university of groningen / faculty of science and engineering

Supervised Versus Self-Supervised: Which is Better for Biomedical Image Segmentation?

Bachelor's Project Thesis

Eric Brouwer, s3934640, e.b.brouwer@student.rug.nl,
Daily Supervisor: Asmaa Haja, a.haja@rug.nl
Main Supervisor: Prof. Lambert Schomaker, l.r.b.schomaker@rug.nl

Abstract: The process of annotating relevant data in the field of digital microscopy can be both time-consuming and especially expensive due to a hard requirement in technical skills and personal knowledge. Resulting from this, large amounts of microscopic image data sets remain unlabeled leaving them unused in deep learning. In contrast, large sets of inherent information can be drawn from the data that remains unlabeled. Self-supervised learning (SSL) is a promising key to solving this issue through feature learning under a pretext task, which is then transferred to a downstream main task - in our case image segmentation. Regarding this task, a ResNet50 U-Net was first trained to restore images of liver progenitor organoids from augmented images obtained by random pixel drop, blurring, and sobel filtering. Using a Structural Similarity Index Metric (SSIM) loss as well as the SSIM combined with Mean Absolute Error (L1) loss, both encoder and decoder were trained in tandem. The weights were transferred to another U-Net designed for segmentation with frozen encoder weights, where they were trained with the Binary Cross Entropy, Dice, and Jaccard loss. Paired with this, we also used the same U-Net to train two supervised models, one utilizing the ResNet50 encoder, and the other a simple CNN. Results showed that SSL models using a 25% pixel drop or image blurring augmentation performed better in comparison to the other augmentation techniques paired with the Jaccard loss. When trained on 114 images for the main task, the SSL approach outperforms the supervised method achieving an F1-score of 0.85 with higher stability, in contrast to the 0.78 scored by the supervised method. Furthermore, when trained with larger data sets (1.000 images), SSL is still able to outperform the supervised achieving an F1-score 0.92, contrasting to the score of 0.85 for the supervised method.

1 Introduction

With the advances in high throughput imaging technology, it is currently possible to produce a large number of microscopic images in a short period of time (Haja and Schomaker, 2021). Comprehensive analyses of biological images are required for medical diagnosis and illness comprehension (Zhang et al., 2020). Detecting diseases through manually analyzing the rich biological information in microscopic images is challenging since it is time-consuming, demands domain knowledge in the field, is biased to the individual human experts and thus not entirely accurate, and is an exhausting task that can lead to fatigue (Adhikari et al., 2021) (Zhu et al., 2021). Accordingly, research in the biological field can be a slow and arduous endeavour. Methods from the field of deep learning can be used to automatically extract relevant information from biological images.

Deep learning has recently found major success in the automation of data processing, manipulation, and understanding of data (Dargan et al., 2020). In regards to biomedical image processing, deep networks have been able to demonstrate exceptional performance in tasks such as classification (Mai et al., 2022), detection, and segmentation (Vu et al., 2019) through **supervised** learning. Their performance and success rely heavily on the use of labelled or annotated data (Sharma et al., 2021). The process of annotating relevant data in this context still involves manual labour, leaving researchers with a similar issue where large

amounts of biomedical image datasets remain unlabeled, keeping them nearly useless for their intended tasks. In contrast, large sets of inherent information can be drawn from the data that remains unlabeled (Jaiswal et al., 2020). **Self-supervised learning** (SSL) is a promising key for processing and extracting relevant information from datasets consisting of a higher proportion of unlabeled images than annotated images (Chen et al., 2019) (Azizi et al., 2021). Deep learning models using the SSL paradigm allow for models to familiarize themselves with the data of interest, where the knowledge is transferred to a supervised approach that would only need very little training afterwards.

The intention of this work is to employ the SSL paradigm for biomedical imaging, specifically on organoid culture images. In essence, organoids are self-organizing three-dimensional structures grown from *in vitro* stem cells, with the ability of mimicking its *in vivo* tissue counterpart (de Souza, 2018). This ability can be used as a powerful tool. For instance, it can be used to indicate different diseases based on changes in their morphology (shape and structure) (Kretzschmar, 2021). Precise measurements of organoids' morphology can be achieved by segmenting organoid objects in the image dataset.

With this aim in mind, this paper explores the ability to use the SSL technique to segment organoid culture images, as well as to compare the supervised with the SSL approach in order to observe the amount of sufficient data required to develop a robust model.

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 the organoids research, semantic segmentation, supervised learning, and self-supervised learning. Section 3 describes the method of the investigation where the organoids data, loss functions, and the supervised and SSL frameworks are discussed. Section 4 is reserved for the experimental design, describing the data distribution and the implementation details. Section 5 presents a discussion of the results and lastly, section 6 the conclusion and future work.

2 Related Works and definition

2.1 Organoids Research

Studies conducted on both animals and humans regarding organs or *in vivo* tissues can be slowed down or limited due to a range of expensive costs, limited resources, and ethical issues (Rossi et al., 2018). This led to the further development of *in vitro* stem cell research, allowing researchers in this field to overcome the previously mentioned concerns (Graudejus et al., 2018). One of the *in vitro* stem cells is the organoid. Organoids are self-organizing three-dimensional structures grown from *in vitro* cells, having the ability to mimic its *in vivo* organ counterpart (Tuveson and Clevers, 2019), (de Souza, 2018), (Kratochvil et al., 2019), (Corrò et al., 2020). This ability to mimic *in vivo* organ tissues is incredibly powerful, leading to a large range of applications of organoids in, for example, modelling organ development and disease (Rossi et al., 2018), cancer research (Drost and Clevers, 2018), regenerative medicine (Marchini and Gelain, 2022), or personalized medicine and drug discovery (Wang and Hummon, 2021). An important aspect of research in drug development using organoids is the ability to measure their morphological changes when responding to the introduction of external treatments (Karolak et al., 2019). It is then crucial to get these measurements accurate, as a drug's effectiveness depends on the morphological change (Karolak et al., 2019). However, manually measuring the volume of each organoid can be a time-consuming operation that may act as a bottleneck to the whole research process. Hence, models from the deep learning field are introduced to resolve this issue.

2.2 Semantic Segmentation

Semantic segmentation is a deep learning technique in which each pixel of an image is associated with its representative class label (Ramesh et al., 2021). This is especially useful in measuring both the morphological characteristics and changes of objects in an image, as groups of pixels that have been identified to be in the same class will represent the same object. Essentially, this

technique allows for the ability to automatically highlight useful contextual information in an image. The deep learning model, built for semantic segmentation, would take an input image, and return an output image with the exact dimensions but where the pixel values are sized to be between 0 and the number of class labels. In the case of organoid semantic segmentation, this would be either 0 and 1, denoting the background or the organoid itself, respectively.

Multiple types of software already exist that attempt to segment organoid culture images. One example is the software package OrganoSeg (Borten et al., 2018), which provides an intuitive, graphical user interface for quantifying transmitted-light images of 3D spheroid and organoid cultures. However, this software requires some manual work from the user to define and finetune thresholds and parameters used for separating the foreground from the background. Another example is the OrganoID (Matthews et al., 2022), a robust image analysis platform that automatically recognizes, labels, and tracks single organoids, pixel-by-pixel, in bright-field and phase-contrast microscopy experiments using deep learning. This software employs Sobel operators, Gaussian filter, and watershed for detecting single organoid. These techniques are highly affected by the image quality (e.g. change in brightness) and cannot be generalized to all microscopic data. Another example is the deepOrganoid model, which is based on a deep learning technique that can be used as a fully automatable analytical tool for high-throughput screens that rely on organoid cultures (Powell et al., 2022). Although researchers in the organoid field can utilize this model by retraining it on their dataset, the problem of possessing sufficiently large labelled data beforehand to train the model is still an issue in this field. All these tools cannot be generalized for various organoid datasets with a limited amount of annotation as they are trained using supervised learning.

2.3 Supervised Learning

Deep learning models based on the supervised method are most common in the context of image processing (Ramesh et al., 2021). The term "supervised" refers to the type of training where some form of instruction is provided to the model throughout the training process (Sen et al., 2020). Generally, data would consist of some form of input which is paired with its target output label where the model will train itself to approximate a mapping function from the input to its associated output (Cunningham et al., 2008). Such methods, however, require large amounts of annotated data in order to make robust approximations that work well on data that the model has never seen (Wang et al., 2019). Despite having demonstrated astonishing prowess in tasks such as facial recognition or object classification, medical fields struggle to achieve similar success through such means (Razzak et al., 2018). This is due to the fact that supervised methods require enormous amounts of data with manual annotation by humans, which can become rather expensive or time-consuming in the medical field due to a limited amount of expertise (Yu et al., 2016).

2.4 Self-Supervised Learning

The self-supervised learning method attempts to address the issue of having limited annotated data by extracting relevant features from unlabelled data (Wang et al., 2022). The self-supervised approach can be subdivided into two tasks: the pretext task and the main task. Firstly, the **pretext task**, essentially allows a model to familiarize itself with the data under the presumption that convolutional neural networks extract various levels of information from their layer counterpart. The low-level features, such as the texture or gradient of an image, are captured by the shallow layers of the CNN, whilst the deeper layers are responsible for capturing the high-level features; being the semantic information (Li et al., 2017). Utilizing this property, the pretext task defines an image transformation problem for the CNN to solve by predicting the properties of that same transformation (Misra and Maaten, 2020). Examples of this could be a context prediction problem shown in figure 2.1 (Doersch et al., 2015), or a rotation prediction problem as shown in figure 2.2 (Gidaris et al., 2018). Resulting from this, the model is then able to learn the low to mid-level features from data that remain unlabelled.

Secondly is the **main task**, which is responsible for capturing the higher-level features (being the

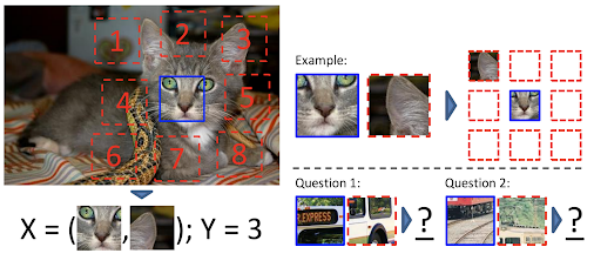

Figure 2.1: Pretext Task: Context prediction problem - An image is divided into tiles and given numeric labels. The center tile (highlighted in blue) is used as an anchor point, where a surrounding random tile is selected. The CNN has to solve for the tile label, given that an image is compared to the anchor point. Image source: (Doersch et al., 2015)

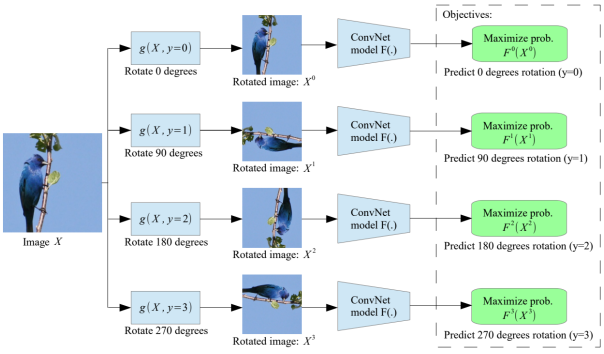

Figure 2.2: Pretext Task: Rotation prediction problem - The network is provided an image with rotation (1) 0 (2) 90 (3) 180 (4) 270 degrees. It must predict the degree of rotation for each input image based on the provided the label. Image Source: (Gidaris et al., 2018)

object of interest itself) where the quality of the features that have been learnt are evaluated. This is done by re-purposing the learnt information from the pretext task to a main task of interest, better known as **transfer learning** (Misra and Maaten, 2020). Since the low to mid-level features have generally been learnt in the pretext task, the model now only needs to learn the high-level features, which can be done by using the limited amount of labelled data to perform supervised learning. To summarize, the self-supervised training method removes the full dependency on labels to a partial dependency (Zhang et al., 2021).

To the authors' knowledge, no work exists in the literature that explores the detection and segmentation of organoid images using the self-supervised concept, which also shows the novelty of this work.

3 Method

Ultimately, this work aims to construct a network with the ability to learn effective representations of the data to accurately segment images given a limited amount of annotated training data. The self-supervised training method was utilized as an approach to accomplish this task and is further compared with two supervised approaches. The various network architectures employed in the project will be discussed in detail in sections 3.1 and 3.2. Section 3.3 presents the data and introduces the different augmentation techniques used on the data. Section 3.4 introduces all the loss functions that are compared in each learning stage. Lastly, sections 3.5 and 3.6 describe the self-supervised and supervised frameworks, respectively.

3.1 U-Net backbone

The U-Net encoder-decoder network (Ronneberger et al., 2015) acts as the backbone of the model. It was initially developed for biomedical image segmentation and gained popularity due to its ability to perform well with a minimal number of training samples (Liu et al., 2020). The network itself consists of a contracting path (left half of Figure 3.1) and an expansive path (right). The contracting path, termed the encoder, consists mainly of convolutional and pooling layers and is designed to capture the context of the images being passed through. At each stage of the contracting path, an unpadded 3x3 convolution is applied two times, followed by a rectified linear unit (ReLU). The spatial resolution is then reduced in half through a 2x2 max-pooling layer, which doubles the number of features. As shown in Figure 3.1, the convolution with relu and pooling block is repeated four times until it reaches the next stage - the expansive path. This component, also named the decoder, mirrors the encoder and performs localization through deconvolution. Up-sampling is performed on the feature maps, and a 2x2 deconvolution follows allowing

the feature maps to be doubled again. Lastly, these feature maps are then concatenated to their corresponding input feature maps, forming the skip connection (illustrated by the horizontal grey lines).

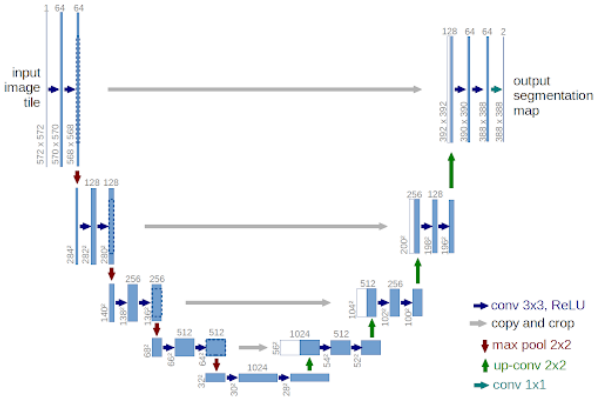

Figure 3.1: U-Net architecture. The encoding component is illustrated by the left side of the U shape, whilst the right characterises the decoder. Source: (Ronneberger et al., 2015)

3.2 Encoder

The ResNet (short for residual net) architecture proposes a novel method to address the critical vanishing gradient problems that affect deeper networks in regards to the convergence of the optimization function (He et al., 2016). This was done by redefining the traditional sequential convolutional layers in order to learn the residual parameters where a residual learning block is introduced. This learning block provides feed-forward connections that map the identity from the input to the output, as shown in Figure 3.2. In this case, the 50-layer variant of the ResNet architecture was chosen - henceforth called ResNet50.

The original U-Net architecture with a simple convolutional neural network (simple CNN) as encoder, as described in section 3.1, is compared to a complex U-Net architecture having a ResNet50 as its encoder. Here, the necessity of the ResNet50 architecture is evaluated for the SSL and supervised cases.

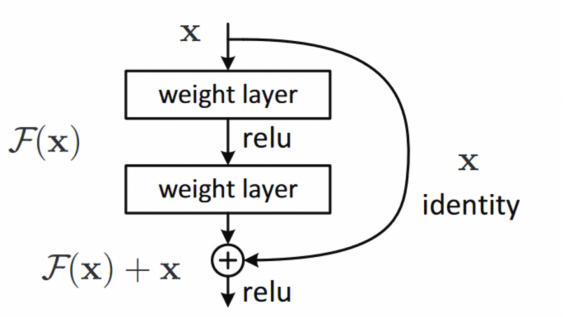

Figure 3.2: The ResNet residual learning block: skip connection with input x, transformation $F(x)$, and output $F(x) + x$ representing the concatenated information. Source: (He et al., 2016)

3.3 Data & Augmentation

The data used in this study consists of liver progenitor organoids, provided by the University Medical Center Groningen (UMCG), the Netherlands. The organoid images were captured by a special microscope across five different time points - ranging from 0 up to 96 hours with 24-hour intervals. Furthermore, the organoids were left in two growing conditions - **(1)** the liver progenitor organoids are grown in a complete medium, **(2)** the organoids were grown in a medium where all amino acids have been removed (essential for their growth). Resulting from this, a total of 10 CZI images were captured. CZI refers to a 3D image consisting of 2D image slices captured at different depths from the organoid culture (Figure 3.4). In this case, each CZI file has 14 2D slices, where each slice has an image size of 3828x2870 pixels. An average of 4 middle slices were used because the upper and lower slices contained little relevant information. Semantic segmentation was then performed on all the selected images using the OrganelX* service. A manual correction also took place to confirm that most organoids have been correctly segmented. The initial image sizes have a high resolution (3828x2870), which is incredibly large for any DL network to process efficiently. As a result, smaller images, called crops, were created by a sliding window technique as explained in Figure 3.5. Crops of a window size 636x636 pixels were created with a window

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

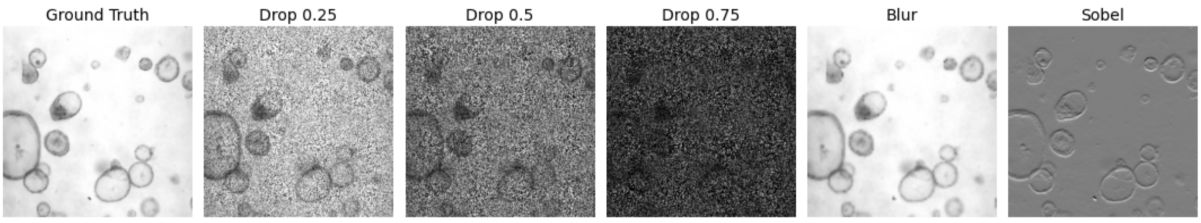

Figure 3.3: Three augmentation techniques. From left to right: Ground truth image, 25% pixel drop, 50% pixel drop, 75% pixel drop, Gaussian Blurring and Sobel Filtering.

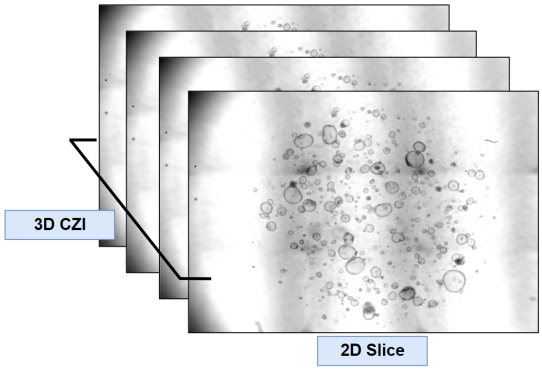

Figure 3.4: An example of a CZI: A 3D image made up of 2D slices at different depths of the organoid culture.

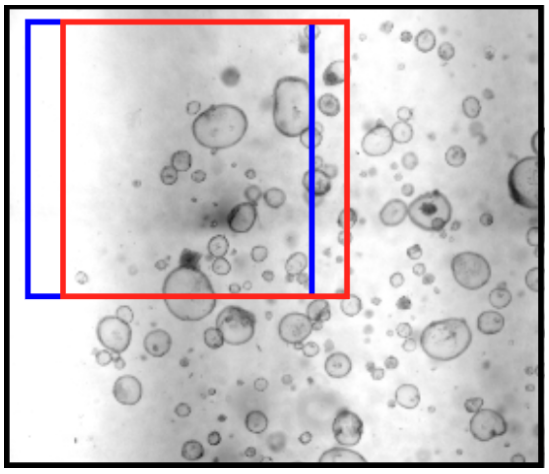

Figure 3.5: Sliding Window Crop: A cropping technique where a window is selected and cropped, as shown by the blue square. The window is then moved by N amount of pixels, and a new crop is made, shown by the red square. As a result, the window is moved across the entire image, where all parts of the image have been cropped.

increment of 60 pixels per step. These images were later resized to be 320x320 pixels to reduce the model training time. Images with less than 5% relevant information (presence of organoids) were also removed. Lastly, image rotation was also used as an augmentation technique to increase the number of total images, resulting in around 100.000 cropped and augmented images being used.

To perform the pretext task, explained in sections 2.4 and 3.5, three augmentation techniques were performed on the images: **(1)** Pixel drop: random noise is added to an image by randomly dropping pixels from the image, **(2)** Gaussian blurring: image resolution is cut in half by performing a gaussian blur function, **(3)** Sobel filtering: a Sobel operation is applied on the image resulting in an emphasis on object edges. Figure 3.3 displays a randomly selected image from the dataset with all augmentation techniques applied.

3.4 Loss Functions

In the context of DL, the loss function is a method of evaluating how suitable a model is at predicting values given an input. Typically, a loss function would calculate the distance between the target output and the predicted output. The distance is then used to update the model's weights. These weights will continuously update until the distance (the loss) between the target and predicted output converges.

Sections 3.4.1 and 3.4.2 describe loss functions used for the pretext task, whilst sections 3.4.3, 3.4.4, and 3.4.5 describe loss functions used in the

main task.

3.4.1 SSIM

The Structural Similarity Index Metric (SSIM), proposed by Wang et al. (2004), measures the similarity between two given images. An image is divided into various windows, where x and y indicate their respective window of the two images with shared sizes $N \times N$. The score is calculated as shown in equation 3.1.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3.1}$$

With: μ_x the average of x; μ_y the average of y; σ_x^2 the variance of x; σ_y^2 the variance of y; σ_{xy} the covariance of x and y; c_1 and c_2 as constants to stabilise cases with weak denominators (i.e. zero). The SSIM value can then be used to compute the SSIM loss as shown in equation 3.2.

$$L_{SSIM}(x,y) = 1 - SSIM(x,y) \tag{3.2}$$

3.4.2 SSIM-L1

In some cases, the SSIM loss suffers from sensitivity biases. Resulting from this, when images are restored, changes in colour or brightness can be observed (Zhao et al., 2015). In contrast, the mean absolute error, also known as the L1 loss (shown in eq. 3.3), suppresses this factor more heavily. Here n indicates the number of pixels, Y the target output, and $\hat{Y}$ the model's predicted output.

$$L_{MAE} = 1 - \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{Y}_i| \tag{3.3}$$

In principle, to get clearer image restorations, the L1 loss is combined with the SSIM loss function in a symmetrical manner, and is shown in eq. 3.4.

$$L_{SSIM-L1} = \frac{1}{2}L_{MAE} + \frac{1}{2}L_{SSIM} \tag{3.4}$$

3.4.3 Binary Cross Entropy

Binary cross entropy (BCE) (Jadon, 2020) compares the probability of the model's predicted output class $\hat{Y}$ to the actual class label Y within a range of 0 and 1, as shown in equation 3.5. Due to this nature, it can then be used for the task of binary pixel-wise classification necessary for the segmentation task, as mentioned in section 2.2. In this case, n refers to the number of pixels present in the image.

$$L_{BCE}(Y,\hat{Y}) = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot log\hat{Y}_i + (1-Y_i)\cdot log(1-\hat{Y}_i)) \tag{3.5}$$

Generally, with cross-entropy functions, the gradients with respect to the logits produce smoother loss values allowing for better stability in training when compared to other loss functions in the same domain (Jadon, 2020).

3.4.4 Dice

Dice loss is a commonly used loss function in the context of semantic segmentation (Yeung et al., 2022). The similarity between the output image $\hat{Y}$ and target output Y is computed as shown in equation 3.6. Here, ϵ denotes a constant value for cases with a weak denominator, known as a smoothing factor (Li et al., 2019).

$$L_{dice}(Y,\hat{Y}) = 1 - \frac{2 \cdot \sum Y \cdot \hat{Y}}{\sum Y^2 + \sum \hat{Y}^2 + \epsilon} \tag{3.6}$$

In contrast to cross-entropy functions, the dice metric can cause gradients to blow up to large numbers, often resulting in unstable training (Eelbode et al., 2020). However, dice losses are more robust when presented with imbalanced datasets, which is relatively common in semantic segmentation; typically, the background accounts for a more significant portion of the pixels than the object of interest. This is also the case for organoid images.

3.4.5 Jaccard

The Jaccard loss, also referred to as the Intersection over Union (IoU), is less commonly used than the dice loss but is also a powerful tool for semantic segmentation (Bertels et al., 2019). Here, the sum of the product between the predicted output $\hat{Y}$ and target output Y is computed, then divided by its union as shown in equation 3.7. Again, ϵ is used as a constant to prevent a zero division.

$$L_{IoU}(Y, \hat{Y}) = 1 - \frac{\sum(Y \cdot \hat{Y})}{\sum(Y + \hat{Y}) - \sum(Y \cdot \hat{Y}) + \epsilon} \tag{3.7}$$

The Jaccard loss suffers from a similar problem to the dice loss regarding blowing up gradients. However, like the dice loss, it works well with an imbalanced dataset with the addition of scale invariance, granting relevance to smaller objects (Bertels et al., 2019). The ability to include such smaller objects is pertinent to the organoids data set as images typically have both large and small organoids spread across the image.

3.5 Self-Supervised Framework

The self-supervised framework consists of two phases, as demonstrated in Figure 3.6. A pretext task is essentially designed to push the model to learn the semantic features of the input images; by allowing the model to 'pre-train' on the data, the model is able to familiarize itself with the data. The proposed pretext task is to perform image restoration on augmented images (section 3.3). For this task, a U-Net with a ResNet50 encoder as the backbone paired with the decoder are trained in tandem to restore an augmented input image. The data is augmented through three techniques as described in section 3.3. The output of the U-Net is compared to the ground truth image where the SSIM and SSIM-L1 losses are computed as described in section 3.4. The concept for this task is to train the network to attempt to generate organoid features, which would set up the network's weights for the second phase. In the second phase, the learnt weights are transferred to perform the main task of segmentation on the images of organoids, where the encoder weights are frozen and only the decoder is re-trained using the loss functions described in sections 3.4.3, 3.4.4, and 3.4.5.

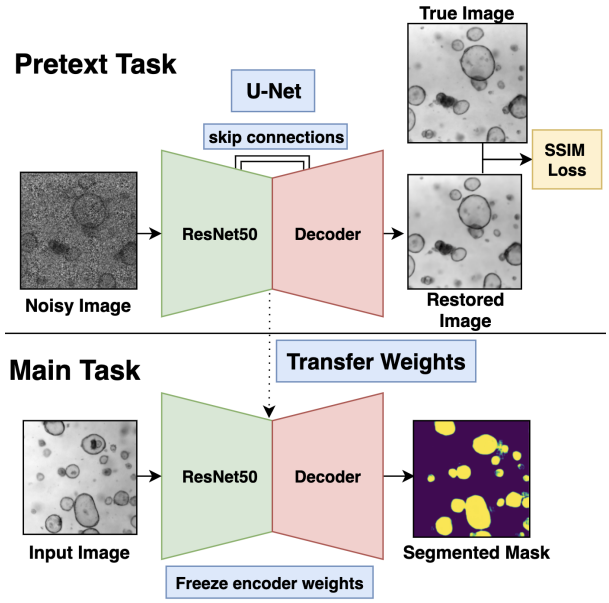

Figure 3.6: Self-supervised pipeline for organoid segmentation. Top: Pretext task - The U-Net model consists of an encoder, ResNet50, and a decoder with a decoder with a skip connection. The ResNet50 is trained to restore augmented images to their original form. The restored image is compared to the ground truth image, and the SSIM or SSIM-L1 loss is computed. Bottom: Main task - The same network as in the pretext task, yet, with a frozen encoder. The decoder learns to segment the ground truth images.

3.6 Supervised Framework

The supervised framework consists of two approaches: **(1)** both the ResNet50 and simple CNN encoders were used in a supervised manner. In order to hold a fair comparison with the self-supervised approach, the same U-Net architecture as shown in Figure 3.6 was chosen for this, where both encoder and decoder were trained with randomly initialised weights. Approach **(2)** employs the identical encoders, however, only the decoder is trained; both encoder and decoder weights are still randomly initialised. In other words, the encoder weights are frozen and were not updated during training (i.e. back-propagation).

All other parameters and settings are kept the same for all approaches and identical to the SSL approach for the semantic segmentation task. In essence, the main task for the SSL approach is almost identical to the supervised approach, with the exception being that in case **(1)** the encoder is also trained.

4 Experimental Design

4.1 Data Distribution

The data set, as described in section 3.3, was shuffled and randomly divided into three subsections in an equal distribution from each CZI file: $Train_{pretext}$, $Train_{main}$, and *Evaluation*. As shown in the top half of Figure 4.1, from the roughly 100.000 crops, 40% was reserved for $Train_{pretext}$, 40% was taken for $Train_{main}$, and the remaining 20% for *Evaluation*.

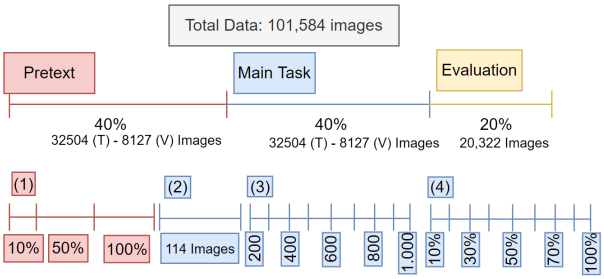

Figure 4.1: Visual of the distribution of the data, with three different training scenarios. (1) indicates the distribution only for the pretext task. (2), (3) and (4) indicate the data for the main task used in both SSL and supervised approaches. (T) denotes training while (V) denotes validation.

The data set was separated in this manner prior to any training as to ensure that at each stage, the model is trained on images it has never seen before; this is done to prevent over-fitting. Additionally, five-fold cross-validation was used to ensure stability during the training of each architecture. Four training scenarios, illustrated in Figure 4.1 and Table 4.2, follow from having a supervised and self-supervised framework: **(1)** the self-supervised framework is pre-trained on the $Train_{pretext}$ data set, which was further sub-divided into another three categories shown in table 4.1, as well as the red illustrations in Figure 4.1. This subdivision of the data set is done to observe the importance of pre-training the networks prior to transferring the knowledge and to see the performance of the models on a various number of images. In this stage, the performance of the augmentation techniques and loss functions will also be measured. In order to evaluate the performance in this regard, the model is then further trained for the main task on 114 images taken from $Train_{main}$ and evaluated on the *Evaluation* set. **(2)** Both supervised and self-supervised are trained on the $Train_{main}$ data set, which again was subdivided. In this case, to observe the self-supervised method's ability to accurately segment images given a small number of labelled data, both supervised and self-supervised are trained on 114 images. **(3)** To observe the point at which the supervised and self-supervised models have similar performances, both networks were also trained from 200 up to 1.000 images, with 100 image increments. **(4)** Lastly, the supervised model was trained on 10% up to 100% of the $Train_{main}$ data with 10% increments to observe performances with large data sets. It is important to note that the same images were used for all training scenarios to establish a fair analysis. A short summary of these four experimental cases is displayed in Table 4.2.

Table 4.1: The number of images used for training the pretext tasks. The first column lists the percentage of the total images considered, out of which the number of the images used to train and validate the model are described in the remaining columns.

Percentage of data	Train set	Validate set
10%	3250	813
50%	16252	4063
100%	32504	8127

4.2 Implementation Details

In this subsection, model implementation and hyper-parameter details are described. Regarding the model, the ResNet50 encoder and decoder architecture follow the same structure as discussed in (Zhang et al., 2018). Figure 4.2 illustrates a building block for the ResNet50 encoder, which is repeated four times for the encoding component and another four for decoding only with the addition of an upsampling layer between each block. Furthermore, a skip connection is formed between the encoder and decoder between each block. The encoder convolutional block starts with a 320x320 input matching the cropped image size, mentioned in section 3.3. Throughout the

Table 4.2: A short summary of the four experiment cases.

Four Experiment Cases
(1) SSL framework trained on $Train_{pretext}$, then 114 images from $Train_{main}$, and lastly *Evaluation* to observe performance of augmentation techniques, loss, and percentage of pretext training data.
(2) SSL and Supervised frameworks are compared with a minimal number (114) training images taken from $Train_{main}$. The models are evaluated on *Evaluation*.
(3) SSL and Supervised frameworks are compared by training from 200-1000 images (from $Train_{main}$) and evaluated on *Evaluation* to observe at which point the frameworks have similar performances.
(4) Supervised framework is trained from 10% to 100% of $Train_{main}$ with 10% increments then evaluated on *Evaluation*, and compared to the SSL to observe supervised performances on large data sets.

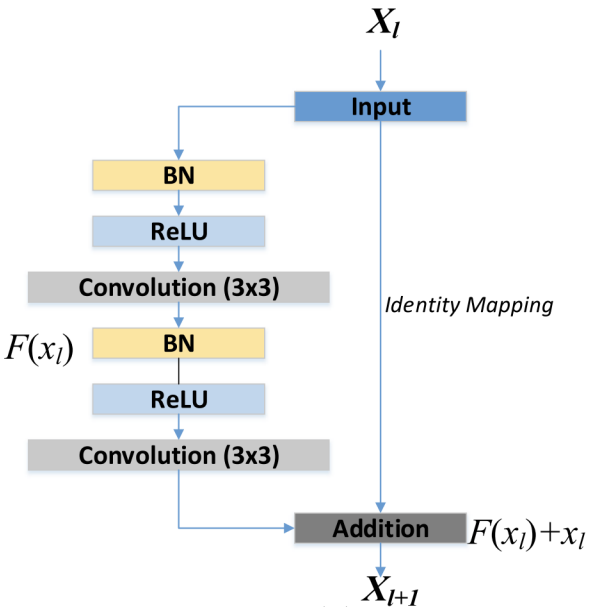

Figure 4.2: Building block for the ResNet50 encoder, with X_l denoting the input, BN for batch normalization, and ReLU for rectified linear unit. Image Source: (Zhang et al. 2018)

four convolutional layers, the input size is halved. Hence, the second layer has a size of 160x160, the third 80x80, and the last layer has a size of 40x40 pixels. The decoder performs this in the reversed order.

During the training phase for both the pretext and the main task, the model was trained over 50 epochs. The Adam optimizer was used as the learning scheduler with a learning rate of 0.003. The data was divided into a batch size of 16, and a seed value of 26 was used to ensure reproducibility when using random variables (i.e. shuffling the batches). Regarding constants used in the loss functions, the L_{SSIM} and $L_{SSIM-L1}$ functions had $c_1 = 0.01$ and $c_2 = 0.03$, whilst L_{dice} and L_{IoU} had $\epsilon = 0.0001$. Lastly, one Nvidia V100 GPU accelerator card was used for training all models.

5 Results

As mentioned in section 4.1, both supervised and self-supervised frameworks were evaluated on the same set of images that both networks have previously not seen before. Due to the nature of the pixel-wise binary classification task a confusion matrix was computed for each image that has been segmented. From this, the accuracy, precision, recall, F1-score, and Jaccard index was computed. The metric that we are most interested in is the **F1-Score**, in some cases called the harmonic mean. The F1-score penalizes large differences between precision and recall, which sets apart the desirable image segmentations from the undesirable ones.

5.1 Self-Supervised Framework

Figure 5.2 displays the results produced by the SSL framework first trained on $Train_{pretext}$, then on 114 images from $Train_{main}$, using the SSIM and SSIM-L1 for the pretext, then BCE, Dice, and IoU for the main task. Initially, it can be observed that in general, as the percentage of $Train_{pretext}$ data increases, the scores across all metrics also increase for all three main task loss functions. This confirms that increasing the amount of pre-trained data will have a positive influence on the main task.

Another observation that can be made is that

out of the five augmentation types, the 25% pixel drop and the blurring methods produce the best results reaching an F1-score as high as 0.85, which is considerably higher than the other three techniques where an F1-score of 0.75 was the highest. This would suggest that changing the image too strongly will make it more difficult for the network to restore the images, and as well as to extract the mid to low-level features. This is especially the case for the 75% pixel drop and Sobel filtering augmentations which scored the lowest, at 0.25 for the SSIM loss with 10% $Train_{pretext}$ and 0.4 for the SSIM-L1. In both cases, this relates to how both these augmentation techniques cause for the pixels to be affected the most. Despite this, by increasing the amount of training data for the pretext task, scores can still be made to improve as a general trend.

Regarding loss functions, it can be observed that the IoU loss, indicated by the green points, was able to achieve the highest scores reaching 0.85 when trained with 100% of $Train_{pretext}$ using either SSIM or SSIM-L1 for the pretext task. Furthermore, the IoU loss on average performed best regardless of augmentation technique with only the SSIM-L1 using the 50% pixel drop being the outlier. This confirms that the addition of scale invariance (section 3.4.5) is effective for the organoids data set due to the smaller organoids being scattered across each image.

Figures 5.3, 5.4, and 5.5 illustrate the segmentation masks generated for the various loss functions in combination with the augmentation techniques using the SSIM loss. The figures indicate visually how well the 25% pixel drop and gaussian blurring has performed when compared to the other augmentation techniques. Furthermore, it can be observed that as the percentage of pretext training data increases, the generated segmentation masks come closer to the true masks regardless of augmentation technique, once again confirming its influence over the main task. The SSIM-L1 loss has generated similar masks, which demonstrates a similar trend and was therefore left out in this case. Lastly, figure 5.1 demonstrates a good segmentation which is compared to a bad one where a stark contrast between performance can be observed.

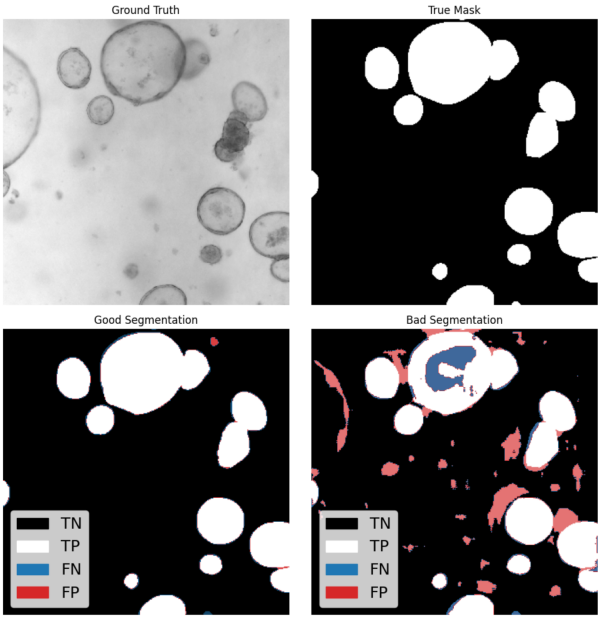

Figure 5.1: A comparison of segmentation masks generated by the SSL framework. Starting from the top left: Ground Truth image, True Mask, example of a good segmentation, and example of a bad segmentation. The good segmentation was generated using 100% of $Train_{main}$ with SSIM-L1 and IoU employing the blur augmentation, while the bad one is using the 10% SSIM and BCE with the Sobel filter.

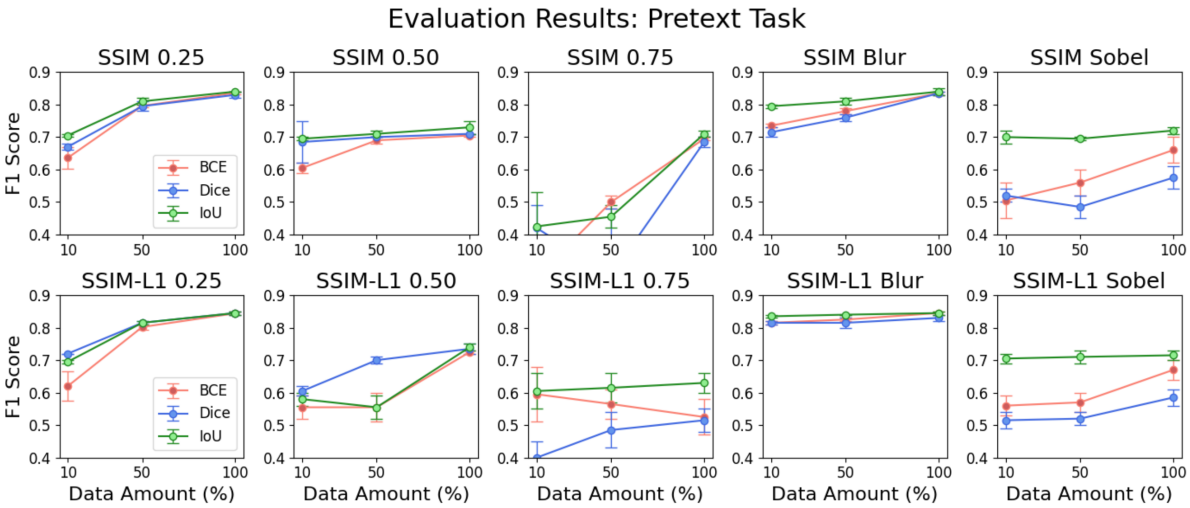

Figure 5.2: Evaluation of the self-supervised pretext task using F1-scores. The mean of the 5 folds for the BCE, Dice, and IoU loss functions were computed and is indicated by the red, blue, and green points respectively. The top row displays the scores using the SSIM loss of the pretext task, while the bottom row displays the SSIM-L1 loss.

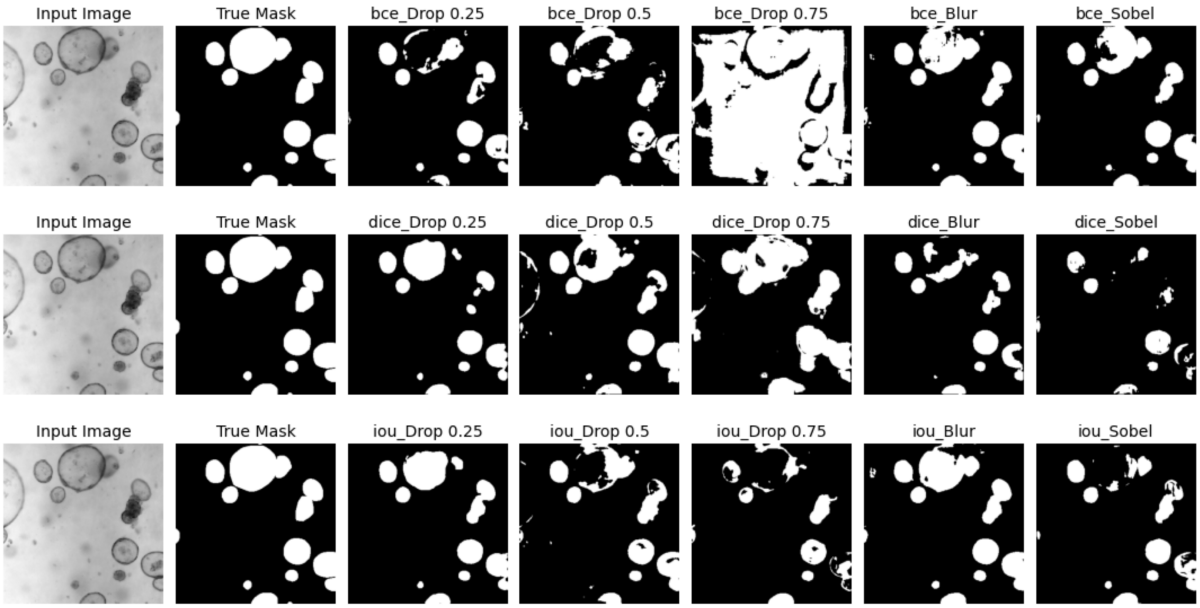

Figure 5.3: Segmentation masks generated by the SSL framework using 10% of $Train_{pretext}$. From left to right is: Input (Ground Truth) Image, True Mask, 25%, 50%, 75% Pixel Drop, Gaussian Blurring, Sobel Filtering. The top row illustrates the masks using the BCE loss, the middle row using the Dice loss, and the bottom using the IoU loss.

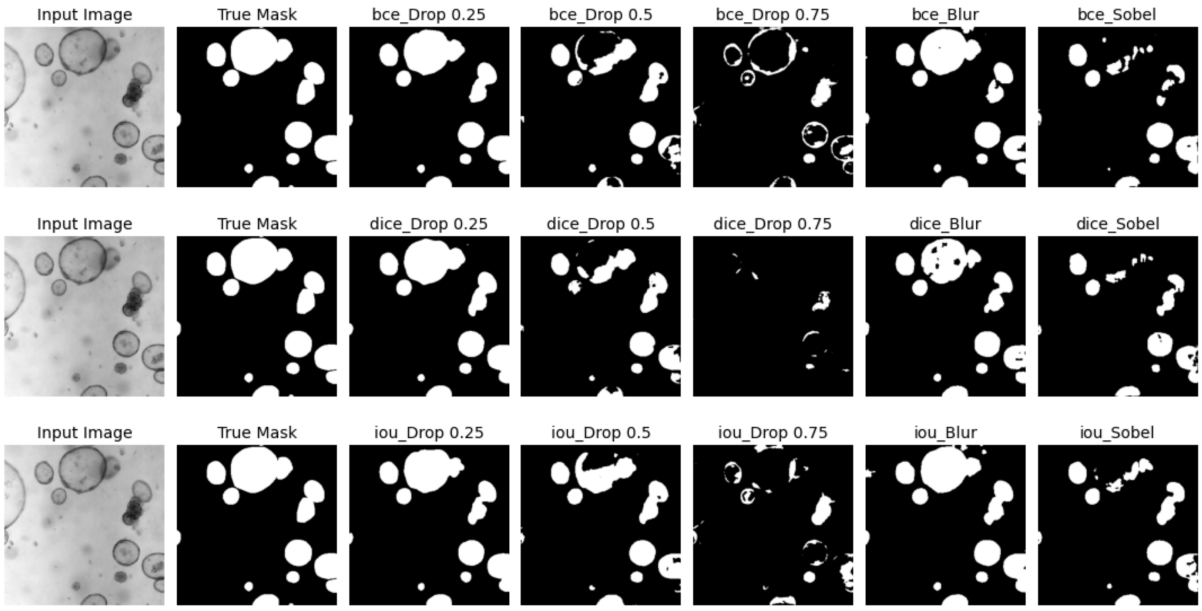

Figure 5.4: Segmentation masks generated by the SSL framework using 50% of $Train_{pretext}$. From left to right is: Input (Ground Truth) Image, True Mask, 25%, 50%, 75% Pixel Drop, Gaussian Blurring, Sobel Filtering. The top row illustrates the masks using the BCE loss, the middle row using the Dice loss, and the bottom using the IoU loss.

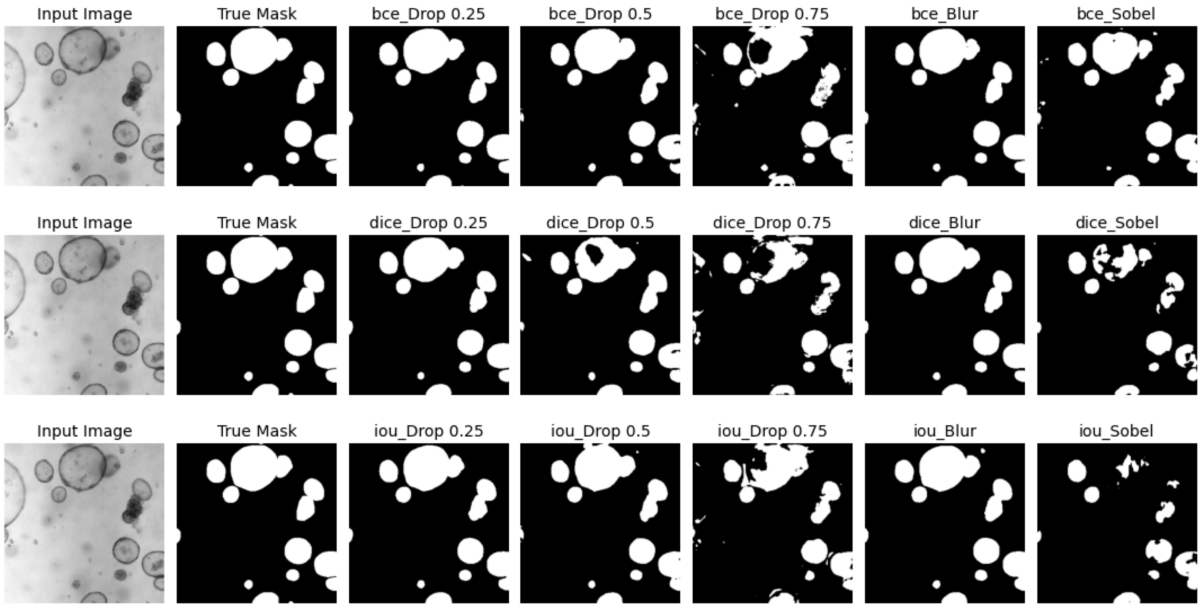

Figure 5.5: Segmentation masks generated by the SSL framework using 100% of $Train_{pretext}$. From left to right is: Input (Ground Truth) Image, True Mask, 25%, 50%, 75% Pixel Drop, Gaussian Blurring, Sobel Filtering. The top row illustrates the masks using the BCE loss, the middle row using the Dice loss, and the bottom using the IoU loss.

5.2 Comparing Frameworks

Tables 5.1, 5.2, and 5.3 display the best F1-scores for both self-supervised and supervised frameworks, trained on 114 images. Tables 5.4, 5.5, and 5.6 display the average F1-scores as well as the deviations for both frameworks trained on the same 114 images. The deviations can be used as an indicator for model stability, with larger deviations meaning lower stability and smaller deviations for higher stability. For the self-supervised approach, only the scores for the 25% pixel drop and blurring augmentation are displayed, as these two techniques reported the strongest results, as shown in section 5.1.

When observing the results of the blurring technique, shown in Table 5.2 a particular point of interest here, is that the 10% SSIM-L1 to IoU network, performs just as well as the 100% SSIM to BCE and Dice, all cases having an F1-score of 0.84. This confirms that suppressing colour changes and brightness with the addition of the L1 loss to the SSIM loss can be effective effective (mentioned in section 3.4.2). The 25% pixel drop, shown in Table 5.1, was able to achieve high F1-scores of up to 0.85 as well. However, when using 10% of $Train_{pretext}$, the scores range from 0.69-0.71 while the blurring technique in contrast was able to achieve a range of 0.73-0.84 which is substantially higher.

For the supervised approach shown in table 5.3, it is rather clear that the ResNet50 encoder outperforms the simple CNN encoder in regards to the F1-scores, as shown by the highest score of 0.78 for the ResNet50 encoder compared to the 0.63 for the simple CNN encoder. This implies that the complexity of the encoder plays an important role in optimizing improvements. In a similar fashion to the self-supervised approach, the IoU loss appears to perform the best in the supervised context achieving a score of 0.78, although the ResNet50 with frozen encoders trained on BCE also performed strongly with a score of 0.75. Another point of interest in regards to the simple CNN architecture, is that BCE seems to be the only loss function to effectively produce higher scores of 0.63 and 0.71, compared to the 0.27 for the Dice and IoU losses. It also appears to be the case that scores don't differ too strongly when comparing the frozen and non-frozen encoders. This could suggest that the decoder does the majority of the work in semantic segmentation tasks, something that is also in agreement with (Goutam et al., 2020).

Figures 5.6, 5.7, 5.8, and 5.9 illustrate the segmentations performed by both ResNet50 and CNN supervised frameworks across the five folds of the cross validation. Regarding the ResNet50 architecture, it can be observed that when the encoder *is not* frozen, the generated segmentation masks are able to fill in the organoid shape. In contrast, when the encoder *is* frozen, it is only able to segment the edges. As for the simple CNN encoder it is rather clear here that the model is unable to converge in most cases, with the exception of the BCE loss which as discussed earlier, was the only loss function to produce meaningful results. This could be due to the limited amount of data (114 images) that is available for training.

Lastly, when comparing the two frameworks, it can be observed that with a small data set (114 images), the SSL framework performs better than either ResNet50 or simple CNN supervised framework. For instance, the F1-score for the blur augmentation technique (Table 5.2) was between 0.73-0.85, which was generally higher than the supervised framework (Table 5.3) having scores between 0.27-0.78. Additionally, the self-supervised approach is able to consistently perform well regardless of loss function. The supervised approach in contrast has strong inconsistencies on this aspect. Furthermore, when observing the deviations in Tables 5.4, 5.5, and 5.6, a clear disparity can be observed in stability between the SSL and the supervised approach, where the SSL has at most a deviation of 0.047 whilst the supervised has a deviation as high as 0.254.

Table 5.1: Self-Supervised architecture F1-Scores with 25% pixel drop trained on 114 images. The best score of the five folds cross validation is computed for each variation of network structure.

Best		Self-Supervised (0.25 pixel drop)					
	Pre-Train	SSIM			SSIM-L1		
		BCE	Dice	IoU	BCE	Dice	IoU
F1-Score	10%	0.69	0.66	0.71	0.58	0.71	0.70
	50%	0.80	0.81	0.81	0.81	0.82	0.82
	100%	0.84	0.84	0.84	0.85	0.84	0.85

Table 5.2: Self-Supervised architecture F1-Scores with blurring trained on 114 images. The best score of the five folds cross validation is computed for each variation of network structure.

		Self-Supervised (Blurring)					
	Pre-Train	SSIM			SSIM-L1		
		BCE	Dice	IoU	BCE	Dice	IoU
F1-Score	10%	0.74	0.73	0.80	0.82	0.82	0.84
	50%	0.79	0.77	0.82	0.83	0.83	0.84
	100%	0.84	0.84	0.85	0.85	0.84	0.85

Table 5.3: Supervised architecture F1-Scores trained on 114 images. The best score of the five folds cross validation is computed for each variation of network structure.

		Supervised					
	Encoder	Freeze			No Freeze		
		BCE	Dice	IoU	BCE	Dice	IoU
F1-Score	ResNet50	0.75	0.6	0.47	0.57	0.73	0.78
	CNN	0.63	0.27	0.27	0.71	0.27	0.27

Table 5.4: Self-Supervised architecture F1-Scores with 25% pixel drop trained on 114 images. The average score and the standard deviation of the five folds cross validation is computed for each variation of network structure.

Mean		Self-Supervised (0.25 pixel drop)					
	Pre-Train	SSIM			SSIM-L1		
		BCE	Dice	IoU	BCE	Dice	IoU
F1-Score	10%	0.64 ± 0.034	0.64 ± 0.028	0.70 ± 0.005	0.54 ± 0.047	0.70 ± 0.04	0.69 ± 0.02
	50%	0.80 ± 0.005	0.78 ± 0.024	0.80 ± 0.012	0.80 ± 0.012	0.81 ± 0.005	0.82 ± 0.007
	100%	0.84 ± 0.005	0.82 ± 0.012	0.84 ± 0.005	0.85 ± 0.005	0.84 ± 0.004	0.85 ± 0.005

Table 5.5: Self-Supervised architecture F1-Scores with blurring trained on 114 images. The average score and the standard deviation of the five folds cross validation is computed for each variation of network structure.

Mean		Self-Supervised (Blurring)					
	Pre-Train	SSIM			SSIM-L1		
		BCE	Dice	IoU	BCE	Dice	IoU
F1-Score	10%	0.73 ± 0.015	0.70 ± 0.014	0.79 ± 0.014	0.81 ± 0.010	0.81 ± 0.08	0.84 ± 0.005
	50%	0.77 ± 0.01	0.75 ± 0.19	0.81 ± 0.008	0.81 ± 0.008	0.82 ± 0.012	0.84 ± 0.005
	100%	0.83 ± 0.005	0.84 ± 0.005	0.84 ± 0.005	0.85 ± 0.005	0.84 ± 0.005	0.85 ± 0.000

Table 5.6: Supervised architecture F1-Scores trained on 114 images. The average score and the standard deviation of the five folds cross validation is computed for each variation of network structure.

		Supervised					
	Encoder	Freeze			No Freeze		
F1-Score		BCE	Dice	IoU	BCE	Dice	IoU
	ResNet50	0.46 ± 0.232	0.22 ± 0.199	0.32 ± 0.137	0.43 ± 0.118	0.49 ± 0.254	0.644 ± 0.08
	CNN	0.37 ± 0.161	0.054 ± 0.108	0.262 ± 0.016	0.44 ± 0.195	0.11 ± 0.132	0.27 ± 0.000

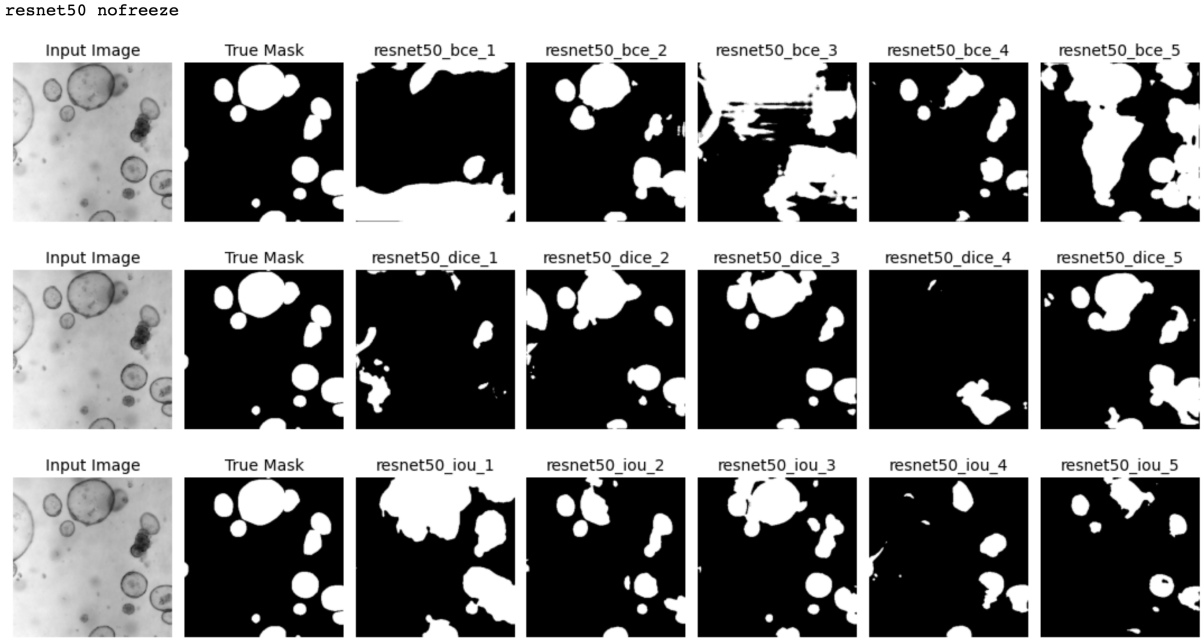

Figure 5.6: Segmentations made with the ResNet50 encoder using supervised approach, where the encoder weights *are not* frozen. On display from the left is: Input (Ground Truth) image, True Mask, then the generated masks of the five folds in ascending order. The top row illustrates the masks using the BCE loss, the middle row using the Dice loss, and the bottom using the IoU loss.

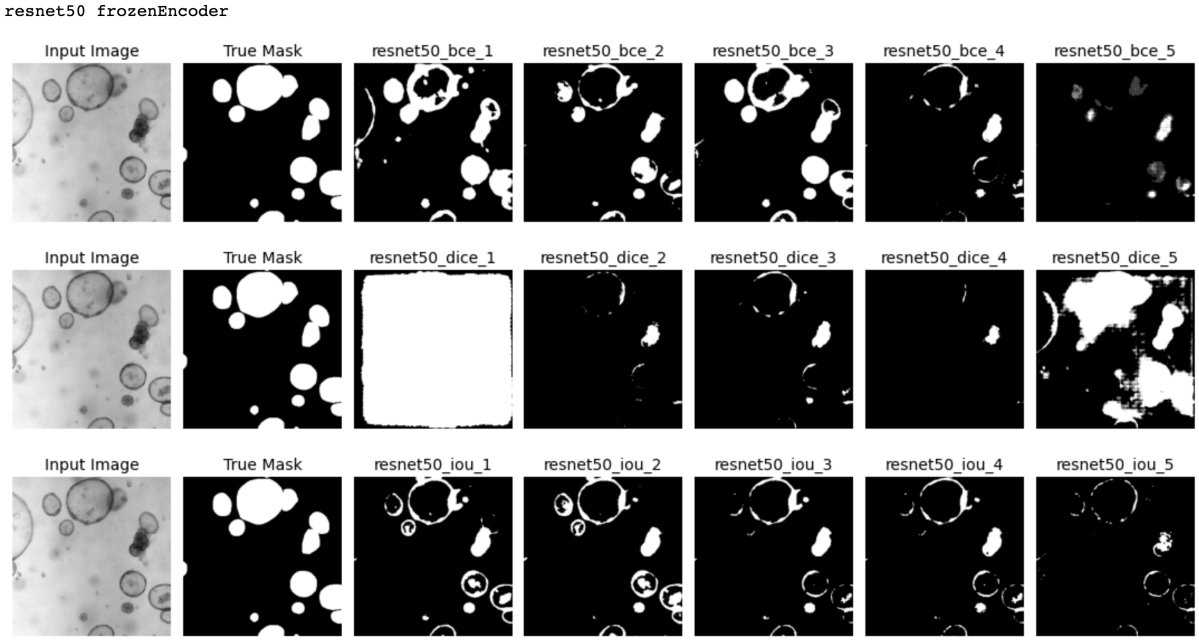

Figure 5.7: Segmentations made with the ResNet50 encoder using supervised approach, where the encoder weights *are* frozen. On display from the left is: Input (Ground Truth) image, True Mask, then the generated masks of the five folds in ascending order. The top row illustrates the masks using the BCE loss, the middle row using the Dice loss, and the bottom using the IoU loss.

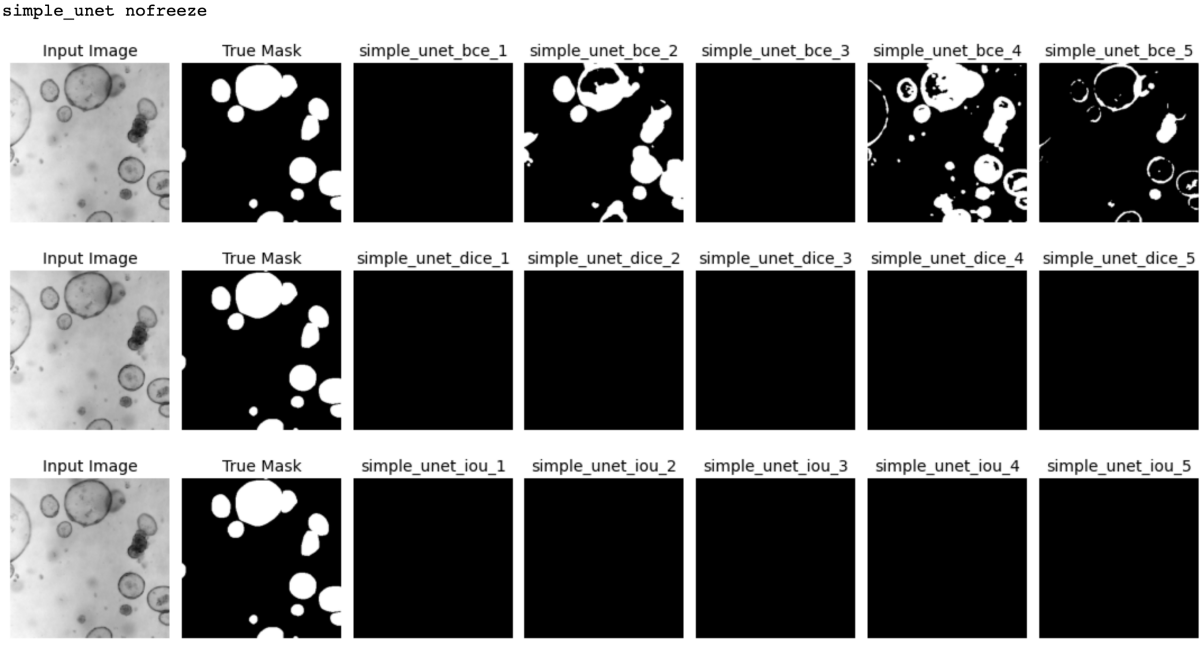

Figure 5.8: Segmentations made with the simple CNN encoder using supervised approach, where the encoder weights *are not* frozen. On display from the left is: Input (Ground Truth) image, True Mask, then the generated masks of the five folds in ascending order. The top row illustrates the masks using the BCE loss, the middle row using the Dice loss, and the bottom using the IoU loss.

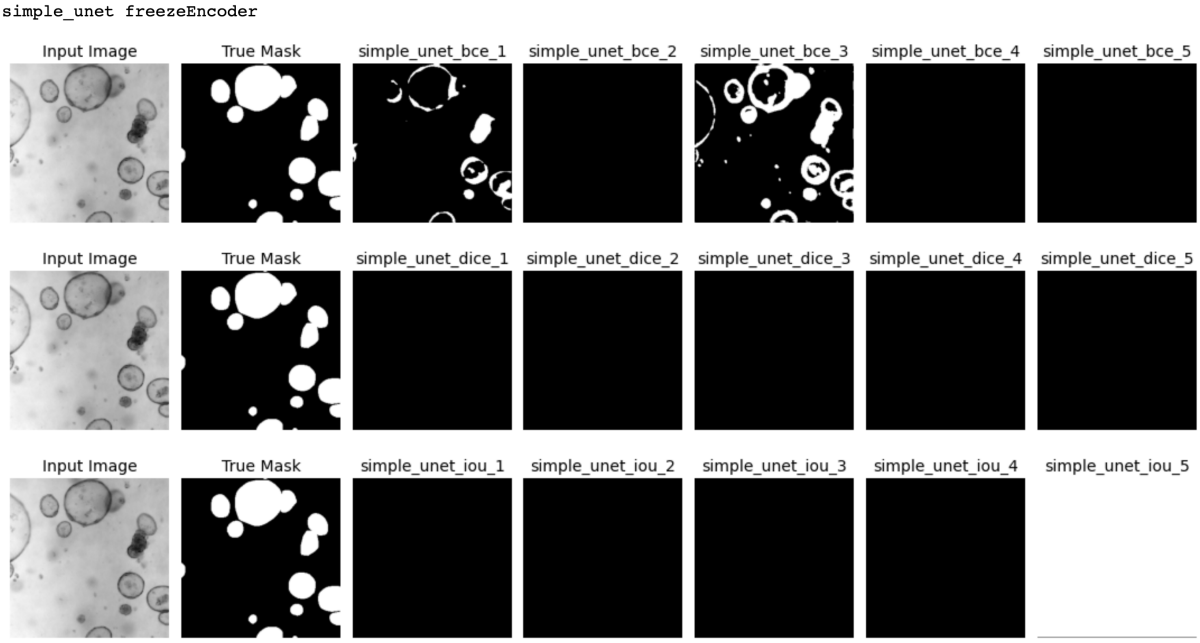

Figure 5.9: Segmentations made with the simple CNN encoder using supervised approach, where the encoder weights *are* frozen. On display from the left is: Input (Ground Truth) image, True Mask, then the generated masks of the five folds in ascending order. The top row illustrates the masks using the BCE loss, the middle row using the Dice loss, and the bottom using the IoU loss.

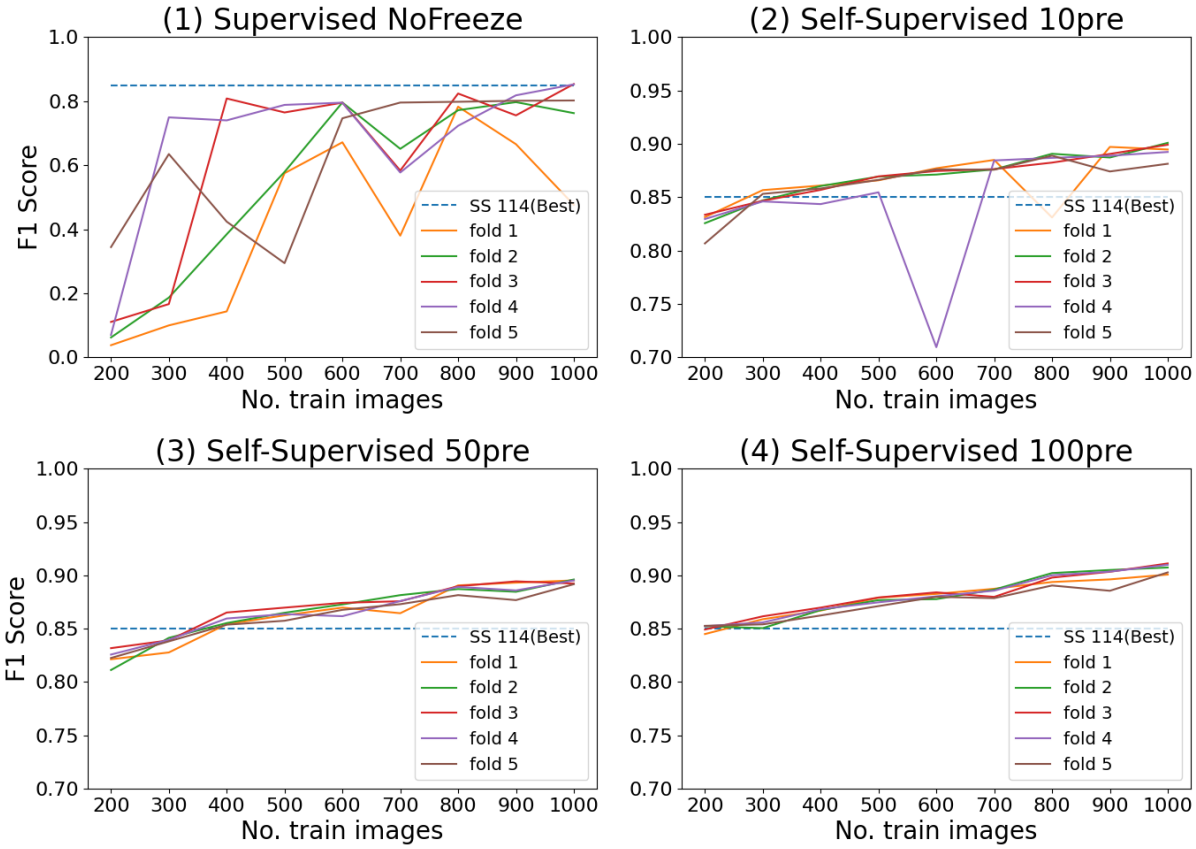

Figure 5.10: F1-scores for (1) supervised trained with the IoU loss using the ResNet50 encoder, and (2)(3)(4) self-supervised trained between 200 up to 1000 images using the SSIM-L1 and IoU loss with 10%, 50%, and 100% of $Train_{pretext}$ data. The blue dotted line indicates the best F1-score of the self-supervised approach trained on 114 images, and is used as a baseline benchmark. Each full line indicates an individual model in the 5 fold cross validation.

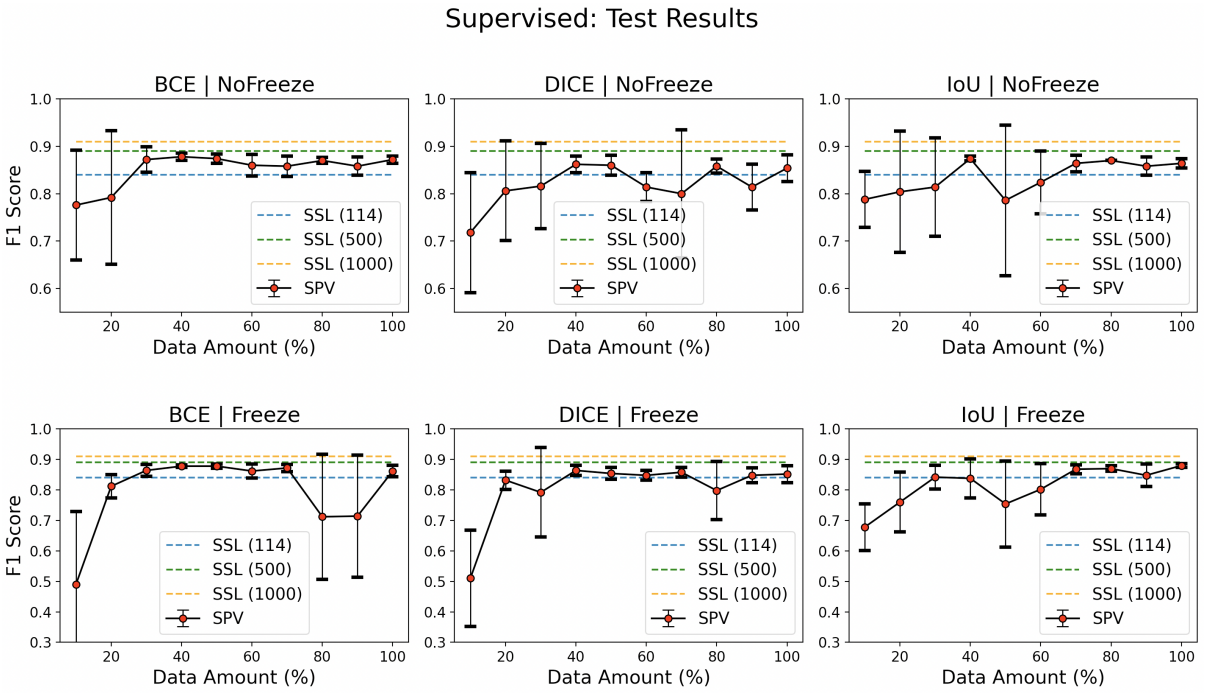

Figure 5.11: Evaluation of supervised (SPV) using F1-Scores trained on various percentage of $Train_{main}$. The mean of the 5 folds was computed and is indicated by the red points, with bars denoting the standard deviation. This is compared to the mean of the self-supervised (SSL) scores shown by the dotted lines. The SSL network was trained with the SSIM-L1 to IoU losses with: 114, 500, and 1.000 images. The top row shows the scores when the encoder is not frozen, whilst the bottom row shows the frozen encoder.

5.3 Comparing Performance with Larger Data Sets

To further observe performance differences between the supervised and the self-supervised approach, the F1-score was measured across networks trained with various numbers of images. Figure 5.10 displays the F1-scores when trained between 200-1.000 images for all five CV folds. What can be observed in graph **(1)**, using the standard supervised approach with a ResNet50 encoder without frozen weights, is that it takes upwards of 1.000 training images to achieve similar results to the best self-supervised approach that was trained on 114 images.

Graphs **(2)**-**(4)**, display the scores for the self-supervised approach pre-trained with 10%, 50%, and 100% of the sub-data respectively (Table 4.1). A major point of interest here is that F1-scores are still improving when trained with more data on the main task, while remaining ahead of the supervised approach. Additionally, when comparing the curves of both approaches, it is clear here that the self-supervised approach has improved stability and reliability over the supervised one. This is a sharp indicator for the effectiveness of the self-supervised approach.

5.4 Supervised Framework

Figure 5.11 displays the F1-scores for the supervised (SPV) framework trained between 10%-100% of $Train_{main}$ which is compared to the self-supervised (SSL) framework trained with: 114, 500, and 1.000 images. As a first observation here, the SPV scores outperform the SSL-114 scores when trained with 30% of $Train_{main}$ or more, but not the SSL-500 and SSl-1000. When comparing the frozen and non-frozen encoders, there appears to be little difference in scores which again, is in agreement with (Goutam et al., 2020) regarding the difference in effectiveness between the encoder and decoder. Lastly, the deviation bars indicate the stability of training across the 5 fold cross validation, where we can see that BCE appears to have the highest stability.

6 Conclusions

6.1 Discussion

This work evaluates the potential of using the SSL paradigm to perform semantic segmentation on images of organoids, specifically by employing the U-Net architecture through image restoration as a pretext task. The use of SSL methods in the context of medical has the potential to greatly improve research in the field by automating processes that otherwise require manual labour from experts.

Regarding the proposed pretext task, a total of 5 augmentations (25%, 50%, 75% pixel drops, Gaussian blurring, and Sobel filtering) paired with 2 loss functions (SSIM and SSIM-L1), trained across 10%, 50% and 100% of data taken from $Train_{pretext}$ were compared. The gained knowledge from the pretext task was then transferred to the main task where the BCE, Dice, and IoU losses were compared afterwards, being trained on 114 images taken from $Train_{main}$. Here, it was discovered that the Guassian blurring augmentation paired with the SSIM-L1 and IoU losses achieved the best results across the range of $Train_{pretext}$ data, with F1-scores of 0.84-0.85, with the 25% pixel drop (with the same losses) coming in close second with a range of 0.70-0.85. Furthermore, it can be concluded that across all metrics in this context, the IoU loss performed the best.

When comparing the SSL and supervised frameworks, it is abundantly clear here that the self-supervised approach is able to achieve better scores than the supervised with smaller labelled data sets. When trained on 114 images from $Train_{main}$, the SSL framework was able to achieve an F1-score of 0.85, whilst the supervised framework scored at most 0.78. It is also important to emphasise the importance of using a complex encoder such as the ResNet50 compared to a CNN, as shown by the comparison between the two supervised frameworks. With larger data sets, a similar conclusion can be drawn that the SSL approach still outperforms the supervised one.

An inherent deficiency in the SSL approach however, is that it will always take longer to train. In the case that labelled data is already available, it

would be more efficient and time effective to take the supervised approach. As mentioned previously however, this in general is not the case with medical imaging. Regardless, it can safely be concluded here that the SSL paradigm can be used effectively to produce robust models when labelled data is not available.

6.2 Future Works

A potential for further investigation regarding the topic of semantic segmentation of organoids could be the use of a different encoder architectures such as the VGG16 (Simonyan and Zisserman, 2014) or the MobileNet (Howard et al., 2017) architectures which have been proven to perform strongly in computer vision tasks. Considering that the encoder plays a significant role in the task of segmentation, it would be beneficial to study the potential of other encoders in this context. Another point of interest could be the use of a different pretext task in the self-supervised method, as this also plays an important role in improving the overall performance. On the topic of pretext task, it would also be interesting to observe the performance of the image reconstruction task on other data sets such as the COCO (Lin et al., 2014) set.

References

Bishwo Adhikari, Esa Rahtu, and Heikki Huttunen. Sample selection for efficient image annotation. In *2021 9th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6. IEEE, 2021.

Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3478–3488, 2021.

Jeroen Bertels, Tom Eelbode, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B Blaschko. Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In *International conference on medical image computing and computer-assisted intervention*, pages 92–100. Springer, 2019.

Michael A Borten, Sameer S Bajikar, Nobuo Sasaki, Hans Clevers, and Kevin A Janes. Automated brightfield morphometry of 3d organoid populations by organoseg. *Scientific reports*, 8(1):1–10, 2018.

Liang Chen, Paul Bentley, Kensaku Mori, Kazunari Misawa, Michitaka Fujiwara, and Daniel Rueckert. Self-supervised learning for medical image analysis using image context restoration. *Medical image analysis*, 58:101539, 2019.

Claudia Corrò, Laura Novellasdemunt, and Vivian SW Li. A brief history of organoids. *American Journal of Physiology-Cell Physiology*, 319 (1):C151–C165, 2020.

Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia*, pages 21–49. Springer, 2008.

Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092, 2020.

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

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

Jarno Drost and Hans Clevers. Organoids in cancer research. *Nature Reviews Cancer*, 18(7):407–418, 2018.

Tom Eelbode, Jeroen Bertels, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B Blaschko. Optimization for medical image segmentation: theory and practice when evaluating with dice score or jaccard index. *IEEE Transactions on Medical Imaging*, 39(11): 3679–3690, 2020.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Kelam Goutam, S Balasubramanian, Darshan Gera, and R Raghunatha Sarma. Layerout: Freezing layers in deep neural networks. *SN Computer Science*, 1(5):1–9, 2020.

O Graudejus, RD Ponce Wong, N Varghese, S Wagner, and B Morrison. Bridging the gap between in vivo and in vitro research: Reproducing in vitro the mechanical and electrical environment of cells in vivo. In *MEA Meeting 2018, 11th International Meeting on Substrate Integrated Microelectrode Arrays*, 2018.

Asmaa Haja and Lambert RB Schomaker. A fully automated end-to-end process for fluorescence microscopy images of yeast cells: From segmentation to detection and classification. In *International Conference on Medical Imaging and Computer-Aided Diagnosis*, pages 37–46. Springer, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Aleksandra Karolak, Sharan Poonja, and Katarzyna A Rejniak. Morphophenotypic classification of tumor organoids as an indicator of drug exposure and penetration potential. *PLoS Computational Biology*, 15(7):e1007214, 2019.

Michael J Kratochvil, Alexis J Seymour, Thomas L Li, Sergiu P Paşca, Calvin J Kuo, and Sarah C Heilshorn. Engineered materials for organoid systems. *Nature Reviews Materials*, 4(9):606–622, 2019.

Kai Kretzschmar. Cancer research using organoid technology. *Journal of Molecular Medicine*, 99 (4):501–515, 2021.

Wenqi Li, Guotai Wang, Lucas Fidon, Sebastien Ourselin, M Jorge Cardoso, and Tom Vercauteren. On the compactness, efficiency, and representation of 3d convolutional networks: brain parcellation as a pretext task. In *International conference on information processing in medical imaging*, pages 348–360. Springer, 2017.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*, 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Liangliang Liu, Jianhong Cheng, Quan Quan, Fang-Xiang Wu, Yu-Ping Wang, and Jianxin Wang. A survey on u-shaped networks in medical image segmentations. *Neurocomputing*, 409: 244–258, 2020.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Amanda Marchini and Fabrizio Gelain. Synthetic scaffolds for 3d cell cultures and organoids: applications in regenerative medicine. *Critical reviews in biotechnology*, 42(3):468–486, 2022.

Jonathan M Matthews, Brooke Schuster, Sara Saheb Kashaf, Ping Liu, Mustafa Bilgic, Andrey Rzhetsky, and Savas Tay. Organoid: a versatile

deep learning platform for organoid image analysis. *bioRxiv*, 2022.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

Reid T Powell, Micheline J Moussalli, Lei Guo, Goeun Bae, Pankaj Singh, Clifford Stephan, Imad Shureiqi, and Peter J Davies. deeporganoid: A brightfield cell viability model for screening matrix-embedded organoids. *SLAS Discovery*, 27(3):175–184, 2022.

KKD Ramesh, G Kiran Kumar, K Swapna, Debabrata Datta, and S Suman Rajest. A review of medical image segmentation algorithms. *EAI Endorsed Transactions on Pervasive Health and Technology*, 7(27):e6–e6, 2021.

Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps*, pages 323–350, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Giuliana Rossi, Andrea Manfrin, and Matthias P Lutolf. Progress and potential in organoid research. *Nature Reviews Genetics*, 19(11):671–687, 2018.

Pratap Chandra Sen, Mahimarnab Hajra, and Mitadru Ghosh. Supervised classification algorithms in machine learning: A survey and review. In *Emerging technology in modelling and graphics*, pages 99–111. Springer, 2020.

Samriti Sharma, Gurvinder Singh, and Manik Sharma. A comprehensive review and analysis of supervised-learning and soft computing techniques for stress diagnosis in humans. *Computers in Biology and Medicine*, 134:104450, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

David Tuveson and Hans Clevers. Cancer modeling meets human organoid technology. *Science*, 364 (6444):952–955, 2019.

Quoc Dang Vu, Simon Graham, Tahsin Kurc, Minh Nguyen Nhat To, Muhammad Shaban, Talha Qaiser, Navid Alemi Koohbanani, Syed Ali Khurram, Jayashree Kalpathy-Cramer, Tianhao Zhao, et al. Methods for segmentation and classification of digital microscopy tissue images. *Frontiers in bioengineering and biotechnology*, page 53, 2019.

Fei Wang, Lawrence Peter Casalino, and Dhruv Khullar. Deep learning in medicine—promise, progress, and challenges. *JAMA internal medicine*, 179(3):293–294, 2019.

Huan Wang, Zhiliang Liu, Yipei Ge, and Dandan Peng. Self-supervised signal representation learning for machinery fault diagnosis under limited annotation data. *Knowledge-Based Systems*, 239: 107978, 2022.

Yijia Wang and Amanda B Hummon. Ms imaging of multicellular tumor spheroids and organoids as an emerging tool for personalized medicine and drug discovery. *Journal of Biological Chemistry*, 297(4), 2021.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13 (4):600–612, 2004.

Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.

Shuang Yu, Di Xiao, and Yogesan Kanagasingam. Automatic detection of neovascularization on optic disk region with feature extraction and support vector machine. In *2016 38th Annual International Conference of the IEEE Engineering*

in Medicine and Biology Society (EMBC), pages 1324–1327. IEEE, 2016.

Pingyue Zhang, Mengyue Wu, Heinrich Dinkel, and Kai Yu. Depa: Self-supervised audio embedding for depression detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 135–143, 2021.

Zheng Zhang, Qi Zhu, Guo-Sen Xie, Yi Chen, Zhengming Li, and Shuihua Wang. Discriminative margin-sensitive autoencoder for collective multi-view disease analysis. *Neural Networks*, 123:94–107, 2020.

Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5): 749–753, 2018.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2015.

Chuang Zhu, Wenkai Chen, Ting Peng, Ying Wang, and Mulan Jin. Hard sample aware noise robust learning for histopathology image classification. *IEEE Transactions on Medical Imaging*, 41(4): 881–894, 2021.