university of groningen / faculty of science and engineering / artificial intelligence

Label-efficient segmentation of organoid culture data using diffusion models

Bachelor's Project Thesis

Jesse Wiers, s3998576, j.l.wiers@student.rug.nl,
Daily Supervisor: Asmaa Haja, a.haja@rug.nl
Main Supervisor: Prof. Lambert Schomaker, l.r.b.schomaker@rug.nl

Abstract: As the volume of biomedical data continues to expand at a rapid pace, the potential of extracting valuable insights from this data through deep learning models is also increasing. However, this process typically necessitates labeled data, which has traditionally been manually annotated. This manual approach is associated with various constraints, including time, financial resources, and expertise, and it can also be prone to errors due to fatigue. The objective of this study is to utilize diffusion models, specifically diffusion denoising probabilistic models (DDPMs), for the segmentation of organoid culture data. Two different methods are employed using DPPMs. Firstly, segmentation will be carried out by utilizing feature maps of DDPMs that have been trained to generate samples of organoid culture data. These feature maps will be combined with an ensemble of multi-layer perceptrons. Secondly, DDPMs will be trained to directly generate segmentation maps for organoid culture data. The methods were evaluated on the MIoU, Dice and HD95 score on a maximum of 42.348 images. On 100% of the data, the representation approach (MIoU=0.92, Dice=0.96, HD95=35) outperformed the direct segmentation approach (MIoU=0.62, Dice=0.71, HD95=62) for all metrics. The representation approach also proved to be suitable for label-efficient segmentation since the aforementioned performance for the representation approach is achieved with as little as 20 labelled images in the training pipeline.

1 Introduction

High-throughput imaging technologies enable the rapid creation of microscopic images (Pegoraro & Misteli, 2017). These technologies use automated microscopy and analysis, allowing researchers to collect data on large sample sets. The vast volume of data has revolutionized biological studies, revealing detailed components of cellular and molecular systems (Pegoraro & Misteli, 2017). Valuable insights include identifying cellular structures and organelles (Yudistira et al., 2020), quantifying protein expression and localization (Crowe & Yue, 2019), and discovering new biological phenomena (Zeune et al., 2020).

Manual analysis of high-dimensional data in microscopy is challenging due to the need for expertise, potential bias, time consumption, and fatigue (Zhu et al., 2021; Adhikari et al., 2021). Deep learning offers an alternative by discovering complex patterns in data. Moreover, it has been successful in various domains, including microscopy (Dargan et al., 2020). However, deep learning relies on large amounts of annotated data. Manual annotation of data leads to similar challenges as manual analysis, leaving much microscopy data unlabeled due to these constraints (Chakraborty & Mali, 2023). Nevertheless, large amounts of intrinsic information can be learned from unannotated data. Several unsupervised deep learning techniques such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs) and Deep Belief Networks (DBNs) have been shown to be successful in learning data representations (Wei & Mahmood, 2021; Goodfellow et al., 2020; Roder et al., 2021).

Another model that can be used for learning data representations is the Diffusion Denoising Probabilistic Model (DDPM). First discovered by Sohl-Dickstein et al. (2015) and inspired by non-equilibrium statistical physics. DDPMs, a type of diffusion model, are able to learn data distributions by iteratively noising data and learning to reverse this process. Recently several works have shown that by making adjustments or extending DDPMs, they can also successfully be utilized for (label-efficient) segmentation of images (Amit et al., 2021; Baranchuk et al., 2021).

Organoid culture data is one type of biomedical data that can lead to insightful insight through the employment of deep learning models (Drost & Clevers, 2018; Fatehullah et al., 2016). Organoid culture data refers to the collection of information, results, and observations derived from the study of organoids in a controlled environment (de Souza, 2018). Organoids are miniature, self-organized, three-dimensional tissue cultures derived from stem cells. This paper attempts to utilize DDPMs to effectively segment organoid culture images. This allows for the identification of organoids, which is useful for further analysis using deep learning models. Currently, still little research has been performed on using DDPMs for segmentation. Moreover, existing organoid segmentation methods have several limitations such as relying on large annotated datasets (Borten et al., 2018; Matthews et al., 2022; Powell et al., 2022). For these reasons, this work aims to further investigate the usefulness of DDPMs for segmentation, specifically of organoid culture data. This work specifically analyses the performance of two different methods utilizing DDPMs parameterized by the U-Net architecture. One in which a DDPM is used as a representation learner after which the learned representations are used as input to an ensemble of multi-layer perceptrons (MLPs) to perform segmentation. In the other method, a DDPM is trained to directly create segmentation maps by utilizing the ground truth images.

This study attempts to investigate to what extent the two methods can serve as effective methods for (label-efficient) segmentation of organoid culture data by answering six research questions. The following four research questions are investigated with respect to the method where a DDPM is used as a representation learner:

- How does the size of the dataset used to train the diffusion model impact the quality of the segmentation result?
- How does the size of the dataset used to train the ensemble of MLPs influence the quality of the segmentation result?
- How do the specific blocks used from the U-Net impact the quality of the segmentation result?
- How does the amount of noise added during the forward diffusion process affect the performance of the segmentation result?

Moreover, the following two research questions are investigated with respect to the method where a DDPM is used to directly create segmentation maps:

- How does the size of the dataset used to train the diffusion model impact the quality of the segmentation result?
- How does the number of samples generated by the diffusion model affect the performance of the segmentation?

This work is organized into 6 sections, with the following structure: Section 2 presents a review of the related works, providing a deeper insight into organoid culture data, semantic segmentation, the U-Net architecture, diffusion models and semantic segmentation performed by generative models. Section 3 describes the method of the investigation, where the two different ways of using DDPMs to perform segmentation are explained, the utilized hyperparameters are discussed and the performance metrics are explained. Section 4 is reserved for the experimental design, describing the data distribution and the manner in which the research questions have been investigated. Section 5 provides an analysis of the results. Lastly, section 6 encompasses the conclusion and future work.

2 Related Work and Definition

2.1 Organoid Culture Data

Growing three-dimensional structures termed organoids from stem cells or tissue samples is done in the lab using the process known as organoid culture (de Souza, 2018). In organoid culture, the cells are given the essential nutrition, growth factors, and physical conditions to enable self-organization and development into miniature organ replicas. Organoid culture data has many important uses in cell research, such as modelling human diseases in a laboratory setting (Drost & Clevers, 2018; Fatehullah et al., 2016), providing insights into the fundamental processes that govern organ development and maturation (Huch & Koo, 2015), and guiding the development of tissue engineering approaches (Takebe et al., 2014). In order to analyse the organoids, precise measurements of their morphology are required, which is done by segmenting the organoid objects.

2.2 Semantic Segmentation

In the deep learning task of semantic segmentation, each pixel of an image is given a meaningful label, such as an object category or a scene component. Semantic segmentation is frequently carried out using deep convolutional neural networks (CNNs) and has applications in a variety of industries such as healthcare, autonomous driving and robotics (Sharifani & Amini, 2023). These networks learn features from the input images using convolutional and pooling layers and use fully connected layers to make predictions. Once trained, the CNN can be used for inference on new images to create a semantic segmentation map, where each pixel is given a class label based on the features that are learned. In the context of organoid semantic segmentation, this would be either 0 or 1, corresponding to the background or the organoid itself, respectively.

Multiple software tools have been developed specifically for organoid culture segmentation, such as OrganoSeg (Borten et al., 2018), OrganoID (Matthews et al., 2022), and deepOrganoid (Powell et al., 2022). However, these tools have certain limitations that need to be considered.

Firstly, OrganoSeg provides a user-friendly graphical interface but requires manual thresholding and parameter tuning. This can be time-consuming and subjective, as optimal values may vary depending on the specific organoid culture and image characteristics. The accuracy of the segmentation may also be affected by variations in image quality, illumination, and contrast.

Secondly, OrganoID uses deep learning techniques for single organoid detection, but it may be sensitive to changes in image quality, such as variations in brightness or contrast. This can result in inaccurate segmentations or missed organoids, especially when dealing with diverse organoid cultures or images with low contrast.

Thirdly, the deepOrganoid model is a deep learning-based tool that can be used for high-throughput screens. More specifically, the model is designed to handle a large number of organoid samples or images in a fast and efficient manner. However, it requires a sufficiently large labeled dataset for training. Obtaining a large annotated dataset for a specific organoid culture may be challenging, as labeling organoids accurately can be time-consuming and labor-intensive. This limitation may restrict the applicability of the model to datasets with limited annotations.

To conclude, all methods carry several limitations with the main concern being the strong reliance on supervised learning of all methods, which requires sufficient labeled data for training. This can be a limitation in the organoid field, as obtaining a large annotated dataset for diverse organoid cultures may be challenging. Moreover, the lack of labeled data may limit the ability of these tools to generalize well to different organoid culture datasets.

2.3 U-Net Architecture

For image segmentation tasks, notably in the study of microscopy image analysis, the U-Net architecture is a neural network architecture that is often utilized. First introduced by Ronneberger et al. (2015), the architecture proved to be ef-

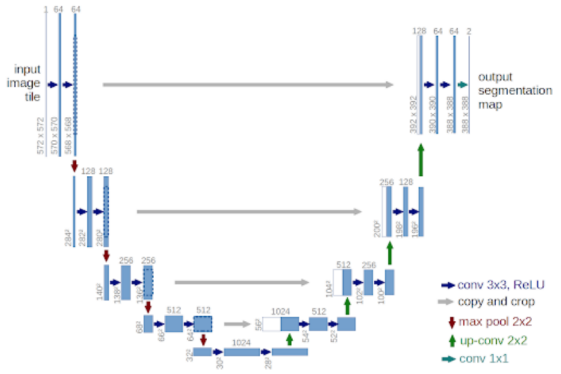

Figure 2.1: A visual representation of the U-Net architecture. Source: Ronneberger et al. (2015).

fective in segmenting images of cells, nuclei, and other structures. The intuition behind the U-Net architecture is to enable the network to learn high-level features from the input image while also preserving spatial information.

The U-Net architecture consists of two paths. The contracting path, also referred to as the encoder, employs a sequence of convolutional and pooling procedures to decrease the spatial resolution of the feature maps while increasing the number of feature channels in order to extract high-level features and extract context from the input picture. The contracting path's high-level characteristics are used by the expanding path, sometimes referred to as the decoder, to reconstruct the output segmentation map. A dense pixel-wise map the same size as the input picture is finally produced as an output. This output is produced by the decoder through a sequence of transposed convolutions and concatenation operations that incrementally raise the spatial resolution of the feature maps while reducing the number of feature channels. A visual representation of the U-Net architecture can be seen in Figure 2.1.

2.4 Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015), a subset of generative models, approximate the distribution of real images. The specific diffusion technique explained in this section will be the Denoising Diffusion Probabilistic Model (DDPM), however, the terms DDPM and diffusion model will be used interchangeably, since in the literature diffusion models usually refer to DDPMs. Essentially, diffusion models operate by modifying training data by the sequential addition of Gaussian noise, whereafter the model learns to recover the original data by reversing this process. Following the training process, diffusion models generate data by gradually transitioning a simple known distribution $x_T \sim N(0, I)$ into a target distribution x_0 via an iterative denoising procedure. Here I refers to the identity matrix and N refers to the normal distribution. A deep neural network effectively learns to reverse the diffusion process using a known Gaussian kernel to describe each Markov step. For a certain image x at step t, the forward noising process q is given by:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{2.1}$$

where $\beta_1, ..., \beta_T$ denotes a fixed variance schedule and T denotes the final noising step of the diffusion process. Importantly, a noisy sample x_t can be derived straight from the data x_0:

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)I), \tag{2.2}$$

where $\alpha_t := 1 - \beta_t$ and $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. With the reparametrization trick, x_t can be directly written as a function of x_0:

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, I). \tag{2.3}$$

The reverse process p_θ learned by the model parameters θ is given by:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \sum_\theta(x_t, t)\right) \tag{2.4}$$

Where $\mu_\theta(x_t, t)$ and $\sum_\theta(x_t, t)$ refer to the mean predictor and covariance predictor respectively. As shown by Ho et al. (2020), x_{t-1} can then be predicted from x_t with:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z, \quad z \sim \mathcal{N}(0, I) \tag{2.5}$$

where $\epsilon_\theta(x_t, t)$ refers to the noise predictor and σ_t denotes the variance scheme that can be learned by the covariance predictor $\sum(x_t, t)$, as opposed to

applying a predetermined sequence of scalar covariances. Learning the covariances has been demonstrated to enhance the model's quality (Nichol & Dhariwal, 2021). Instead of predicting the mean of the distribution in Equation (2.4), in reality, the noise predictor network $\epsilon_\theta(x_t, t)$ predicts the noise component at step t; a linear combination of x_t and this noise component then forms the mean. The training objective of the model amounts to maximizing the log-likelihood of the generated sample x_t belonging to the original data distribution. To achieve this objective, the following variational lower-bound loss is utilized:

$$L_{vlb} := D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) \quad (2.6)$$

Here q and p_θ refer to the posterior and prior respectively. Additionally, D_{KL} refers to the KL-Divergence, a measure of dissimilarity between two probability distributions. During training, the goal of the deep-learning model is to approximate the parameters of the posteriors such that the KL divergence is minimal. The publications by Ho et al. (2020) and Nichol & Dhariwal (2021) include the whole derivations of the formulas above.

Several variations of the U-Net architecture are commonly used to parameterize the denoising model $\epsilon_\theta(x_t, t)$ (Ronneberger et al., 2015). The state-of-the-art model architecture proposed by Dhariwal & Nichol (2021) is used to parameterize the denoising model in this work.

2.5 Semantic Segmentation with Generative Models

While there is currently significant research in the area of generative models for image segmentation, the majority of approaches that have been investigated are GAN-based. The first line of research (Voynov & Babenko, 2020; Voynov et al., 2021; Melas-Kyriazi et al., 2021) is based on the discovery that the latent spaces of state-of-the-art GANs contain directions that can selectively influence the foreground and background pixels in synthesized images. This has paved the way for the development of techniques that utilize synthetic data generated by GANs to train segmentation models. However, because of the ability to distinguish between foreground-background pixels these lines of work were especially useful to perform binary segmentation. A Second line of works focused on intermediate representations obtained in GANs, which were shown to allow for multi-class segmentation (Zhang et al., 2021; Tritrong et al., 2021; Xu & Zheng, 2021; Galeev et al., 2021). Inspired by this line of works, Baranchuk et al. (2021) investigated the intermediate representations obtained by DDPMs and used those to train an ensemble of neural networks for semantic segmentation. This minimized the need for extensive labelling. The setup of Baranchuk et al. (2021) is used to assess label-efficient organoid segmentation using a DDPM as a representation learner in combination with an ensemble of neural networks. Simultaneously, Wolleb et al. (2022) and Amit et al. (2021) investigated the use of diffusion models for image segmentation by using a different type of method. Instead of training a diffusion model to generate images from the original domain, they trained diffusion models to generate segmentation maps directly. The setup from Wolleb et al. (2022) is utilized to assess organoid segmentation using a DDPM to directly create segmentation maps given input images.

Looking at the most recent line of works. Rahman et al. (2023) proposed a diffusion model for ambiguous medical image segmentation, such that it can segment images with multiple possible interpretations. Moreover, Hu et al. (2023), also facilitate weakly supervised segmentation with higher efficiency using a conditional diffusion model with guidance from an external classifier, allowing models to be trained with limited labelled data. Lastly, Laousy et al. (2023) proposed a diffusion model that combines diffusion models with randomized smoothing to produce segmentation masks that are more robust to adversarial attacks. These varied applications underscore the substantial contributions and potential of diffusion models in image segmentation tasks.

3 Method

Ultimately, this work aims to utilize two different approaches of using DDPMs for the segmentation of organoid culture data. In this section, both methods and their respective implementations will be

explained. First, the data on which the experiments are performed will be discussed in section 3.1. Secondly, how DPPMs are utilized as representation learners in conjunction with an ensemble of neural networks will be explained in section 3.2. Thereafter, how DPPMs are used to directly produce segmentation maps of organoid culture data will be explained in section 3.3.

3.1 Data and Augmentation

The dataset utilized for this experiment comprised of liver progenitor organoids obtained from the University Medical Center Groningen (UMCG) in the Netherlands. The organoid images were captured using a specialized microscope at five different time points, spanning from 0 to 96 hours with 24-hour intervals. Two growing conditions were considered: (1) organoids grown in a complete medium and (2) organoids grown in a medium lacking essential amino acids required for their growth. As a result, a total of 10 CZI images were obtained. A CZI image refers to a 3D representation consisting of 2D image slices captured at various depths within the organoid culture (Figure 3.1).

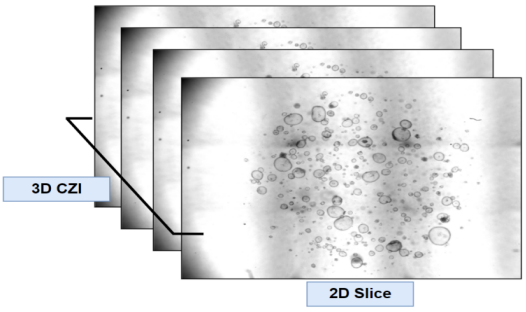

Figure 3.1: A CZI representation: a 3D image that consists of a stack of 2D slices taken at various depths within the organoid culture. Source: Brouwer (2022).

Each CZI file contains 14 2D slices, with each slice having an image size of 3828x2870 pixels. Since the upper and lower slices contain limited relevant information, an average of 4 middle slices is used. The organoid images underwent semantic segmentation using the OrganelX service*, with manual correction performed to ensure accurate segmentation.

Given the high resolution of the initial images (3828x2870 pixels), which is computationally demanding for deep learning networks, a sliding window technique is employed to create smaller image sections called crops (Figure 3.2). Crops of

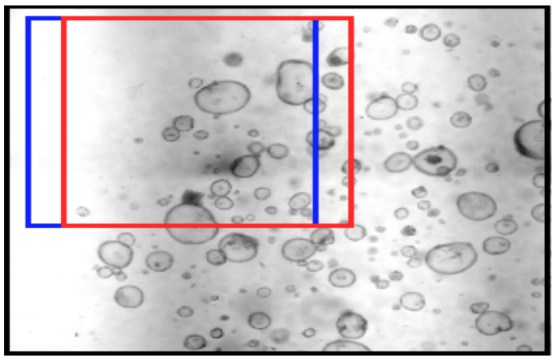

Figure 3.2: Sliding window crop: a window, represented by a blue square, is chosen and used to create a crop of the image. The window is then shifted by a certain number of pixels, and a new crop is generated within a red square. This process is repeated, sliding the window across the entire image and producing crops from different parts of the image. As a result, the entire image is covered, and crops are obtained from all regions of the image. Source: Brouwer (2022).

size 636x636 pixels are generated with a window increment of 60 pixels per step. These cropped images are subsequently resized to 256x256 pixels to expedite model training time for all experiments. Crops containing less than 5% relevant information (presence of organoids) are removed from the dataset. To augment the dataset and increase its diversity, image rotation is applied as an augmentation technique. This process results in approximately 100,000 cropped and augmented images, of which subsets are used to train the diffusion models in all the experiments with the aim of investigating label-efficient segmentation. More information about the exact data distributions used for the experiments can be found in sections 4.1 and 4.2.

*https://organelx.hpc.rug.nl/organoid/

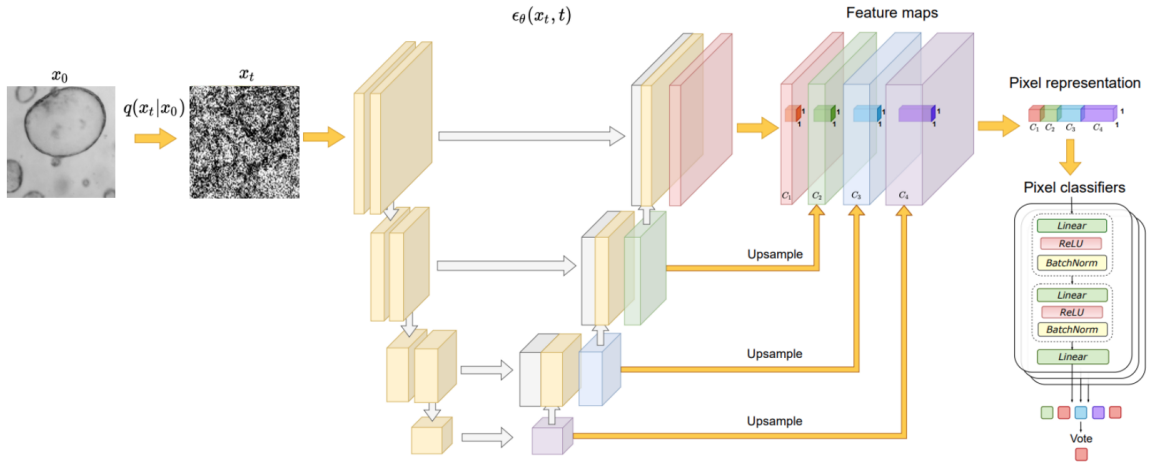

Figure 3.3: Overview of the representation learning method. In this method, noise is added to an image for t timesteps. The U-Net then predicts the image for timestep $t-1$. The feature maps of the expanding path used for this process are then concatenated and upsampled to the image size. Finally, the pixels of these concatenated feature maps are then fed into an ensemble of MLPs to predict the labels of these pixels. Source: Baranchuk et al. (2021).

3.2 DDPM as Representation Learner

For the method where representations are utilized, the approach brought up by Baranchuk et al. (2021) has been implemented. In this method, a DDPM parameterized by the U-Net architecture is learned to produce organoid culture data after which an ensemble of neural networks is trained on learned feature maps to classify pixels by utilizing majority voting to decide the pixel labels. A visual representation of this method can be seen in Figure 3.3. The method consists of the following steps:

1. Noise is added to x_0 according to $q(x_t|x_0)$.
2. The feature maps of the noise predictor $\theta(x_t, t)$ used for the reverse diffusion process are extracted.
3. The feature maps are upsampled to the picture resolution and concatenated to gather pixel-level representations.
4. The pixel-wise feature vectors are used to train an ensemble of MLPs to determine the class that each pixel belongs to.

The loss function used to train the diffusion model is the variational lower-bound loss which was explained in Section 2.5. Moreover, the loss function used to train the ensemble of MLPs is the binary cross-entropy loss which can be seen in equation 3.1.

$$L = -y \log(p) - (1-y) \log(1-p) \tag{3.1}$$

In this formula, y is the true label and p is the predicted probability for class 1 (organoid). Binary cross-entropy loss is particularly effective for class-imbalanced segmentation problems since it heavily penalizes confident incorrect predictions and hence, naturally encourages the model to adjust predictions closer to the actual class distributions. This makes it a good fit for the problem at hand since the organoids are underrepresented in the images.

To parameterize the diffusion and denoising model, the state-of-the-art model suggested by Dhariwal & Nichol (2021) is utilized. In this section, when talking about hyperparameters, a distinction is made between the diffusion model and the denoising model. With the term diffusion

model, all hyperparameters associated with the diffusion process are referred to, including the noise schedule (e.g., linear, cosine), the maximum number of noise steps (1000 in all experiments), and the schedule sampler. The schedule sampler refers to whether all timesteps are sampled from e.g. a uniform distribution during training or whether another method is used for sampling, such as giving priority to timesteps with a high loss. In this research, the former is used. The hyperparameters associated with the U-Net, such as the attention resolutions or the dropout rate, are referred to as the denoising model.

Specifically, the same hyperparameters are used as Dhariwal & Nichol (2021) used for training their model on the LSUN bedroom dataset (Yu et al., 2015), with which they showed high performance in segmenting a multi-class dataset. The hyperparameters for the denoising model can be seen in Table 3.1. The hyperparameters shown in

Table 3.1: Selection of denoising model hyperparameters. The hyperparameters are based on: Dhariwal & Nichol (2021).

Attention Resolutions	32x32,16x16,8x8
Dropout	0.1
Learn Sigma	True
Number of Channels	256
Number of Head Channels	64
Number of Resolution Blocks	2
Resolution Blocks on Both Paths	True
Using Scale Shift Norm	True

the Table are not all hyperparameters but those which showed the highest impact on performance in Baranchuk et al. (2021) their work. Some of these hyperparameters, which might be unfamiliar, are briefly explained in the following paragraphs. The full list of hyperparameters can be found in Table A.1 and Table A.2 in the appendix.

The hyperparameter *Learn Sigma* refers to learning the covariance of the reverse process as opposed to applying a predetermined sequence of scalar covariances. Learning sigma has been demonstrated to enhance the model's quality (Nichol & Dhariwal, 2021). The hyperparameter *Using Scale Shift Norm* changes the manner in which the temporal information (time step) is added to the image features. Instead of adding the embedded temporal representations to the image features, two chunks are created from the temporal representations of which one is used to scale the image and the other is added to the image, which has been shown to improve performance (Nichol & Dhariwal, 2021). Some of the hyperparameters for the diffusion model can be seen in Table 3.2. The

Table 3.2: Selection of diffusion model hyperparameters. The hyperparameters are based on: Dhariwal & Nichol (2021).

Diffusion Steps	1000
Noise Schedule	Linear
Use KL	True
Predict x_start	False

Noise Schedule parameter refers to the manner in which the betas for the forward process are sampled. The betas in the experiments will be sampled from a linear function. Another commonly used function to sample the betas from is a cosine function which causes the images to be noised more gradually. Nevertheless, Nichol & Dhariwal (2021) showed greater performance with the less gradual noising that the linear beta schedule provided. For the ensemble of MLPs, the architecture from Zhang et al. (2021) is used, which they used to train a model to label images generated by Generative Adversarial Networks (GANs). The ensemble of MLPs consists of 6 independent models. These models consist of two hidden layers with ReLU nonlinearity and batch normalization. The sizes of the hidden layers are 128 and 32.

3.3 DDPM for Direct Segmentation

For the second method, the implementation of Wolleb et al. (2022) is used. In this method, a DDPM parameterized by the U-Net architecture learns to directly create segmentation maps. This is done by training the DDPM on the segmentation maps and concatenating the original images (the actual organoid images) at each step of the reverse diffusion process. A visual representation of this method can be seen in Figure 3.4. By

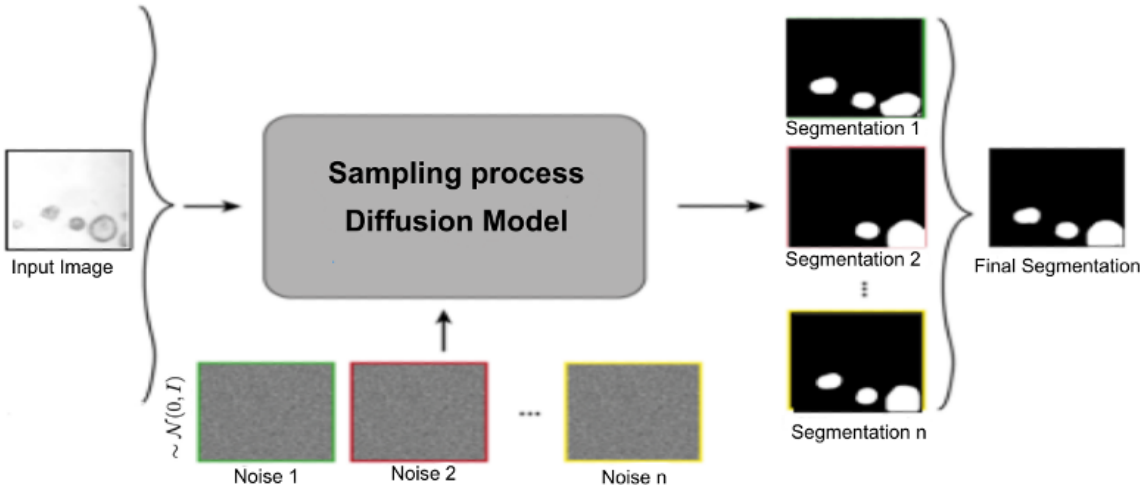

Figure 3.4: Overview of the direct segmentation method. In this method, a diffusion model is trained to produce segmentation maps of organoid culture data. To ensure that the segmentation maps are not random segmentation maps but segmentation maps of the input organoid image. The organoid image is concatenated during each step of the reverse diffusion process. By sampling several segmentation maps and averaging over the produced samples, the quality of the segmentation is expected to increase. Source: Wolleb et al. (2022)

concatenating the images at each step of the reverse diffusion processes the model does not produce any random segmentation map but a segmentation map tailored towards the input image since the anatomical information is induced at every step. Since DDPMs are stochastic models, the segmentation map that is generated for a given input varies each time the model produces a sample. For this reason, the model is used to generate several samples, which are then averaged to provide a more accurate segmentation. The loss function used to train the diffusion model is the variational lower-bound loss which was explained in Section 2.5.

For the U-Net architecture, the same architecture and hyperparameters are used as those used for the method where a DDPM is used as a representation learner, which can be seen in Table 3.1 and Table 3.2. Moreover, the number of segmentation maps created for each sample n is set to $n = 3$. However, an experiment to investigate different sample sizes is also performed. All of The models' training, validation, and testing were completed on a single NVIDIA V100 GPU node.

4 Experimental Design

In this work, as explained in Section 1, two sets of experiments will be performed to evaluate the two methods of using diffusion models for segmentation. The aim of these experiments is firstly to evaluate the performance of a DPPM used as a representation learner in combination with an ensemble of neural networks in its segmentation performance. Secondly, the aim is to evaluate the performance in segmenting organoid culture data by using a DDPM being trained to directly produce segmentation maps. In this section, the two sets of experiments are explained. In section 4.1, the experimental designs for the method where a DDPM as a representation learner is used are explained. In section 4.2, the experimental designs for the method where a DDPM is used for direct segmentation are explained.

4.1 DDPM as Representation Learner

The data distribution used to investigate the DDPM as a representation learner can be found

in Figure 4.1. The numbers of images in the blue blocks of the Figure: (42.348, 20, 20, 20), refer to the total amount of images used in the entire deep learning pipeline (training, validating, testing). However, dependent on the different experiments, different percentages of the data in each stage are used. Observing the upper blocks of Figure

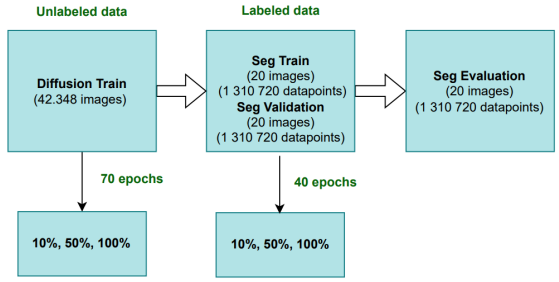

Figure 4.1: Experimental Design: Representation Learner

4.1 and moving from the left to right, it can be observed that the first block (Diffusion Train) contains 42348 images. This number refers to the number of images used to train the diffusion model. All of these images are unlabeled since training the diffusion model requires the actual organoid images and not the organoid masks. The number 42348 is chosen such that it is high enough to have the diffusion model be able to learn the data distribution. Since training the other method requires these images to be labelled for training and the aim is to use as limited labelled images as possible, a higher number is not utilized. Moving on to the next block (Seg Train and Seg Validation), it can be observed that the block contains 20 images twice, 20 images are used for training and 20 images are used for validating the diffusion model. This may seem like a low number, however, the ensembles are meant to classify pixels based on the features of these pixels. To put it, each image fed to the pixel classifiers equals 256x256 (image resolution) samples. The number 1310720 (256x256x20) that can be seen in the block refers to the total amount of pixels used to train and validate the segmentation model. To put it differently, the method can be investigated as a means of performing few-shot image segmentation, as only little annotated data is used to train the model. To train and validate the model, labelled data is used since the predicted segmentations are compared to the ground truth segmentations. Lastly, moving to the block completely at the right (Seg Evaluation), it can be observed that this block contains 20 images, which is the total amount of images used to evaluate the entire model. Again, the number 1310720 refers to the total amount of pixels used, this time for the evaluation.

Moving to the lower blocks and starting from the left, it can be observed that the block contains the numbers: [10%,50%,100%]. These numbers refer to the different percentages of data used to train the diffusion models in the different experiments. In the training stage, the diffusion models are trained for a total of 70 epochs. Moving one block to the right, again a block can be observed containing the values: [10%,50%,100%]. These numbers refer to the different percentages of data used to train and validate the segmentation models in the different experiments. The ensemble of MLPs is trained for 40 epochs. The number of epochs for training the segmentation part of the method is lower in practice since early stopping is performed, which means that the models only train for 3 to 4 epochs.

In the following subsections, the different experiments are briefly explained and the corresponding amount of data used is explained.

4.1.1 The Effect of the Blocks and Timesteps

The first experiment attempts to investigate how the specific blocks used from the U-Net impact the quality of the segmentation result and how the amount of noise added during the forward diffusion process/timestep affects the performance of the segmentation result. For this experiment, 100% of the diffusion training data is used and 100% of the segmentation training data. To test the effect, the following blocks are used from the 18 decoder blocks: [2,4,6,8,10,12,14,16,18], where the decoder blocks are numbered from deep to shallow blocks in the U-Net, such that a higher number refers to a higher resolution feature map. The decoder blocks are used since they also aggregate the information from the encoder blocks because of the skip connections between the en-

coder and decoder path of the U-Net. Moreover, the following timesteps are investigated: [25,50,75,100,200,300,400,500,600,700,800,900,925, 950,975]. These timesteps refer to the timesteps of the reverse diffusion process, such that t=0 refers to an image composed of entirely random Gaussian noise and t=1000 refers to a fully denoised image. For these timesteps and blocks, the high to low-level features of the fully noised and fully denoised images and most gradations in between are captured, to test their effectiveness in the segmentation.

4.1.2 The Effect of the Training Size

To test the effect of the training dataset size, the diffusion model is trained on 100%, 50% and 10% of the diffusion training data. Moreover, to test the effect of the training dataset size the segmentation model is trained on, 100%, 50% and 10% of the diffusion training data is used. Moreover, the blocks: [6,8,10] and the timesteps: [850,900,950] are used to obtain the feature maps in this experiment. These blocks and timesteps are chosen as they were found to be the most informative in the blocks/timesteps experiment. See section 5.1.1 for more details.

4.2 DDPM for Direct Segmentation

The data distribution used to investigate the DDPM used for direct segmentation can be found in Figure 4.2. Dependent on the different experiments, different percentages of the data in each stage are used.

Observing the upper blocks of Figure 4.2 and moving from left to right, it can be observed that the first block (Diffusion Train) contains 42368 images. These images refer to the number of images used to train the diffusion model. The number of images used to train the diffusion model for this method is the same as the total number of training images used for the representation method. For this method, however, the annotations of the images are required for the entire training process, since the diffusion model needs to learn how to sample segmentation maps instead of actual organoids. Moving to the next block (Seg Evaluation), it can be observed that this block contains 1000 images. This number refers to the number of images used for the evaluation of the diffusion model. Since this

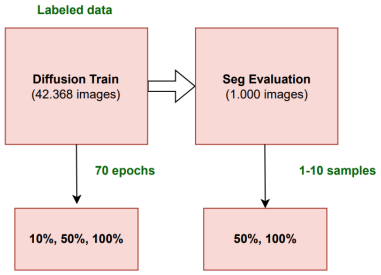

Figure 4.2: Experimental Design: Direct Segmentation

method is highly time-consuming while sampling a single image can take up to 80 seconds, using more images for the evaluation is not feasible. Moving to the lower blocks and starting from the left, it can be observed that the block contains the numbers: [10%,50%,100%]. These numbers refer to the different percentages of data used to train the diffusion models in the different experiments. In the training stage, the diffusion models are trained for a total of 70 epochs. Moving one block to the right, again a block can be observed containing the values: [50%,100%]. These numbers refer to the different percentages of data used to evaluate the diffusion model in the different experiments. Moreover, 1 up to 10 samples are used in the evaluation to produce a final segmentation map. This number differs dependent on the specific experiment. In the following subsections, the different experiments are briefly explained and the corresponding amount of data used is explained.

4.2.1 The Effect of the Ensemble Size

All of the diffusion training data is used to examine the impact of the number of samples produced on the final segmentation. However, only 50% of the total data is used for the evaluation. This is due to the fact that sampling takes a long time, and adding more samples to the evaluation prolongs the process even more. The evaluation is performed with 1 up to 10 samples. Moreover, the diffusion model underwent 70 epochs of training.

4.2.2 The Effect of the Training Size

To test the effect of the training data size, 100%, 50% and 10% of the training data is used. More-

over, the entire evaluation dataset is used in this experiment. The number of samples generated in this experiment is n=3, ensuring better performance compared to using a single sample but not increasing evaluation time significantly.

4.3 Metrics

Three prevalent metrics for the binary semantic segmentation task—the mean intersection over union (MIoU), the Dice coefficient, and the Hausdorff distance—are used in the quantitative analysis of the experimental results. These measurements are chosen because of their diversity and their applicability to the data.

4.3.1 MIoU

The predicted and actual segmentation masks' overlap is evaluated using the intersection over union (IoU), commonly referred to as the mean intersection over union (MIoU), since intersections are computed for different classes (in this work organoids and background). It calculates the intersection area to union area ratio of the two masks. By calculating the IoU, a thorough knowledge of how well the model represents the target regions of interest is gained. The equation for the MIoU is presented in equation 4.1.

$$\text{MIoU} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{4.1}$$

The total number of classes is represented by N, the number of true positive predictions for class i is represented by TP_i, the number of false positive predictions for class i is represented by FP_i and the number of false negative predictions for class i is represented by FN_i. The average intersection over union for all classes is determined by the MIoU.

4.3.2 Dice

Another key metric used for binary segmentation evaluations is the Dice coefficient, often known as the Dice similarity coefficient. It quantifies the similarity between the predicted and the ground truth masks, similar to the IoU. The Dice coefficient calculates the proportion of the areas of both masks added together to twice the junction area. In situations where there are class imbalances (which is the case for organoid culture data), it provides a clear indicator of segmentation accuracy.

$$\text{Dice} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4.2}$$

The equation for the Dice score is presented in equation 4.2. In this equation, TP stands for the total number of accurate positive predictions, FP for false positive predictions, and FN for false negative predictions. The similarity or overlap between the predicted and actual segmentation masks is determined by the Dice score.

4.3.3 Hausdorff Distance

The Hausdorff distance is another metric used to measure the dissimilarity between two sets of points, which in this case represents the predicted and ground truth masks. It calculates the maximum separation between any two points in one set and their nearest neighbours in the other set. By integrating the Hausdorff distance, an understanding of the model's capability to capture spatial information and accurately designate object boundaries is gained. The equation for the Hausdorff distance is presented in equation 4.3.

$$\text{Hausdorff}(A, B) = \max\left(\sup_{a \in A} \inf_{b \in B} \|a - b\|, \sup_{b \in B} \inf_{a \in A} \|a - b\|\right) \tag{4.3}$$

In this equation, A and B stand for two sets of points, and $\| \cdot \|$ indicates the distance metric that is employed, which could be the Euclidean distance or any other compatible metric. In this work, the Euclidean distance is applied. The Hausdorff distance quantifies the greatest separation between the nearest point in one set and the point in the other set. The Hausdorff distance at the 95th percentile, or HD95, is applied in this work. The HD95 distance represents the distance between the 95th closest point in one set and the closest point in the other set. Due to its reduced sensitivity to outliers, the HD95 is a more reliable indicator of segmentation accuracy than the traditional Hausdorff distance. It is to be noted that the Hausdorff distance is fundamentally a relative measure, and as such, it is not suitable for direct comparisons across different domains.

5 Results

For all performed experiments described in sections 3 and 4, different DDPMs were trained, tested, and evaluated on the MIoU, Dice and HD95 score. In the following two subsections, the results of the two different methods are presented. Subsequent subsections will go over the results of the different experiments performed for the two different methods.

5.1 DDPM as Representation Learner

5.1.1 The Effect of the Blocks and Timesteps

In this subsection, an attempt is made to answer the research questions: "How do the specific blocks used from the U-Net impact the quality of the segmentation result?" and "How does the amount of noise added during the forward diffusion process affect the performance of the segmentation result?" Figures 5.1, 5.2 and 5.3 display the results of the experiment performed to evaluate the effect of the different timesteps and blocks on the segmentation.

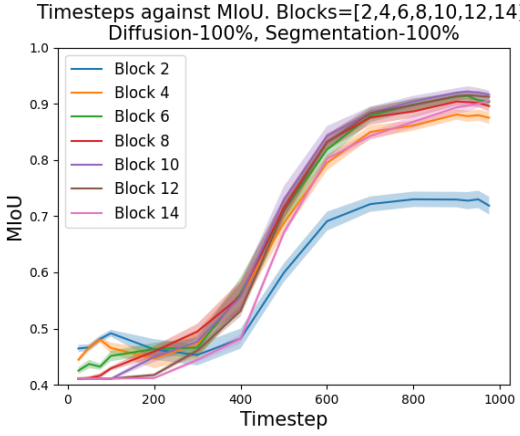

Figure 5.1: Performances of the different blocks/timesteps. The x-axis corresponds to the timesteps of the reverse diffusion process. The y-axis corresponds to the MIoU. The different colours represent the different blocks of the U-Net. The bounds represent the measured uncertainty.

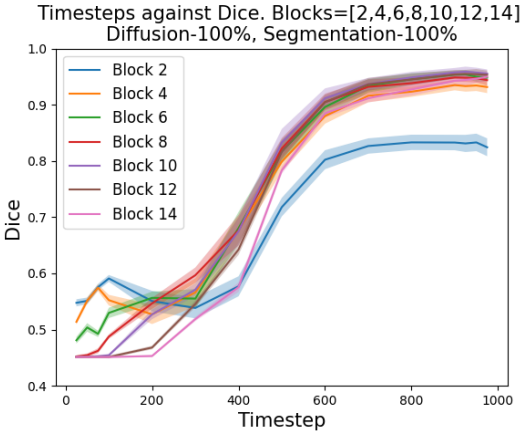

Figure 5.2: Performances of the different blocks/timesteps. The x-axis corresponds to the timesteps of the reverse diffusion process. The y-axis corresponds to the Dice score. The different colours represent the different blocks of the U-Net. The bounds represent the measured uncertainty.

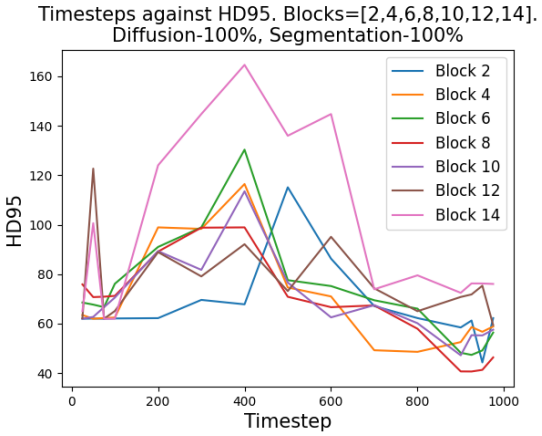

Figure 5.3: Performances of the different blocks/timesteps. The x-axis corresponds to the timesteps of the reverse diffusion process. The y-axis corresponds to the HD95 distance. The different colours represent the different blocks of the U-Net.

Figures 5.1 and 5.2 display the results of the MIoU and Dice against different timesteps for the different blocks. Each line illustrates a different block of the U-Net. The bounds displayed for each graph correspond to the uncertainties of the predictions as the predictions were made by

an ensemble of MLPs. It is to be noted that the timesteps in these Figures and in the remaining Figures in the results section refer to the timesteps of the reverse diffusion process (contrary to most literature where t refers to the forward diffusion process), this is done such that it is more intuitive to understand how the performance increases. It can be observed in both Figures that as the timesteps increase, the overall performance increases as well. To put it, when the images contain little noise, the predictive power of the features increases. This is likely the case due to the overall structure of the images still being present in these later timesteps, from which useful features can be learned for the segmentation, whereas for the earlier timesteps not much can be learnt from the heavily noised images. The largest increase in performance occurs between t=400 and t=600, whereafter the performance starts to flatten out. The strong increase in performance between the aforementioned timesteps can likely be attributed to the overall structure of the images starting to appear between these timesteps. Comparing the different blocks of the U-Net, it can be observed that the middle to high blocks have the best predictive performance. The lacking performance of the lower blocks is likely due to these blocks picking up on large abstract features which are less relevant in detecting the uniformly shaped and uniformly looking organoids. This hypothesis is evaluated later in this section. This is also supported by the fact that the uncertainty is the largest for block 2, whereafter the uncertainty decreases for each consecutive higher block (Uncertainty block 2 is 0.013, uncertainty block 14 is 0.006). In Figure 5.3, the HD95 distance can be observed for the different blocks and timesteps. Similar effects as were seen for the MIoU and Dice can be observed in this Figure. Since a lower Hausdorff distance indicates better performance, it can be observed that the performance increases later on in the reverse diffusion process. Different from the MIoU, the worst-performing block in terms of Haussdorf distance is block 14, whereafter blocks 2 and 12 follow. The poor performance of the high blocks might be attributed to the fact that the HD95 distance compares structures. The high blocks capture fine-grained details in the images which are not useful in capturing the exact shape of the organoids and thus there can be large differences between the structures of the predictions and the true segmentations.

In order to investigate what the blocks in combination with the timesteps pick up upon and to examine the aforementioned hypotheses for the observed results, it is necessary to perform visual analyses of the segmentation maps and the feature maps of the U-Net. Figure 5.4 and Figure 5.5 show segmentations for two different images with respect to different blocks and timesteps. It can be observed that the blocks only start to pick up upon the organoids from timestep 500 onwards, which confirms the hypothesis that the overall structures of the organoids start to appear around this timestep and which explains the earlier observed performance increase around this timestep. Moreover, the middle (Block 8) and later block (Block 14) show the best performance, where block 8 shows the overall highest performance. It can be observed that block 14 compared to block 8 is failing to pick up on pixels within the organoid. To get a better insight into the causes of the differences between the segmentations, a k-means algorithm (k=5) has been trained to investigate the representations of the different blocks and timesteps. Figure 5.6 and Figure 5.7 display the clusters that have been picked up upon by different blocks and different timesteps for 2 images. It can be noticed that block 2 captures low-level features which are not useful for the segmentation of the organoids. This explains the overall worst performance of the lower blocks. Block 14 in contrast picks up on fine-grained details. The features of this block explain the worst performance of this block in terms of HD95 as the exact structure of the organoids are not picked up upon by these blocks. It can be observed that in the later timesteps (t=975) of block 14, boundary identification of the organoids starts to occur. However, since there exist different clusters within the organoids for this block, it is difficult to segment the organoids based on these feature maps. This inability of these later blocks in full segmentation of the organoids is in line with the results of block 14 in Figure 5.5, in which it could be seen that the block is not predicting all pixels within the cells correctly. Block 6 and 8, on the other hand, are more accurately able to identify the entire organoids, where block 8 compared

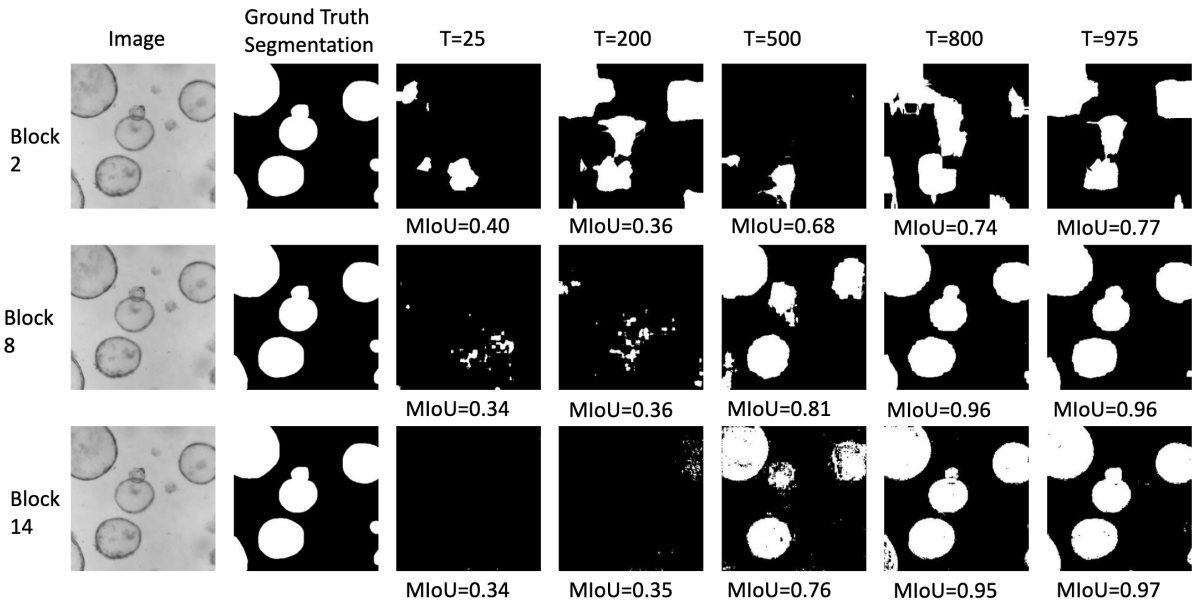

Figure 5.4: Segmentation maps and ground truth segmentations for 2 images for blocks=[2,8,14] and timesteps T=[25,200,500,800,975]. Under each image, the MIoU is displayed in order to allow for a better comparison of the segmentation maps.

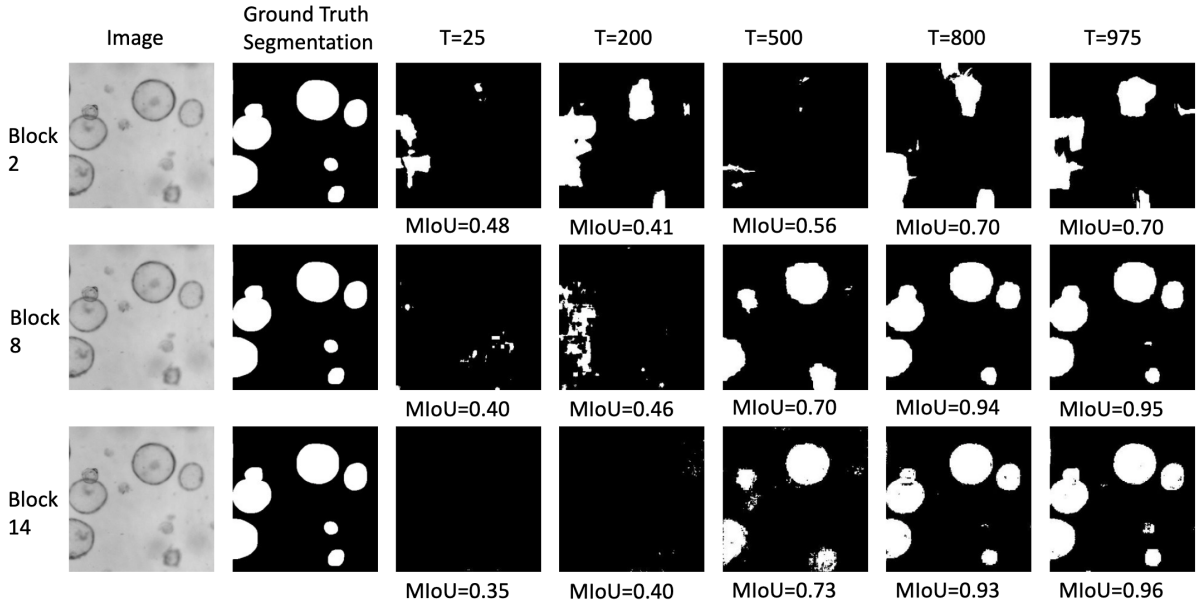

Figure 5.5: Segmentation maps and ground truth segmentations for 2 images for blocks=[2,8,14] and timesteps T=[25,200,500,800,975]. Under each image, the MIoU is displayed in order to allow for a better comparison of the segmentation maps.

to block 6 is also able to identify the borders of the organoids. To answer the research questions mentioned at the beginning of the subsection, the features that result from the layers in the

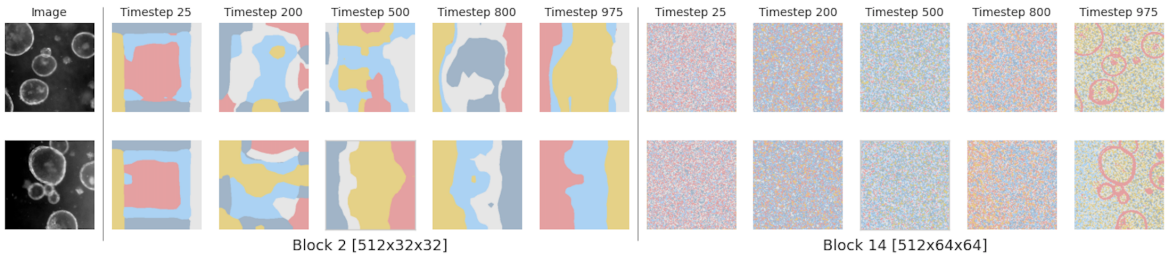

Figure 5.6: Examples of k-means clusters (k=5) formed by the features extracted from the U-Net decoder blocks= [2,14] on the diffusion steps = [25, 200, 500, 800, 975]

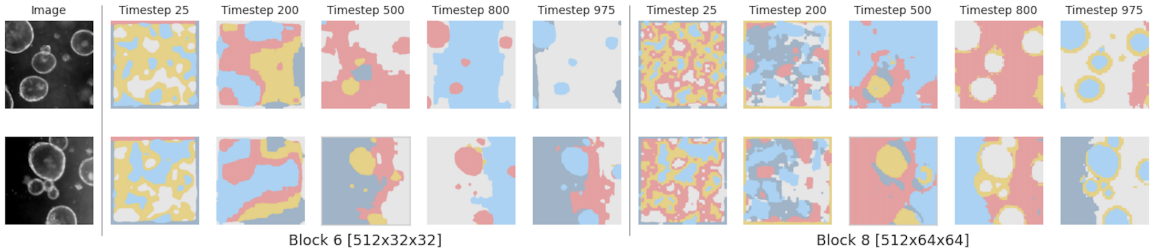

Figure 5.7: Examples of k-means clusters (k=5) formed by the features extracted from the U-Net decoder blocks= [6,8] on the diffusion steps = [25, 200, 500, 800, 975]

middle of the U-Net decoder appear to have the most semantic meaningness for the task at hand when compared across different blocks. Moreover, the later timesteps (t=800 and onwards) are the timesteps with the most semantic meaningness.

5.1.2 The Effect of the Training Size

In this subsection, an attempt is made to answer the research questions: "How does the size of the dataset used to train the diffusion model impact the quality of the segmentation result?" and "How does the size of the dataset used to train the ensemble of MLPs influence the quality of the segmentation result?".

Figures 5.8, 5.9 and 5.10 display the results of the experiment performed to evaluate the effect of the different training sizes used. The graphs present the performance versus the training percentages used, where the x-axis shows the different percentages used to train the MLPs and the different colours represent the different percentages used to train the actual diffusion model.

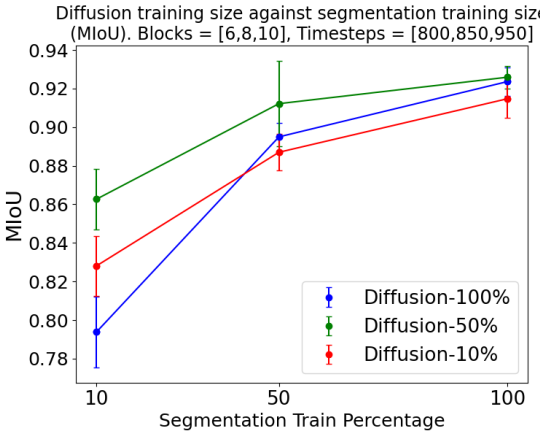

Figure 5.8: Performances of the different training percentages used. On the x-axis, the percentage of data used for training the MLPs is displayed. On the y-axis, the MIoU is displayed. The different colours refer to the different amounts of training data used for training the diffusion model. The bounds display the uncertainty of the predictions.

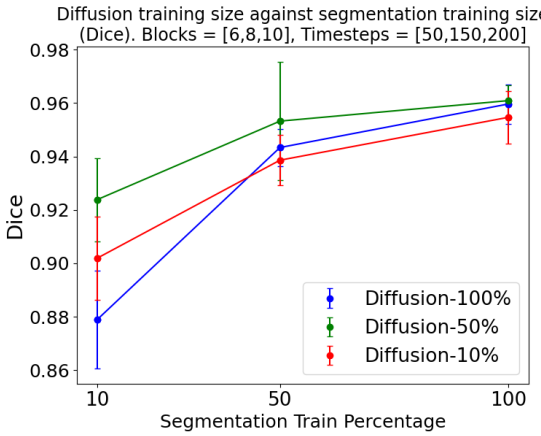

Figure 5.9: Performances of the different training percentages used. On the x-axis, the percentage of data used for training the MLPs is displayed. On the y-axis, the Dice score is displayed. The different colours refer to the different amounts of training data used for training the diffusion model. The bounds display the uncertainty of the predictions.

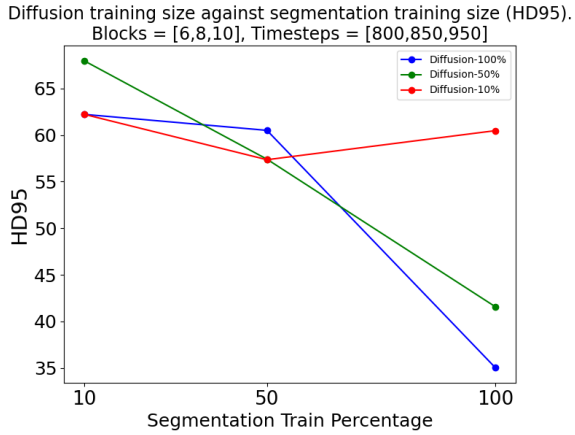

Figure 5.10: Performances of the different training percentages used. On the x-axis, the percentage of data used for training the MLPs is displayed. On the y-axis, the HD95 score is displayed. The different colours refer to the different amounts of training data used for training the diffusion model.

The bounds in Figure 5.8 and Figure 5.9 represent the uncertainties, which were measured for the MIoU and Dice. It can be observed in Figure 5.8 and Figure 5.9 that the percentage of training data used to train the MLPs has the largest effect on the performance, where the largest increase happens going from 10% to 50% of the data. The segmentation performance of only using 10% of the data is relatively weak compared to using 100%, which shows high performance, with the MIoU ranging between 0.91 and 0.93 and the Dice score ranging between 0.94 and 0.96. The decreasing uncertainties when using more data to train the MLPs also support the superiority of using more data for the segmentation. When comparing the different amounts of diffusion data used, it can be observed that this difference is large when little segmentation data is used. This difference, however, starts to become more and more negligible once the amount of segmentation data increases. Moreover, using 100% of the diffusion data only shows the best performance, when 100% of the segmentation data is used as well. This suggests that to accurately make use of the feature maps by the U-Net, sufficient data needs to be used to train the segmentation models, otherwise, the model cannot effectively map the feature maps to correct segmentation maps. Figure 5.10, which shows the HD95 distance, shows the same effect as the other two Figures. However, the effect of using more segmentation training data is better visible. The Figure illustrates that the structures of the organoids are hard to determine when using little data to train the diffusion model, as the red graph shows by far the worst performance when using 100% of the segmentation training data.

To conclude and answer the research questions mentioned at the beginning of this subsection, using more training data for the MLPs improves the performance the most. Moreover, the diffusion model training percentage is more sensitive to limited segmentation data, but using 100% of both types of data achieves the best results (MIoU=0.92, Dice=0.96, HD95=35). Finally, using 50% (21174 images) of the diffusion training data results in similar results as using 100% (42348 images) of the diffusion training data. This implies that using 50% of the diffusion training data is enough to achieve close to optimal results.

5.2 DDPM for Direct Segmentation

5.2.1 The Effect of the Ensemble Size

This section answers the research question: "How does the number of samples generated by the diffusion model affect the performance of the segmentation?". Table 5.1 displays the results of the experiment that was performed to evaluate the effect of the different sample sizes used to generate the final segmentation, where n represents the sample size. It can be observed that over all performance

Table 5.1: The effect of the sample size (n) on the performance. The best performances for each metric are in bold.

	MIoU	Dice	HD95
n=1	0.53	0.62	73.43
n=2	0.50	0.42	74.75
n=3	0.62	0.71	62.20
n=4	0.52	0.61	67.70
n=5	0.65	0.75	58.97
n=6	0.58	0.67	63.62
n=7	**0.69**	**0.78**	**55.46**
n=8	0.62	0.72	60.8
n=9	**0.69**	**0.78**	56.23
n=10	0.65	0.74	57.18

metrics, the performance increases as the number of sampled images increases. The largest improvements in performance start to occur from n=5 after which the overall highest performance is reached when n=7. Moreover, n=9 shares the highest performance with n=7 in terms of MIoU and Dice. In general, there is a lot of fluctuation in terms of performance when the sample size is increased. Overall, the trend is however upward until n=7. To conclude, using more samples to produce the final segmentation map improves the quality of the produced segmentation maps. The results suggest that with a sample size of 7, increasing the sample size further does not provide better results.

5.2.2 The Effect of the Training Size

This section answers the research question: "How does the size of the dataset used to train the diffusion model impact the quality of the segmentation result?". Table 5.2 shows how the dataset sizes affect the segmentation outcomes. It is clear that there are significant disparities between the various data percentages; for example, utilizing 50% and 10% compared to 100% of the data yields no effective segmentations at all because the MIoU and Dice are both below 0.5 and the HD95 distance is also about twice as large compared to the other Dice scores. Clearly, a large annotated dataset is required for training a diffusion model to directly produce segmentation maps.

Table 5.2: The effect of the training data on the performance. The best performances for each metric are in bold.

	MIoU	Dice	HD95
100%	**0.62**	**0.71**	**62.15**
50%	0.37	0.49	81.94
10%	0.11	0.20	157.73

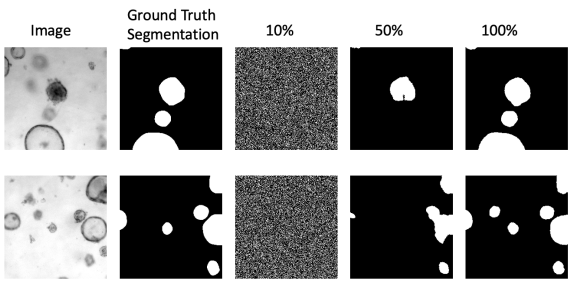

Figure 5.11: Predictions for different training sizes for 2 example images

Figure 5.11 shows sampled segmentation maps for two example images for different training percentages used. When observing these visual results, it can be seen that when the models are trained on only 10% of the data, they are not able to learn the data distribution at all. Moreover, when only half of the data was used to train the model, only rough representations of the data are learned. With 100% of the data, the model is clearly able to identify the organoids.

To conclude, increasing the number of data used to train the diffusion model strongly improves the quality of the segmentation map. It is however necessary to use a large annotated dataset to

ensure the highest performance as using 100% of the data (42368 images) gave a much higher performance (MIoU=0.62, Dice=0.71, HD95=62) compared to the performance (MIoU=0.37, Dice=0.49, HD95=82) of using 50% of the data (21184 images).

6 Conclusions

6.1 Discussion

In this work, the ability of DDPMs to effectively segment organoid culture data as a representation learner or by directly sampling segmentation maps has been evaluated.

The representation approach showed to be effective as a label-efficient segmentation method by showing high performance (MIoU=0.92, Dice=0.96, HD95=35) with as little as 20 labelled training samples. This is a significant advantage over the contemporary organoid segmentation techniques since those techniques all rely on sizable annotated datasets. Do, however, note that a large dataset is still required since another 42348 unannotated images were used in the training process. Moreover, for the representation method, it is found that the middle blocks and early timesteps provide the best predictive performance (see section 5.1.1). Also, it has been found that in the entire training pipeline, the size of the segmentation dataset is more important for performance than the size of the diffusion dataset (see section 5.1.2).

The direct segmentation approach cannot be used for label-efficient segmentation as labelled images are required in the entire training process. For the direct segmentation approach, it is found that increasing the sample size facilitates more stable segmentation maps and consequently better performance (see section 5.2.1). The amount of training data used also greatly impacted the performance considering that the performance of using 10% (42368 images) of the training data (MIoU=0.11, Dice=0.20, HD95=158) was significantly lower than the performance (MIoU=0.62, Dice=0.71, HD95=62) of using 100% (42368 images) of the training data (see section 5.2.2).

Overall, a conclusion can be drawn that the approach that uses a DDPM as a representation learner is greatly superior for segmenting organoid cultures. When using 100% of the data (42368 images) for all stages of the two methods, the representation approach (MIoU=0.92, Dice=0.96, HD95=35) outperformed the direct segmentation approach (MIoU=0.62, Dice=0.71, HD95=62) across all metrics (See Figure 5.8, 5.9, 5.10 and Table 5.1 of the result section). Given the high performance of the representation method and the little amount of annotated data required, the method is found to be effective for label-efficient segmentation of organoid culture data. The direct segmentation approach is found to be ineffective in the overall segmentation of organoid data. Even when a large annotated dataset is used, the overall performance is much lower than the representation approach.

6.2 Future Work

Several aspects with regard to the performed research can be further investigated. The method used to construct segmentation maps directly is picking up on organoid-looking objects that are not categorized as organoids in the ground-truth segmentation. When observing the predicted segmentations on the test set, this is a recurrent phenomenon that lowers the overall performance of all approaches. These objects appear much more blurred compared to the actual organoids. Experimenting with different loss functions such as the Structural Similarity Index (SSIM) loss could help prevent the issue. The SSIM loss for example better takes into account luminance, contrast, and structure information (Wang et al., 2004), making it more perceptually relevant than pixel-wise metrics like the used Cross-Entropy loss. The chaotic losses that can be observed for the representation learner on the validation data (Figure A.1 of the appendix) support the idea that other loss functions may be better suited for the task.

The impact of larger sample sizes for the direct segmentation approach is another area to research in the future since it is unclear whether even greater sample sizes than those currently investigated would improve the performance. As

the later sample sizes are still fluctuating a lot, this might suggest that a plateau might not have been reached yet. Moreover, the same area can be investigated for the segmentation training data.

References

Adhikari, B., Rahtu, E., & Huttunen, H. (2021). Sample selection for efficient image annotation. In *2021 9th european workshop on visual information processing (euvip)* (p. 1-6). doi: 10.1109/EUVIP50544.2021.9484022

Amit, T., Nachmani, E., Shaharbany, T., & Wolf, L. (2021). Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*.

Baranchuk, D., Rubachev, I., Voynov, A., Khrulkov, V., & Babenko, A. (2021). Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*.

Borten, M. A., Bajikar, S. S., Sasaki, N., Clevers, H., & Janes, K. A. (2018). Automated brightfield morphometry of 3d organoid populations by organoseg. *Scientific reports*, *8*(1), 5319.

Brouwer, E. (2022). *Supervised versus self-supervised: Which is better for biomedical image segmentation?* (Unpublished doctoral dissertation).

Chakraborty, S., & Mali, K. (2023). An overview of biomedical image analysis from the deep learning perspective. *Research Anthology on Improving Medical Imaging Techniques for Analysis and Intervention*, 43–59.

Crowe, A. R., & Yue, W. (2019). Semi-quantitative determination of protein expression using immunohistochemistry staining and analysis: an integrated protocol. *Bio-protocol*, *9*(24), e3465–e3465.

Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2020). A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 1-22.

de Souza, N. (2018). Organoids. *Nature Methods*, *15*(1), 23–23.

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, *34*, 8780–8794.

Drost, J., & Clevers, H. (2018). Organoids in cancer research. *Nature Reviews Cancer*, *18*(7), 407–418.

Fatehullah, A., Tan, S. H., & Barker, N. (2016). Organoids as an in vitro model of human development and disease. *Nature cell biology*, *18*(3), 246–254.

Galeev, D., Sofiiuk, K., Rukhovich, D., Romanov, M., Barinova, O., & Konushin, A. (2021). Learning high-resolution domain-specific representations with a gan generator. In *Structural, syntactic, and statistical pattern recognition: Joint iapr international workshops, s+ sspr 2020, padua, italy, january 21–22, 2021, proceedings* (pp. 108–118).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851.

Hu, X., Chen, Y.-J., Ho, T.-Y., & Shi, Y. (2023). *Conditional diffusion models for weakly supervised medical image segmentation.*

Huch, M., & Koo, B.-K. (2015). Modeling mouse and human development using organoid cultures. *Development*, *142*(18), 3113–3125.

Laousy, O., Araujo, A., Chassagnon, G., Revel, M.-P., Garg, S., Khorrami, F., & Vakalopoulou, M. (2023). *Towards better certified segmentation via diffusion models.*

Matthews, J., Schuster, B., Kashaf, S. S., Liu, P., Bilgic, M., Rzhetsky, A., & Tay, S. (2022). Organoid: a versatile deep learning platform for organoid image analysis. *bioRxiv*, 2022–01.

Melas-Kyriazi, L., Rupprecht, C., Laina, I., & Vedaldi, A. (2021). Finding an unsupervised image segmenter in each of your deep generative models. *arXiv preprint arXiv:2105.08127*.

Nichol, A. Q., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International conference on machine learning* (pp. 8162–8171).

Pegoraro, G., & Misteli, T. (2017). High-throughput imaging for the discovery of cellular mechanisms of disease. *Trends in Genetics*, *33*(9), 604–615.

Powell, R. T., Moussalli, M. J., Guo, L., Bae, G., Singh, P., Stephan, C., ... Davies, P. J. (2022). deeporganoid: A brightfield cell viability model for screening matrix-embedded organoids. *SLAS Discovery*, *27*(3), 175–184.

Rahman, M. M., Amato, G. L., Schilling, F., Seyed-Ahmad, M., & Tajbakhsh, M. (2023). Ambiguous medical image segmentation using diffusion models. *Medical Image Analysis*, *73*, 102–116.

Roder, M., Almeida, J., De Rosa, G. H., Passos, L. A., Rossi, A. L., & Papa, J. P. (2021). From actions to events: A transfer learning approach using improved deep belief networks. In *2021 ieee symposium series on computational intelligence (ssci)* (pp. 01–08).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–miccai 2015: 18th international conference, munich, germany, october 5-9, 2015, proceedings, part iii 18* (pp. 234–241).

Sharifani, K., & Amini, M. (2023). Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal*, *10*(07), 3897–3904.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). *Deep unsupervised learning using nonequilibrium thermodynamics.*

Takebe, T., Zhang, R.-R., Koike, H., Kimura, M., Yoshizawa, E., Enomura, M., ... Taniguchi, H. (2014). Generation of a vascularized and functional human liver from an ipsc-derived organ bud transplant. *Nature protocols*, *9*(2), 396–409.

Tritrong, N., Rewatbowornwong, P., & Suwajanakorn, S. (2021). Repurposing gans for one-shot semantic part segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4475–4485).

Voynov, A., & Babenko, A. (2020). Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning* (pp. 9786–9796).

Voynov, A., Morozov, S., & Babenko, A. (2021). Object segmentation without labels with large-scale generative models. In *International conference on machine learning* (pp. 10596–10606).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, *13*(4), 600–612.

Wei, R., & Mahmood, A. (2021). Recent advances in variational autoencoders with representation learning for biomedical informatics: A survey. *IEEE Access*, *9*, 4939-4956. doi: 10.1109/ACCESS.2020.3048309

Wolleb, J., Sandkühler, R., Bieder, F., Valmaggia, P., & Cattin, P. C. (2022). Diffusion models for implicit image segmentation ensembles. In *International conference on medical imaging with deep learning* (pp. 1336–1348).

Xu, J., & Zheng, C. (2021). Linear semantics in generative adversarial networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9351–9360).

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Yudistira, N., Kavitha, M., Itabashi, T., Iwane, A. H., & Kurita, T. (2020). Prediction of sequential organelles localization under imbalance using a balanced deep u-net. *Scientific reports*, *10*(1), 2626.

Zeune, L. L., Boink, Y. E., van Dalum, G., Nanou, A., de Wit, S., Andree, K. C., ... Brune, C. (2020). Deep learning of circulating tumour cells. *Nature Machine Intelligence*, *2*(2), 124–133.

Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J.-F., Barriuso, A., ... Fidler, S. (2021). Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10145–10155).

Zhu, C., Chen, W., Peng, T., Wang, Y., & Jin, M. (2021). Hard sample aware noise robust learning for histopathology image classification. *IEEE Transactions on Medical Imaging*, *41*(4), 881–894.

A Appendix

Table A.1: Full list of Diffusion Model Hyperparameters(For an explanation of all hyperparameters we refer to Nichol & Dhariwal)

Diffusion Steps	1000
Noise Schedule	Linear
Learn Sigma	False
Sigma Small	False
Use KL	True
Predict x_{start}	False
Rescale Timesteps	False
Rescale Learned Sigmas	False
Timestep Respacing	False

Table A.2: Full list of Denoising Model Hyperparameters (For an explanation of all hyperparameters we refer to Nichol & Dhariwal)

Attention Resolutions	32x32,16x16,8x8
Dropout	0.1
Learn Sigma	True
Number of Channels	256
Number of Head Channels	64
Residual Block up/down	True
Number of Residual Blocks	2
Residual Blocks on Both Paths	True
Using Scale Shift Norm	True
Num of heads upsample	0
New attention order	False

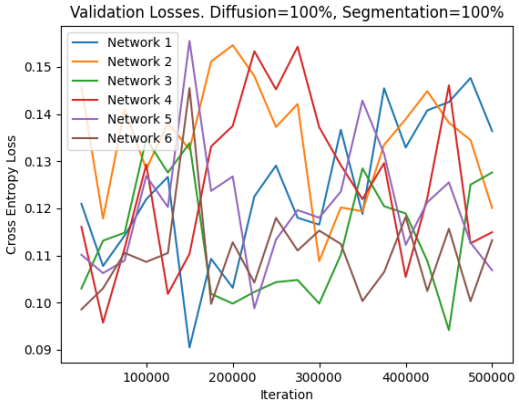

Figure A.1: Example validation losses for the representation learner