Segmentation Of Organoid Cultures Images Using Diffusion Networks with Triplet Loss

Asmaa Haja  
Bernoulli Institute, University of Groningen, The Netherlands  
a.haja@rug.nl

Ivaylo Zhelev  
University of Groningen, The Netherlands  
i.zhelev@student.rug.nl

Lambert Schomaker  
Bernoulli Institute, University of Groningen, The Netherlands  
l.h.schomaker@rug.nl

ABSTRACT
The present research study explores the use of Diffusion Networks with Triplet Loss for the semantic segmentation of liver organoid cultures images. Since diffusion networks are generative, they encode more abstract and higher-level representations of the training images. In order to adapt such a generative architecture for image segmentation, we implemented triplet loss as its loss objective so that we encourage the diffused predictions to resemble the segmentation maps instead of the original images. The research question is whether the triplet loss is applicable to such an architecture and task, and whether diffusion networks with triplet loss are more performant, and trainable than previous supervised and self-supervised alternatives. This was tested by training the proposed model with a maximum of 500 images, evaluating its F1-score and comparing it to a supervised and self-supervised baseline. Our model (F1=83.7%) significantly outperformed the supervised baseline (F1=64.4%) but was outperformed by the self-supervised one (F1=85.0%). Nevertheless, our model showed greater robustness for typical images but low reliability, as it struggled with ambiguous inputs, causing a skewed distribution of the results and a lower mean score. It was also shown that the proposed model is more trainable than supervised approaches as it needed only 50 training samples to outperform the supervised baseline, which was trained with 114. However, it was less trainable than the self-supervised baseline, whereas it took our model over 200 training samples to match the accuracy of the self-supervised one (trained on 114).

1 INTRODUCTION
In the contemporary data-oriented advancement of the biomedical field, image data has become central to medical diagnoses, academic research and the development of drugs (European Society of Radiology, 2015). To fully leverage the capabilities of medical imaging, it is necessary to automate data analysis in the field while maintaining high analytical accuracy [3]. This can be achieved through the use of advanced computational techniques, which can help to improve the efficiency, accuracy, and reproducibility of image analysis. Deep Learning, specifically, has recently achieved state-of-the-art results in all biomedical computational task and has become the standard method for bioinformatical processing [26].

Among the numerous challenges in biomedical image processing, semantic segmentation is recognized as one of the most important tasks, as it may prove to be integral to illness understanding and medical diagnoses [21]. It involves the identification and labeling of specific structures or regions within an image. The development of automated semantic segmentation has shown great potential in various biomedical applications, including brain imaging, histopathology, and ophthalmology [16]. However, the goal of the present research study is to address the semantic segmentation of organoid cultures images.

Organoid cultures offer a promising avenue for studying the complex cellular interactions and dynamics within liver tissue [6]. However, analyzing the large amount of image data generated by such cultures can be time-consuming and labor-intensive [9]. Automated semantic segmentation of organoid images can facilitate the quantification and visualization of cellular structures, leading to a deeper understanding of liver development and disease progression. Nevertheless, accurate segmentation of organoids is a challenging task due to the complexity and heterogeneity of the images, as well as the lack of annotated data. Therefore, the present research study aims to develop and evaluate advanced computational methods for precise and reliable segmentation of liver progenitor organoid images, with the goal of improving our understanding of liver biology and disease.

The rise of Deep Learning (DL) has significantly improved the state of art in semantic segmentation and helped mitigate the problem with data complexity and heterogenity. Convolutional Neural Networks (CNNs) such as AlexNet [12], VGG [22], and Microsoft’s ResNet [10] pioneered the development of deep learning methods for semantic segmentation. However, these approaches are fully supervised and are reliant on the quality and quantity of available annotated data. Conversely, self-supervised architectures mitigate this limitation for binary segmentation, as their training needs a insignificant amount of labeled data whereas their architecture is pre-trained by a self-supervised model. [7] tested the performance
of supervised and self-supervised approaches and demonstrated that self-supervised approaches are significantly more generalizable and reliable for semantic segmentation of organoid images.

One promising example of self-supervised approaches are generative methods such as Generative Adversarial Networks (GANs) [27], [28]. Generative approaches learn to generate synthetic images representative of the real-world data instead of purely translating between an image and its segmentation. Therefore, they have the advantage of deriving more abstract and higher-level representations of data that are semantically meaningful [14]. [17] showed that the latent space learned by a GAN can capture meaningful semantic features that can be used for downstream tasks, such as image classification and image segmentation. Furthermore, GANs are self-supervised and enable the use of synthetic datasets for training.

In spite of GANs’ early success, their adversarial element makes them difficult to train and hypertune [15]. This is because adversarial methods employ competition between two neural networks (the generator and discriminator), which makes the training equivalent to finding a Nash equilibrium to a two-player non-cooperative game. Finding such an equilibrium is a computationally complex problem, and there is no guaranteed convergence in a high-dimensional, continuous parameter space [25]. Recently, Diffusion Networks [23], and specifically Denoising Diffusion Probabilistic Models (DDPMs) [11] were introduced to provide a generative alternative without an adversarial element.

Their general principle (illustrated in Fig. 1) is to iteratively pollute each sample image with more and more Gaussian noise in order to produce a $T$ number of noisy images corresponding to each ground-truth image. The noisy image at each step $t$ is generated by adding noise to the image at step $t - 1$, whereas the image at step 0, is the original ground-truth. Then, an ensemble of Multi-Layer Perceptrons (MLPs) are trained to reverse this process by learning to map a noisy image to a less noisy one (e.g., mapping $x_t$ to $x_{t-1}$). The architecture of these MLPs may vary depending on the specific implementation but U-Nets [18] are typically used as the backbone of the Diffusion Model. Ultimately, this allows the inference of a clear image from pure noise.

2 CONTRIBUTION

We seek to build a DDPM architecture that is both generalizable and trainable. The latter can be achieved by minimizing the complexity of the architecture and the number of trainable parameters. Therefore, the ideal solution would involve a diffusion network that produces the segmentation map directly during the diffusion process such as the one proposed by [30]. However, unlike the latter research, we want to avoid passing the segmentation map directly to the training as this hinders the generalizability and the scientific validity of the findings.

The proposed solution is a native DDPM which generates the segmentation map in the decoding process like the one depicted in Fig. 2. In order to implement such an architecture, we propose modifying the loss objective to the DDPM such that we encourage the denoised predictions to resemble the noisy segmentation maps instead of the noisy ground truths.

To that end, we propose to adapt Triplet Loss to generative DDPM-based segmentation. Triplet loss was first introduced in [20]. It was originally designed for classification tasks but we assert that it can be adapted for the semantic segmentation task in a way that allows segmentation to be integrated seamlessly into the diffusion process. A detailed description and formalization of the triplet loss can be found in Section 4.2.

In line with exploring such a model’s comparative accuracy, generalizability, and trainability, our contribution is to answer the following Research Questions:

- Is triplet loss applicable to such an architecture and task?
- Do diffusion networks with triplet loss yield higher accuracy than previous models (supervised and self-supervised)?
- Are diffusion networks with triplet loss more trainable? Do they require less data than previous models (supervised and self-supervised)?
- What are the optimal training and sampling parameters?

Given the simple architecture of the proposed model, we postulate the following hypotheses:

- Triplet loss is applicable to the task of semantic segmentation and will outperform the Mean Square Error (MSE) loss, as measured by the resultant model’s accuracy and trainability
- Diffusion networks with triplet loss will yield higher testing accuracy than previous supervised approaches, as measured by the F1-score of the produced segmentations
- Diffusion networks with triplet loss will be more trainable than previous self-supervised approaches as measured by the amount of data needed to reach convergence

3 RELATED WORKS

The general principle of diffusion networks allows them to generate images which are visually, semantically and functionally similar to the training data. In order to adapt this principle to the task of semantic segmentation, alterations to the architecture are needed and
various approaches have been tested and reported in the scientific literature. [30] developed a diffusion-based model for detecting anomalies in biomedical images, specifically brain MRIs. This is done by training a DDPM to generate images of healthy brains from images of medically anomalous ones with an Encoder-Decoder denoising network. The resultant image simulates the brain of the corresponding patient if it was healthy. Finally, the difference between the produced image and the ground truth is taken to produce an anomaly map. This approach works well for detecting visually significant irregularities from reference images, however it is not expected to be useful for the segmentation of organoid cultures. This is because the latter are the only visually significant element of the images and the rest is a relatively uniform background.

Another way to perform a post-priori segmentation of the generated images is by deriving the semantic features of the images during generation and using them to train a classifier. The semantic features can be extracted by obtaining the activations of each layer during the decoding process [4]. These activations are then upsampled to fit into the dimensionality of the original image and concatenated. The resultant tensor is then used to train an MLP classifier which classifies each pixel to a semantic label.

While this proves to be a very effective approach for semantic segmentation, the segmentation is pixel-wise which makes it slow. Furthermore, the derived feature maps may contain a lot of information that has low leverage over the final classification, making the feature maps informationally sparse and the classifier may be difficult to train and hypertune.

Instead of performing such a post priori discrimination between the generated image and ground truth, [31] developed a DDPM which directly infers the segmentation map during the diffusion process. In other words, the image is polluted with noise in the encoding process but the decoding process does not regenerate the original image. Instead, it generates progressively denoised versions of the segmentation map. This is done by concatenating the ground truth segmentation map to the noisy ground truths during the training of each block. This approach allows for a direct segmentation as part of the diffusion process. However, passing the ground truth as input to the training process causes some scientific invalidity to this process. This is why the testing accuracy of this approach is not closely matched to the one during training.

This scientific invalidity may be fixed by passing the original image to each block instead of the segmentation map as done by SegDiff [1]. The features from each noisy image are derived by a neural network F and are added to the features derived from the ground truth with a network G and the sum is passed to an Encoder-Decoder network E + D. This approach contains 4 separate models for each block of training which dramatically increases the number of trainable parameters and hinders the model’s trainability. This is also expressed by the fact that underrepresented labels in the data are inferred with a significantly reduced accuracy. Even though, the present research is dealing only with binary segmentation, this may be a problem for the task at hand due to class imbalance.

4 METHODS

4.1 Diffusion Models

As mentioned before, Denoising Diffusion Probabilistic Models (DDPMs) are generative models, which learn to generate simulated images from noise. This is done by iteratively polluting each training image with more and more Gaussian noise (diffusion process). The diffusion produces a T number of noisy versions of the training image (see Fig. 1). At each timestep t, the image x_t is generated by adding Gaussian noise to the image at the previous step x_t−1, whereas x_0 is the original ground truth. Then, an MLP backbone is trained to reverse the diffusion (decoding process) by predicting each image x_t from its slightly noisier variant x_{t+1} at the next timestep.

The loss objective used to train a Diffusion model is equivalent to a Mean Square Error loss and is described in Eq. 1, where e(x_t, t) refers to the output of the MLP decoder at step t and e denotes the noisy ground truth.

\[ L_{simple}(\theta, t) = \|e(x_t, t) - e\|^2 \]  

4.2 Triplet Loss

The desired solution is a native DDPM which generates the segmentation map in the decoding process like the one depicted in Fig. 2. In order to implement such an architecture, we propose modifying the loss objective to the DDPM such that we encourage the denoised predictions to resemble the ground truth segmentation maps instead of the original input images. Such a modification to the loss function will allow the decoding process to deviate further from regenerating the original image, and will steer it closer to the segmentation at each iteration.

To that end, we propose to adapt Triplet Loss to generative DDPM-based segmentation. Triplet loss was first introduced in [20]. It was originally designed for image classification tasks but one of the objective of this study is to test if it can be adapted for the semantic segmentation task in a way that allows segmentation to be integrated seamlessly into the diffusion process.

In the image classification task, triplet loss encourages dissimilar pairs to be distant from any similar pairs by at least a certain margin value [19].

The proposed research introduces a modified variant of the triplet loss, which encourages the images inferred from the U-Net backbone to be more similar to the segmentation map than to the original image (see Fig. 3). This can be formalized, as shown in Eq. 2, where x'_t is the generated noisy image from sample i at step t. The output of the decoding model is denoted \( e_\theta(x) \), given the set of trainable parameters \( \theta \) and \( b^t \) represents the ground truth segmentation map of sample i.

\[ L_t = \sum_i^N \left[ \|e_\theta(x'_t) - b^t\|^2 - \|e_\theta(x'_t) - x^i_0\|^2 + \alpha \right] \]  

The triplet loss, as formulated above, depends on the timestep. The diffusion process presents varying levels of difficulty at different timesteps (denoising the image at step \( t = 1 \) is much easier than the image at timestep \( t = T \) which is effectively pure noise). Therefore, in order to ensure stable loss descent, the loss function is weighted
After the model has been trained, unseen organoid images can be segmented by first polluting them with noise, equivalent to N noisy timesteps, and then iteratively sampling denoised images from the Diffusion Model for S timesteps (see Fig. 5). At each sampling timestep, the produced image resembles the ground truth segmentation more, and the original image less. The sampling of the images is performed via an accelerated non-Markovian sampling procedure called Denoising Diffusion Implicit Model (DDIM) sampling [24].

Stochastic DDIM sampling, as postulated in [24], is performed using Eq. 4, where $\sigma_t$ is a parameter determining the stochastic component. The values of $\sigma_t$ for each timestep are formulated in Eq. 5, with $\eta$ being a sampling hyperparameter.

$$x_{t-1} = \frac{x_t - \sqrt{1-\sigma_t}\varepsilon_0^{(t)}(x_t)}{\sqrt{\sigma_t}} + \sqrt{1-\sigma_t-\sigma_t^2}\varepsilon_t^{(t)}(x_t) + \sigma_t \varepsilon_t$$

$$\sigma_t = \eta \sqrt{\frac{1-\sigma_{t-1}}{\sigma_{t-1}}} \sqrt{1 - \frac{\alpha_t}{\alpha_{t-1}}}$$

From Eq. 4, it is apparent that if $\eta$ is set to 0, the stochastic element of the sampling process is eliminated, and it becomes deterministic. Through means of explorational experimentation, it was determined that stochasticity elicits instability in the sampling process, which gives rise to accumulative error and low-quality segmentations. For this reason, we set $\eta$ to 0 in our validation experiments.

5 EXPERIMENTAL DESIGN

5.1 Dataset

This study utilized liver progenitor organoids provided by the University Medical Center Groningen (UMCG) in the Netherlands. The organoids were imaged at five different time points using a specialized microscope, with 24-hour intervals, and were grown under two different conditions: one where all necessary amino acids were present (Control group), and another where they were absent (Starvation group).

A total of 10 3D CZI images (see Fig 6), each consisting of 14 2D slices, were captured. Each of the 2D slices were originally 3828x2870 pixels, but they were used to generate a large number of smaller images that can be used as a training dataset. This has been done by moving a square sliding window, of size 636x636, over each 2D slice and generating cropped subsections of the latter. This procedure created a large number of 2D images which have all been resized to 320x320 pixels, in order to reduce the number of needed training parameters, and increase the trainability of the model, while not losing any significant details.

This image dataset was further curated by discarding images composed of less than 5% organoid data. Due to the desired orientation-invariance of the final model, the images were also augmented by generating 4 different rotations for each image - 0°, 90°, 180°, and 270°. The entire data generation procedure yielded 101,577 images usable for training, validation and testing. Each of them was semi-manually labeled with a corresponding segmentation using the OrganetIX service [2] with manual correction, where needed.
5.2 Metrics

The quantitative analyses over the experiments results were performed using four of the most commonly used metrics for the binary semantic segmentation task - the F1-score [15], the SSIM metric [29], the Jaccard Index (also known as IoU) [5], and the Mean Absolute Error (MAE). The F1-score and the Jaccard index are only applicable for binary segmentations (meaning each pixel can only have a value of 0 or 1), therefore the produced segmentations were thresholded with a threshold $\tau$ when evaluated on the latter metrics.

5.3 Experimental Setup

The entire dataset was combined, randomly shuffled, and split into training and testing subsets in a $90\% / 10\%$ proportion respectively, as stated in Table 1. The splitting was performed randomly, with the added constraint that no augmented (rotated) version of an image in the training subset ends up in the testing subset. The latter was done in order to ensure full isolation of the training and testing subsets, with the objective of reaching scientific validity and preventing overfitting.

Table 1: Data distribution of the total training, validation and testing dataset. The images actually used for training were taken as a subset of the full training set.

<table>
<thead>
<tr>
<th>Subset</th>
<th>Samples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>81262</td>
<td>80%</td>
</tr>
<tr>
<td>Validation set</td>
<td>10157</td>
<td>10%</td>
</tr>
<tr>
<td>Testing set</td>
<td>10157</td>
<td>10%</td>
</tr>
<tr>
<td>Total</td>
<td>101577</td>
<td>100%</td>
</tr>
</tbody>
</table>

In order to test the performance and trainability of the diffusion network, trained with triplet loss, 4 variants of the triplet loss Diffusion Model were trained, using 50, 100, 200, and 500 training images, respectively. For each of these experiments, the images used for training and validation were randomly sampled from the full training dataset (consisting of a total of 81262 images) and validated under the paradigm of a $k$-fold cross validation, with $k=5$. The latter was done in order to test the reliability of the model.

After the training of the 4 models was completed, the results of each folds were aggregated and averaged.

An identical experimental procedure was performed for 4 variants of a diffusion network trained with MSE Loss, acting as the control baseline. The 4 variants were equivalently trained on 50, 100, 200, and 500 training images, chosen with the same method described above.

An ablation study was subsequently performed, with the aim of determining the optimal value of the triplet loss margin and sampling timesteps. To this end, four variants of a triplet loss diffusion network were trained on 500 randomly sampled images from the training dataset. The four variants were trained with different values of the triplet loss margin - namely 4, 8, 16, and 24. For each of the trained models, all intermediate images generated during the sampling process at each timestep were collected, in order to analyze the optimal number of sampling timesteps.

5.4 Implementation Details

The diffusion network used for all experiments was trained with 50 timesteps. The choice of this value was motivated by the intention to provide a small number of timesteps in order to test trainability, but still sufficient to facilitate further analysis of the sampling. We also used 5 noising timesteps for the sampling for every experiment, as preliminary experimentation showed that more than 5 noising timesteps makes the image passed to the model for inference too noisy to distinguish any detail from, even for a human observer.

The sampling hyperparameter $\eta$ was set to 0, in order to keep the DDIM sampling deterministic as mentioned before. For the $p_2$ loss weighting schedule hyperparameters, we set $k$ and $\gamma$ both to 1.0, as recommended by [8].

As mentioned before, the backbone of the diffusion network for all experiments was a U-Net. The architecture of the U-Net used for the Diffusion Models had 4 downsampling blocks, 4 upsampling ones and 2 ResNet blocks in the middle. Each block consisted of 2 ResNet layers with 8 block groups and Linear Attention, and an Upsampling / Downsampling layer. Downsampling layers implemented a $4 \times 4$ convolution halving the width, and the height of their input tensor, while doubling its feature vector dimensionality. Upsampling layers performed a $3 \times 3$ convolution and an unsampling operation - doubling the width and height of their input while halving their feature vector dimensionality. The initial Downsampling layer had a dimensionality of 320x320x64 - equivalent to the input image size where each pixel contains a 64-dimensional feature vector.

As for the global training hyperparameters, the gradient descent was performed with the standard Adam optimizer with no weight decay and the standard values of $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learning rate was set to $8 \times 10^{-6}$, as preliminary exploration showed that higher values lead to unstable loss descent, gradient explosion, and even NaN loss values. The random seed used for the splitting of the datasets was 42. The batch size was set at 8, and all experiments were performed on models trained for the same number of epochs - 80, in order to guarantee the scientific validity of the comparisons. Finally, the training and testing of all the models were performed on a single NVIDIA V100 GPU node.
6 RESULTS

For all performed experiments, described in Section 5.3, different DDPM models were trained, tested, and evaluated on their F1-score, Jaccard Index, SSIM, and MAE. All experiments were also subject to a 5-fold cross validation scheme.

We used the collected F1-score results to compare our models’ performance to [7], which we used as a baseline, since it was tested on the same dataset. The latter research tested the performance of supervised and self-supervised approaches on the organoid segmentation task. The best reported results from both model types were compared to the performance of our models.

6.1 Model Performance

![Comparison of the F1-score of our approaches and the baselines](image)

Figure 7: Comparison of the F1-score of our model compared to the supervised and self-supervised baseline [7].

The ablation experiments (further discussed in Section 6.4) demonstrated that the best-performing model was the triplet loss Diffusion Model trained on 500 examples with a margin value of 24.0. The average F1-score from all 5 folds was compared to the supervised and self-supervised baselines, reported in [7] on Fig. 7. From the latter Figure, it is clear that our model significantly outperforms the supervised baseline, as hypothesized, while producing similar variance of the results.

It is also evident from Table 2 that our model is outperformed by the self-supervised baseline on the basis of the mean F1-score. Nevertheless, when evaluated on the median F1-score, our model yields a significantly higher performance (90.0%) than the self-supervised baseline (85.0%). Furthermore, the performance variance within the results of our model is much larger (standard deviation amounts to 16.1%) than the self-supervised baseline (0.5%). This suggests that our model is more performant with more typical images but struggles with rarer, challenging inputs. It can be seen on Fig. 7 that the median F1-score of our model is also significantly larger than the mean, amounting to a skewed distribution of the F1-score over the testing images. In other words, the model demonstrates a higher robustness with the large majority of inputs compared to the baselines but it produces a much higher number of outliers. Therefore, the triplet loss model is considerably less reliable than the self-supervised approach.

6.2 Model Trainability

In order to test the relative trainability of the Triplet DDPM models, a comparison was made between the performance of the model for different training dataset sizes, specifically for 50, 100, 200, and 500 samples. The results are shown in Table 2, where it is apparent that the performance increases with the number of samples, as expected. However, the performance of the 50-sample model is comparatively high, it is not clear how significantly the model improves with more training iterations. It is possible that the proposed model has a steep learning curve but has a low asymptotic performance. In other words, triplet loss diffusion networks may reach a convergence plateau with a relatively small dataset. This is certainly possibly since, unlike MSE, the triplet loss does not guarantee that its minimum equates to the optimum solution.

A comparison was made between the performance of the baselines (each trained on 114 images) and our model with a different number of training samples. On Fig. 11 (the red boxplot displays the performance of the baseline), it is clear that our model is significantly more performant, as well as more trainable than the supervised baseline. The Triplet DDPM needs only 50 images in order to outperform the supervised model, trained on 114. This result was expected since, as previously mentioned, diffusion networks are generative models and they learn more abstract and higher-level representations of the data, as compared to supervised architectures.

However, our model seems to be less trainable than the self-supervised baseline, as displayed on Fig. 10, where the red boxplot displays the self-supervised baseline. Our model only starts outperforming the self-supervised network when trained on 500 samples, compared to the 114, that the latter was trained with. This may be because our model still makes use of the ground truth labels and utilizes them in the loss objective. Therefore, Triplet DDPMs are not...
Table 2: Average F1-score from the 5 folds of all tested models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training samples</th>
<th>F1-score</th>
<th>Jaccard</th>
<th>SSIM</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet DDPM (ours)</td>
<td>50</td>
<td>0.761 ± 0.166</td>
<td>0.640 ± 0.196</td>
<td>0.863 ± 0.078</td>
<td>0.068 ± 0.055</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.782 ± 0.160</td>
<td>0.667 ± 0.195</td>
<td>0.860 ± 0.075</td>
<td>0.063 ± 0.050</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.786 ± 0.169</td>
<td>0.675 ± 0.202</td>
<td>0.868 ± 0.071</td>
<td>0.061 ± 0.050</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.837 ± 0.161</td>
<td>0.747 ± 0.200</td>
<td>0.895 ± 0.066</td>
<td>0.047 ± 0.051</td>
</tr>
<tr>
<td>MSE DDPM</td>
<td>50</td>
<td>0.762 ± 0.157</td>
<td>0.639 ± 0.188</td>
<td>0.829 ± 0.093</td>
<td>0.078 ± 0.057</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.798 ± 0.165</td>
<td>0.691 ± 0.199</td>
<td>0.855 ± 0.079</td>
<td>0.064 ± 0.052</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.808 ± 0.170</td>
<td>0.707 ± 0.205</td>
<td>0.864 ± 0.080</td>
<td>0.059 ± 0.053</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.839 ± 0.152</td>
<td>0.747 ± 0.189</td>
<td>0.874 ± 0.075</td>
<td>0.052 ± 0.050</td>
</tr>
<tr>
<td>Supervised baseline</td>
<td>114</td>
<td>0.644 ± 0.080</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Self-supervised baseline</td>
<td>114</td>
<td>0.850 ± 0.005</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 8: Full display of the inferred images from our models trained on different number of samples. On the right are shown the reference outputs from the baselines [7] - all trained on 114 samples.

Table 3: Experiment results for triplet loss diffusion networks with different margin values.

<table>
<thead>
<tr>
<th>Margin</th>
<th>F1-score</th>
<th>Jaccard</th>
<th>SSIM</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>0.7480 ± 0.1549</td>
<td>0.6198 ± 0.1823</td>
<td>0.8471 ± 0.0790</td>
<td>0.0708 ± 0.0504</td>
</tr>
<tr>
<td>8.0</td>
<td>0.7364 ± 0.1603</td>
<td>0.6065 ± 0.1880</td>
<td>0.8491 ± 0.0753</td>
<td>0.0696 ± 0.0457</td>
</tr>
<tr>
<td>16.0</td>
<td>0.8149 ± 0.1744</td>
<td>0.7181 ± 0.2087</td>
<td>0.8749 ± 0.0713</td>
<td>0.0548 ± 0.0503</td>
</tr>
<tr>
<td>24.0</td>
<td>0.8369 ± 0.1610</td>
<td>0.7470 ± 0.2003</td>
<td>0.8952 ± 0.0659</td>
<td>0.0472 ± 0.0514</td>
</tr>
</tbody>
</table>

fully self-supervised, and even though they yield a comparatively high trainability and generalizability, they are still reliant on the quantity of annotated data and do not reach the robustness of fully self-supervised architectures.

This conclusion is also evident by the qualitative analysis, displayed in Fig. 8, where sample output segmentations are displayed from each trained model. While our model learns to perform satisfactory segmentation even with 50 training samples, it is clear that...
Margin | Original | Ground Truth | Timestep 1 | Timestep 2 | Timestep 3 | Timestep 4 | Timestep 5
---|---|---|---|---|---|---|---
4.0
8.0
16.0
24.0

Figure 9: Display of the inferred image from the first 5 sampling timesteps for each value of the margin.

Figure 10: Trainability of the triplet loss diffusion network compared to the supervised baseline.

Figure 11: Trainability of the triplet loss diffusion network compared to the self-supervised baseline [7].

it struggles with blurry and ambiguously-bounded areas. It only starts to handle ambiguity when trained on 500 samples, while the self-supervised baseline produces robust and smooth segmentations with the insignificant training dataset size of 114.

6.3 Loss Function Comparison
In order to evaluate if the triplet loss is applicable to the proposed architecture and task, we compared its performance to equivalent models trained with a more standard loss objective - the Mean Square Error. The results of the comparison are displayed in Table 2. It is evident from the latter analyses and from the visual inspection of the produced images (see Fig. 8) that both loss objectives produce images with similar segmentation quality, and the triplet loss is undoubtedly competitive in terms of applicable loss functions. Nevertheless, MSE marginally outperforms the triplet loss for all training dataset sizes, as measured based on the F1-score. This suggests that MSE is still the more appropriate loss for this task, even with DDPMs, but the significance of the difference is subject to further investigation.

However, it is noteworthy that the triplet loss outperforms MSE on every other metric for the full 500 training images. A higher SSIM and Jaccard accuracy suggests that the triplet loss models recognizes more structurally meaningful details but often bears the burden of over-segmentation because of the recognition of dead/dying organoids. This results in a high number of false positives in certain samples, which may explain the comparatively low F1-score.

6.4 Ablation Study
An ablation study was performed over the triplet loss diffusion network paradigm with the objective of determining the optimal
would be needed. This was tested by measuring the F1-score of the

The objective of the present research study was to determine whether

This results suggests that the Diffusion paradigm of this approach

networks are significantly more performant and trainable than

and analyses brought about evidence that triplet loss diffusion

7.1 Conclusions

7.2 Potential Problems and Future Research

8

Figure 12: Performance of the pre-trained triplet loss Diffu-

Figure 12: Performance of the pre-trained triplet loss Diffu-

set of hyperparameters. Specifically, the effect of the triplet loss

margin value on the performance of the model was measured. We

also ablated the model with respect to the number of sampling

The results of the former study are displayed in Table 3. As

shown from the Table, higher margin values yield a higher F1-score

of the inferred segmentations. This was expected as higher margins

lead to a stronger distinction between each negative sample (the

original organoid image) and the positive sample (the ground truth

segmentation).

However, it was also expected that with smaller margins, the

sampling would be more gradual and more sampling timesteps

would be needed. This was tested by measuring the F1-score of the

segmentation at each timestep for every margin value. The results

are shown on Fig. 12. As expected, the sampling becomes more

unstable with higher margins, even though at the first timestep, the

highest margin leads to the best performance.

A display of the produced segmentations at each timestep is

also available on Fig. 9. It is evident from the latter Figure that the

segmentation quality drops significantly after the first timestep.

This results suggests that the Diffusion paradigm of this approach

may be redundant and a U-Net trained with triplet loss would

produce similar results.

7 DISCUSSION

7.1 Conclusions

The objective of the present research study was to determine whether
diffusion networks, trained with triplet loss, are applicable to the
problem of semantic segmentation. Furthermore, the study aimed to
investigate whether the diffusion paradigm guided by a triplet loss
objective gives rise to more generalizable, trainable, and performant
models as compared to alternative supervised and self-supervised
approaches. Finally, we sought to ascertain the optimal set of train-
ning and sampling hyperparameters for the proposed approach.

Our results conclusively showed that the triplet loss is applicable
to this task and to the diffusion network architecture, although
it may not be optimal. Furthermore, the performed experiments
and analyses brought about evidence that triplet loss diffusion
networks are significantly more performant and trainable than
supervised approaches. However, fully self-supervised approaches

still marginally outperform the proposed architecture and with
fewer training examples (hence, they are also more trainable than
our model). Nevertheless, our model yields a significantly higher
median score, suggesting that it is more performant with typical
images but encounters major difficulties with ambiguity, specifically
when dead or dying organoids are present in the input image. Thus,
triplet loss diffusion networks are trainable and robust, but possibly
unreliable.

It was also determined that higher margin values lead to less
stable diffusion but overall better segmentation quality. This can be
attributed to the better distinction between the original images and
the segmentation, guaranteed by higher margin values. However,
the diffusion becomes less stable since there is a larger difference
between each sampled image at every timestep. In fact, even for
smaller margin values, it was discovered that the performance
deteriorates significantly after the first sampling timestep. This is
evidence to suggest that the diffusion element of this approach
is redundant and causes instability. Therefore, a U-Net with an

equivalent architecture to the tested diffusion network’s backbone
should lead to similar results.

In the larger context of the present research’s implications, it
was shown that the triplet loss is applicable to tasks beyond image
classification, for which it was designed. Specifically, it is fully
compatible with architectures, designed for semantic segmentation.
However, it is not clear if it is optimal for the latter task. Moreover,
the present study brings about evidence that models trained with
triplet loss may reach a low convergence plateau, since triplet loss
does not guarantee that the optimal solution would be reached
by its minimization (unlike more standard loss objectives such as
MSE).

Another broader conclusion that can be drawn from our findings
is that self-supervised architectures are both more trainable and per-
formant than supervised alternatives. The diffusion network used
in this study can be considered semi-supervised. It outperforms
supervised methods but is outperformed by fully self-supervised
ones, both in terms of trainability and accuracy. This finding con-
tributes to the larger body of evidence that stronger reliance on the
data leads to less generalizable and trainable models in the context
of Machine Learning.

7.2 Potential Problems and Future Research

As mentioned earlier, the most significant potential problem of the
triplet loss Diffusion Models may be its apparent lack of reliability.
The loss optimization is shown to be chaotic and unstable, which
may be an inherent property of the proposed model but may also
suggest a methodological problem. However, it is likely that this
can be attributed either to the architecture of diffusion networks
and its loss weighting scheme (see Section 4.2) or to the relatively
small number of training images and epochs which do not allow
for a further stabilization of the training process.

Another potential problem, as mentioned before, can be that the
model may reach a suboptimal convergence plateau, due to the
nature of the triplet loss. However, there is not sufficient evidence
to make this claim from the study’s findings and further experi-
mentation is necessary. Finally, some of our results suggest that
the diffusion element of our proposed approach may be redundant as the segmentation quality deteriorates after the first sampling timestep. These, however, are probably not methodological problems but valid findings from our experimentation with diffusion networks and triplet loss. Nonetheless, these speculations need to be validated with further experimentation.

In terms of avenues for future research, it would be valuable and relevant to train the triplet loss diffusion network on the full dataset, in order to evaluate whether the triplet loss facilitates a convergence plateau. Furthermore, the contribution of the diffusion element within this architecture can be investigated by comparing our model results to a U-Net with triplet loss. This would determine whether the diffusion paradigm is redundant, as some of our results may suggest.

Moreover, the number of timesteps during training can also be ablated in order to determine how they influence the training. If such a small number of sampling timesteps are needed to produce a high-quality segmentation, then it is possible that fewer timesteps are needed during training as well. This would lead to faster training sessions and may yield similar results.

Finally, it would be worthy to study how generalizable the proposed architecture would be to the broader task of semantic segmentation. In order to do this, the proposed architecture can be trained and evaluated on a public dataset such as BRATS2020 or MoNuSeg. This would enable a comparison between our model and a larger pool of scientific papers exploring the use of deep learning for the general task of semantic segmentation.

ACKNOWLEDGMENTS

We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high-performance computing cluster. We also thanks the University Medical Center Groningen (UMCG) for sharing with us the data.

8 FUNDING

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 812968.

REFERENCES


