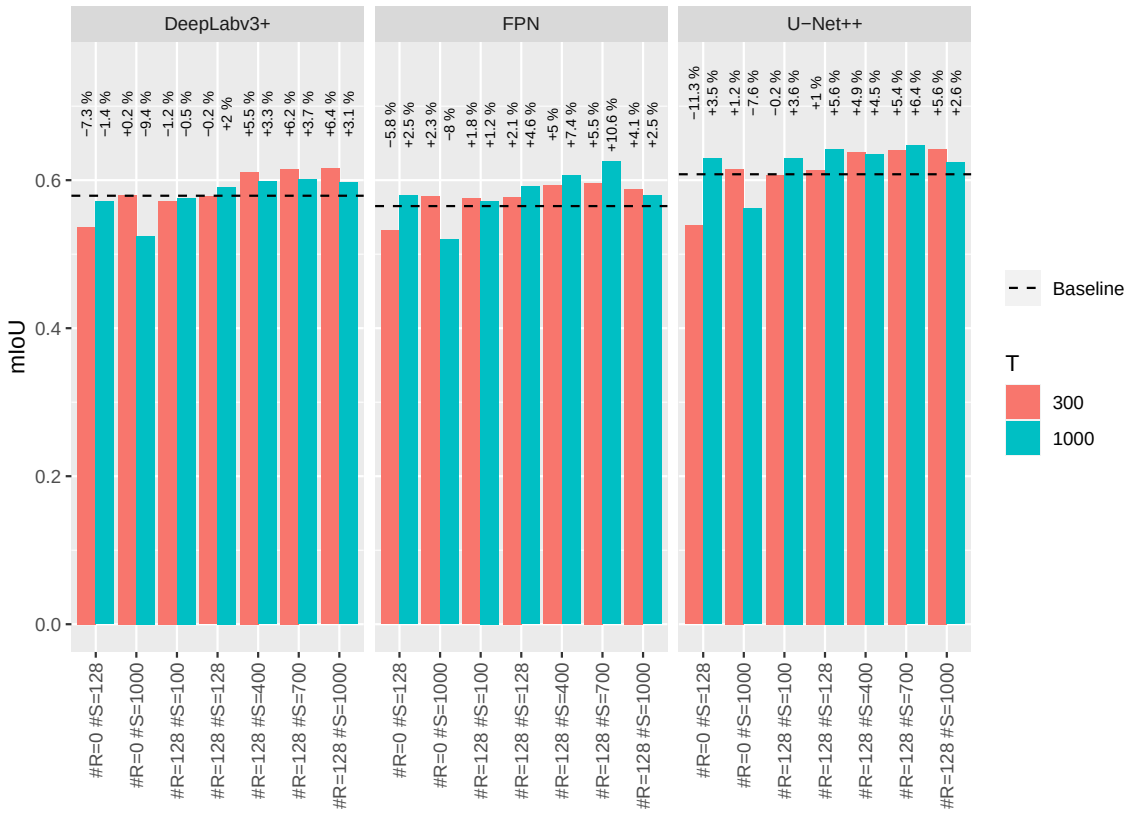


Figure 7.16: Comparison of Different Numbers for the Sampling Steps (T) on the Binary Blemish Dataset. The label below each bar indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline.

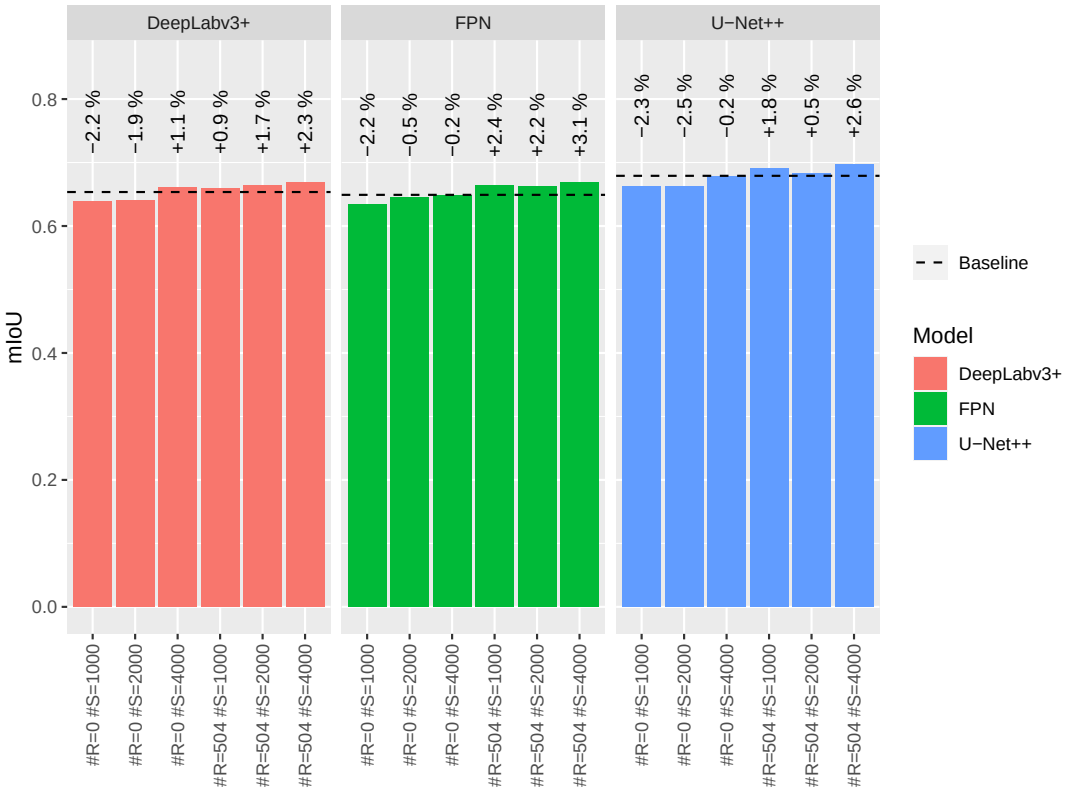


Figure 7.17: Comparison of Different Amounts of Synthetic Data on the Binary Blemish-Illness-Mould Dataset. The label below each bar indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The percentages refer to the improvements/deterioration compared to the baseline.

The results differed depending on the model architecture and configuration, with some surprisingly significant differences. For the configurations with synthetic data alone, it was noticeable that 1000 time steps worked better for all models when using few data (128 samples), but worse when more data (1000 samples) was used.

For the configurations where synthetic and real data were mixed, there were different behaviours depending on the model architecture: the DeepLabv3+ benefited more from the data sampled with 300 steps. The FPN and the U-Net++ benefited, in some cases significantly more, from the 1000 time steps.

In any case, this ablation showed once again that it is possible to improve different models for semantic segmentation with synthetic data and that the synthetic data itself is crucial. Depending on the semantic segmentation model, changes in synthetic data and the amount of data can have a different effect. This aligns with the paradigm shift mentioned in Section 1.1 because it indicates that the data generation really requires to be fit to the model.

Size of the Generated Datasets

For most of the experiments with the ISDM method presented so far, a synthetic dataset of 1000 image-mask pairs was created and configurations with different-sized subsets of this dataset were tested to determine the influence of the amount of synthetic data on the semantic segmentation models. The number 1000 was chosen mainly because it corresponds approximately with the cardinality of all tested datasets. However, for some of the larger datasets, it is also conceivable that better results could be achieved by using more synthetic data, which is why this design decision was ablated for the Binary Blemish-Illness-Mould dataset with twice and four times as much synthetic data respectively. The results of this ablation are shown in Figure 7.17.

Larger amounts of synthetic data helped especially when the semantic segmentation models were only trained on such data. The configuration #R=0 #S=4000 could keep up with the baseline trained only

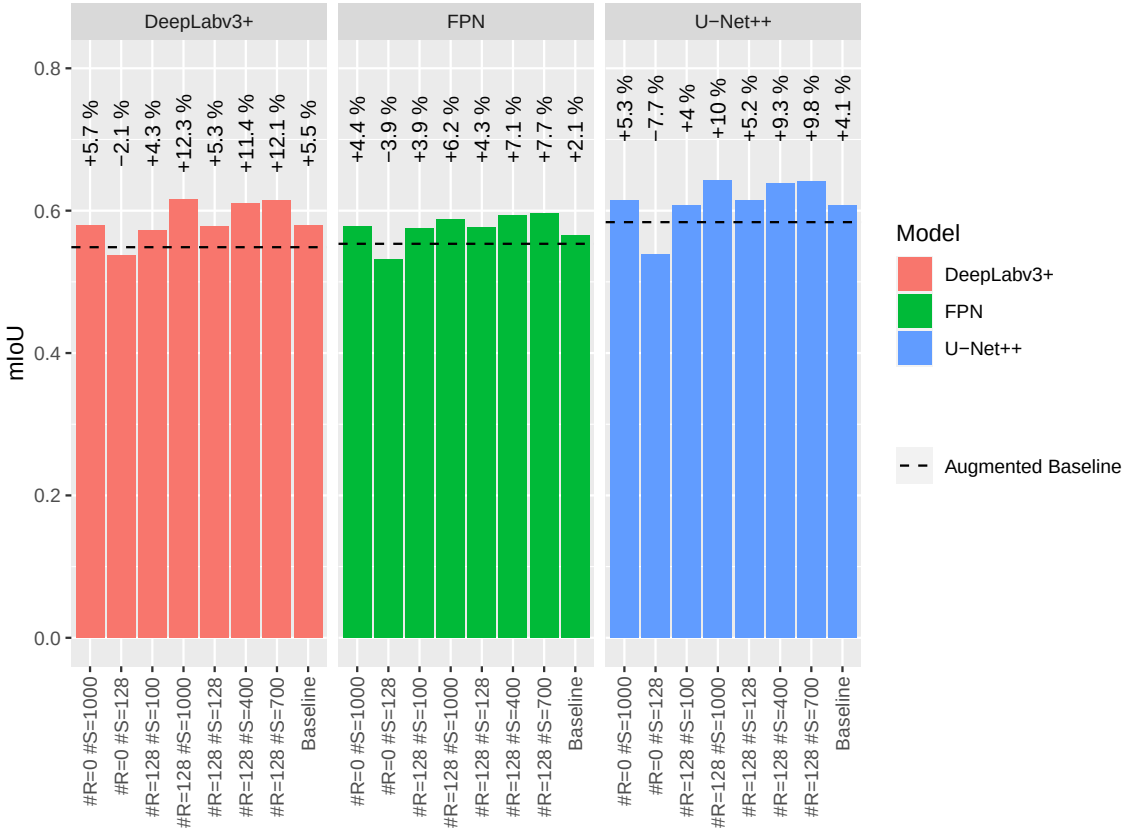


Figure 7.18: Comparison of Synthetic Data to (Traditional) Data Augmentation. The label below each bar indicates the number of real ("#R") and synthetic ("#S") image-mask pairs used to train the downstream segmentation model. The results correspond to the ones for the Binary Blemish Dataset, but are compared to a baseline trained with data augmentation. The baseline trained without data augmentation is also reported. The percentages refer to the improvements/deterioration compared to the augmented baseline.

on real data for all models. The original configuration with 1000 synthetic image-mask pairs had a deficit of approximately 2 % compared to the baseline for all models when using synthetic data alone.

The differences were smaller for the configurations with a mix of synthetic and real data. The DeepLabv3+ was the only model were the improvement seemed to correlate with the size of the synthetic dataset. The FPN and the U-Net++, on the other hand, performed worse for the configuration #R=504 #S=2000 than for #R=504 #S=1000. The configuration #R=504 #S=4000 proved to be the best for all models. Thus, all models could be improved to some extent with larger synthetic datasets.

Comparison to (Traditional) Data Augmentation

In this work, the use of synthetic data to augment real datasets was presented as a paradigm shift in deep learning, allowing the data to be adapted instead of the models. Another view would be to interpret generative models as a kind of "learned" data augmentation: instead of using predefined, well-known techniques for data augmentation, it is left to the generative model. From this point of view, it makes sense to compare the obtained results to traditional data augmentation. For this purpose, the semantic segmentation models were trained to the real data while using random affine transformations (shift, scale, rotate), random horizontal flip, random sharpening/blurring as well as random contrast and hue changes. These transformations were used as they were already contained in the implementation of Macháček et al. (2023) on which the validation proposed in this thesis was based. The results for the models trained with these transformations are referred as "augmented baseline". They were compared to the results for the experiments on the Binary Blemish Dataset from Section 7.3.1. This comparison is shown in Figure 7.18.

The results in the figure demonstrate that it is also possible to deteriorate the semantic segmentation

models, as the augmented baseline is lower than the baseline trained without data augmentation for all three models. This indicates that the application of data augmentation harmed the models, which was in general not the case for the experiments with synthetic data on this dataset. The only configuration with synthetic data that was lower than the augmented baseline and also the non-augmented baseline is #R=0 #S=128, which was expected since this configuration used no real and only low amounts of synthetic data. For all the other configurations the use of synthetic data worked significantly better than traditional data augmentation.

Chapter 8

Conclusion

In this work, the current trend around generative deep learning models was addressed. In the field of computer vision, this trend focuses primarily on DMs, which are already being used for different applications in the field of art. However, this work did not deal with the use of DMs for artistic purposes but rather addressed how they can be used to enhance existing datasets with synthetic data. This topic aligns with a recent paradigm shift in deep learning that was made possible through the advances in generative modelling: instead of fixing the dataset and attempting to improve the methods applied to the data, the methods are fixed, and the dataset is changed through generative models to better fit to the methods. In particular, this thesis focused on the use of DMs to enhance datasets for semantic segmentation with the ultimate goal of improving the performance of some widely used models for this task. The main difficulty when generating such datasets is that the labels in semantic segmentation consist of pixel-level annotated semantic masks. Therefore, it is not sufficient to create arbitrary images to compose a synthetic dataset, but the images have to match with corresponding semantic masks. To achieve this, there are certain strategies to control generative models so that they consider certain information when generating images. These techniques are also referred to as conditioning, which is why, the term conditional image synthesis is used for tasks that aim at generating images with conditioning techniques involved. In this thesis, DMs were chosen as generative models to create the synthetic datasets, because, although they are still underexplored compared to other generative models, they have already shown state-of-the-art results for unconditional and conditional image synthesis tasks.

The use of synthetic data for semantic segmentation tasks proved to be particularly interesting because there is a consensus in the literature that many of the available datasets are scarce. This has to do with the required pixel-level labelling, which is time-consuming and often requires domain expertise. The latter is especially true for the medical domain, where privacy issues with the available data pose additional difficulties. For these reasons, many researchers, such as Fernandez et al. (2022), Cechnicka et al. (2023), Han et al. (2023), Shao et al. (2023) and Macháček et al. (2023) focused on enhancing medical semantic segmentation datasets. While these medical datasets are often rather small and are mostly for binary segmentation with only two classes (background and anomaly), other researchers, including Wang et al. (2022) and Wu et al. (2023), have also already tested the impact of DMs on large benchmark datasets for semantic segmentation. These datasets differ not only in size from medical datasets, but they also serve a different purpose, namely the one of object and scene understanding. A thorough search of the relevant literature yielded that no approach was applied to industrial semantic segmentation datasets for anomaly detection. The reason for this is probably that only a few such datasets are publicly available. Therefore, the major contribution of this work was to test some of the approaches for enhancing semantic segmentation datasets on the industrial Lemons Quality Control Dataset for fruit quality control.

More specifically, the present work examined two possible approaches for enhancing this dataset through extensive experiments: semantic image synthesis and paired image mask synthesis. The first approach is better known in the literature. It can best be described as the opposite of semantic segmentation because the aim is to generate realistic images that fit the given semantic masks. The second approach has also already been used in the literature, but there is no common name for it yet, which is why it was simply described as paired image-mask synthesis in this thesis. The name also serves as a description for the task, because it is about generating both, images and matching semantic masks. The approach used in this work first generates the masks and then uses them to condition the image-

generating DMs. Two different methods were used to implement these two approaches, one based on the LDM implementation and the other on the SDM implementation. The former method, named All-Latent, was designed to support both approaches. It can either reuse the real semantic masks to condition an image-generating LDM, or use a second, unconditional LDM to generate semantic masks, which in turn are used for conditioning.

The results showed that both approaches for All-Latent could improve the downstream semantic segmentation models for binary datasets, but it was necessary to filter the generated data in postprocessing, otherwise, it could even deteriorate the performance of the models. Since the entire method proved to be very complex, it was decided to test a second method, which promised a better correspondence between the generated image and the semantic mask through the use of SDMs. This method was called Improved Semantic Diffusion Model (ISDM) because it improved some sub-optimal aspects of the SDM implementation, especially the employed checkpointing mechanism during training.

The improved correspondence between images and masks could be confirmed throughout the experiments for the SDMs. However, the images sampled with the ISDM method appeared noisier compared to the ones generated with All-Latent. Nevertheless, the second method was able to outperform the baseline as well as the All-Latent method without any filtering applied to the generated data. In most cases, the ISDM method improved all three semantic segmentation models, although the amounts of the improvements differed between model architectures and datasets and were often only marginal. The most important result of this thesis is that with the ISDM method, it was possible to create synthetic datasets on which some of the semantic segmentation models could achieve comparable performance as if they had been trained on the real data. This result could be particularly valuable for the medical domain and other areas where strict privacy rules apply. The ISDM method was also tested on a multi-class dataset, where for two of the tested models (FPN and U-Net++) mixed datasets could be found that achieved improvements for all classes. However, it was also observed that the improvements for the multi-class dataset and generally for larger datasets were lower.

In summary, the extensive experiments with the All-Latent and ISDM methods showed that it is generally possible to enhance datasets for semantic segmentation with synthetic data from DMs. This was possible even though the generated data itself revealed some obvious problems, which is a sign that more significant improvements are conceivable if these problems are solved. Some of these problems had to do with the DMs and their implementations themselves, as they did not seem to be fully developed yet. Therefore, many possibilities for future work remain:

- In the experiments with the All-Latent method, it was shown that filtering in postprocessing could improve the results significantly because poorly generated data had a negative impact. Furthermore, with the ISDM method, inconsistencies in sample quality were observed, even when sampled from the same SDM with the same conditioning information. Future work should aim to find out why these inconsistencies occur and/or find suitable methods and metrics for filtering to remove bad data afterwards or already during the sampling process.

- The training of the SDMs themselves proved to be inconsistent across different datasets too. This could be remedied by the proposed checkpointing implementation, but it was also explicitly pointed out that this is not an optimal solution. It would be worthwhile to research better strategies and methods for this as well. Probably the reason why this has not yet been done is that many methods are based on the implementations of Guided Diffusion and Improved Diffusion, which work well, but also have certain drawbacks. For example, the lack of tracking of epochs and validation metrics during the training process. These would have to be implemented for appropriate monitoring and checkpointing.

- The implementations mentioned above allow the adjustment of numerous hyperparameters, but their effect is not always considered in the literature. The best example of this is the “rescale time steps” parameter, which was most probably not even considered by the original SDM implementation but could be used to speed up the sampling process significantly, as noted in this work. For this reason, a hyperparameter search or a hyperparameter tuning on the SDM implementation, but also on DM implementations in general, would certainly be an interesting path to follow in the future.

- Another shortcoming that should be addressed in the future is that it would be important to enable sampling in the original resolution of the dataset natively.

These are only some of the possible improvements so that tasks like semantic image synthesis and paired image-mask synthesis can really lead to the aforementioned paradigm shift for semantic segmentation and become of proper value for enhancing or even replacing real datasets with synthetic ones.

Bibliography

Maciej Adamiak. Lemons quality control dataset, Jul 2020. URL `https://github.com/softwaremill/lemon-dataset`.

Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt, 2023.

Sarah Cechnicka, James Ball, Callum Arthurs, Candice Roufosse, and Bernhard Kainz. Realistic data enrichment for robust image segmentation in histopathology. *arXiv preprint arXiv:2304.09534*, 2023.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Marc Everingham, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

Virginia Fernandez, Walter Hugo Lopez Pinaya, Pedro Borges, Petru-Daniel Tudosiu, Mark S Graham, Tom Vercauteren, and M Jorge Cardoso. Can segmentation models be trained with fully synthetically generated data? In *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 79–90. Springer, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Kun Han, Yifeng Xiong, Chenyu You, Pooya Khosravi, Shanlin Sun, Xiangyi Yan, James Duncan, and Xiaohui Xie. Medgen3d: A deep generative framework for paired 3d image and mask generation. *arXiv preprint arXiv:2304.04106*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Pavel Iakubovskii. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2019.

Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021.

Muhammad Zubair Khan, Mohan Kumar Gajendran, Yugyung Lee, and Muazzam A Khan. Deep neural architectures for medical image semantic segmentation. *IEEE Access*, 9:83002–83024, 2021.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

Roman Macháček, Leila Mozaffari, Zahra Sepasdar, Sravanthi Parasa, Pål Halvorsen, Michael A Riegler, and Vajira Thambawita. Mask-conditioned latent diffusion for generating gastrointestinal polyp images. *arXiv preprint arXiv:2304.05233*, 2023.

Dominik Müller, Iñaki Soto-Rey, and Frank Kramer. Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes*, 15(1):1–8, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/nichol21a.html`.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019.

Shengxiang Qi, Jiarong Yang, and Zhenyi Zhong. A review on industrial surface defect detection based on deep learning technology. In *Proceedings of the International Conference on Machine Learning*, pages 24–30, 2020.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Shitong Shao, Xiaohan Yuan, Zhen Huang, Ziming Qiu, Shuai Wang, and Kevin Zhou. Diffuseexpand: Expanding dataset for 2d medical image segmentation using diffusion models. *arXiv preprint arXiv:2304.13416*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Vadim Sushko, Edgar Schönfeld, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. Oasis: only adversarial supervision for semantic image synthesis. *International Journal of Computer Vision*, 130(12):2903–2923, 2022.

Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models. *arXiv preprint arXiv:2207.00050*, 2022.

Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021. URL `https://lilianweng.github.io/posts/2021-07-11-diffusion-models/`.

Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. *arXiv preprint arXiv:2303.11681*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10145–10155, 2021.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 633–641, 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*, 2018a. URL `https://github.com/CSAILVision/semantic-segmentation-pytorch/blob/master/teaser/ADE_val_00001519.png`.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In Danail Stoyanov, Zeike Taylor, Gustavo Carneiro, Tanveer Syeda-Mahmood, Anne Martel, Lena Maier-Hein, João Manuel R.S. Tavares, Andrew Bradley, João Paulo Papa, Vasileios Belagiannis, Jacinto C. Nascimento, Zhi Lu, Sailesh Conjeti, Mehdi Moradi, Hayit Greenspan, and Anant Madabhushi, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11, Cham, 2018b. Springer International Publishing. ISBN 978-3-030-00889-5.