The interest in automatically analyzing biomedical images increased in the past years, as an accurate localization and segmentation of organoids can help with the early detection of malignancies and predict diseases, such as cancer. The morphometric appearances of these images and the high level of overlapping in the organoids make the segmentation task challenging. This chapter studies a simple U-Net and also proposes a double U-Net model with a shared encoder and two decoders, one for binary segmentation of the mask and one for the multi-class segmentation of overlaps. A significant addition to the U-Net is the residual-atrous skip connections, which reduce the semantic gap between the encoder and the decoder. The issue of high imbalance between the classes is addressed using a combination between Focal Loss and Focal Tversky Loss, which significantly improved the performance of the model. Ten networks were trained on more than 20,000 crop images with overlapping and non-overlapping organoids and obtained promising results. When tested on 88 new images, the final models achieved an F1 score of 0.83 for the mask channel and 0.43 for the overlapping channel. The Jaccard Index was 0.72 for the mask and 0.34 for the overlap. To the authors’ knowledge, there exists no work in the literature that explores the detection and segmentation of overlapping objects in biomedical images, especially segmenting overlapping organoids, which also shows the novelty of this work.

This chapter has been previously published as:
5.1 INTRODUCTION

Organoids are 3D tissue structures derived from adult or embryonic stem cells that can replicate the micro-anatomy of any organ [134]. Unlike traditional 2D cell lines, organoids contain several types of cells and facilitate the study of tissue physiology or the development of diseases. In the study of cancer, for instance, cells are taken directly from the tumor and used to create organoids that will be an in-vitro equivalent of the in-vivo tissue [135]. They are used to study the development of infectious diseases, such as the one caused by the Zika virus [136] or genetic disorders, especially in organs with no regenerative capacity such as the brain [137]. As they are harvested directly from the patient’s tissue, organoids make personalized and more effective treatments easier to obtain. In order to reach the treatment phase as early as possible, automatic techniques are used to accelerate the analysis process.

Deep learning (DL) is one of the most popular methods for computer-aided diagnosis and examination of biomedical data, such as cells or organoids. Due to the heterogeneous morphometric appearance and the high level of overlapping [134], the study of organoids is challenging. As a consequence, many DL models are focusing on hematologic images consisting of cells on a background, which differs from the bright field images of organoids. They are successfully used to diagnose diseases, such as acute leukemia [138], cervical cancer [139] or brain tumors [140]. A popular architecture is the encoder-decoder U-Net model [141], which is generally applied to biomedical data and serves as a starting point for this chapter. However, the U-Net focuses mostly on the binary discrimination task and very few models had been applied to organoid datasets.

Different and enhanced versions of the U-Net can be used to solve the problem of multi-class semantic segmentation in overlapping organoids. Semantic segmentation differentiates between the classes in an image by assigning a class label to each pixel. Binary segmentation is used for the mask channel and it is significantly more facile than multi-class segmentation, where the model has to choose between multiple candidate classes, as it is the case with overlaps. When these organoids overlap, some are partially or completely obstructed by others, which might result in them not being analysed and relevant information being lost along the way. Moreover, clusters of cells can form masses, which are an indicator of tumors, as in the case of breast cancer [142]. This is why the detection of the overlaps is a difficult problem, especially when there is a high imbalance between classes: background, object, overlap.

Class imbalance had been shown to have a detrimental effect in many real-life classification tasks [143] and the current dataset encounters the same issue.
Class imbalance involves having one class in the training set with much more examples than the rest of the classes. Several methods are used to address this problem, for example, under-sampling or over-sampling, which remove or add data, such that a balance is achieved. Other techniques that operate at the model level include the introduction of weights based on the frequency of classes [144]. Loss functions with factors that modulate the number of false positives (FPs) or false negatives (FNs) [145]–[147] represent an effective solution, which does not change the inner distribution of the data.

The current study attempts to solve the problem of overlapping organoids in a highly imbalanced dataset. To accomplish this, multiple additions had been brought to the classical U-Net model, including a second decoder for the multi-class segmentation, residual-atrous skip connections, and a learning rate scheduler. Moreover, to address the class imbalance issue, the models use a combination of Focal loss [145] and Focal Tversky loss [146], which down-weight the most frequent class.

This chapter is organized into 5 sections and the structure is as follows: section 5.2 presents a literature review; section 5.3 contains information about the data, the structure of the proposed models and a description of the different experimental designs; section 5.4 discusses the results of the experiments, and section 5.5 concludes this study and indicates the future works.

5.2 RELATED WORKS

Semantic segmentation has a wide range of applications. Several datasets include images of common everyday objects [148] or traffic scenarios [149] used to train autonomous driving agents. In essence, semantic segmentation is a classification task, where each pixel is assigned a class label. Binary segmentation is the most elementary case, as it only involves 2 classes, usually the background and the foreground.

To differentiate between the 2 classes, early methods use edge detection matrices that perform convolution over an input image to detect boundaries. Some examples are the Sobel filter or the Laplace kernel, which detects both horizontal and vertical edges [150]. Thresholding segmentation methods like the Otsu algorithm [151] can classify the background of an image using a global threshold. In region-based segmentation, the goal is to detect the immediate boundaries of pixels, and one popular approach is watershed [152], where the image is decomposed into catchment basins that are enclosed by watersheds, that act as boundaries. These methods can not deal with complex multi-class segmentation.
DL models started to gain more popularity due to their better performance in the detection of multiple classes [153]. The power of convolutional neural networks, or short CNNs, lies in the convolutional layers that extract features, the non-linear layers that apply an activation function and the pooling layers that reduce spatial resolution [154]. One architecture for object recognition that achieved impressive results on large datasets is the VGGNet [155]. In [138], LeuFeatx – was used as a variation on the VGG16 – to differentiate between multiple leukemia sub-types. However, in order to make the input reconstruction task possible, the encoding path of the VGGNet is insufficient. This led to a new category of DL models, those with an encoder-decoder architecture. The segmentation task involves pixel-wise classification and to achieve this, the information in the input needs not only to be harvested but also rearranged in such a way that the final output displays relevant knowledge. Networks such as the SegNet [156] use the encoding features of the VGG16 but add a decoder with max-pooling layers that perform non-linear up-sampling. Therefore, these networks become more powerful and can currently address the object overlapping problem in more specific scenarios such as microscopic data.

In the field of biomedical image segmentation, many of the segmentation models are inspired by the U-Net, proposed by [141]. It works by using a constricting path to extract features, followed by a symmetrical decoding path that places the features into context and recreates the image. The classical U-Net was applied on neuronal structures from microscopic stacks, as well as light microscopic images, and solved the segmentation task successfully with Intersection Over Union (IoU) scores of 0.93 and 0.77, respectively. Several variations on the U-Net were designed, including the DenseRes-Unet [157], which segments clustered nuclei from histopathology images and uses residual skip connections instead of traditional ones. [158] focused on cell counting and compared two Fully Convolutional Regression Networks based on the encoder-decoder architecture. The study demonstrates that results achieved on synthetic data generalize properly on real microscopy images. [159] uses CNNs and regression to address the problem of touching nuclei in stained images. In all these cases, one recurrent problem is the class imbalance, in addition to the multi-class classification problem.

In semantic segmentation, a class imbalance is more difficult than in regular classification because balancing would require some changes made to the input image. Therefore, a counteracting imbalance needs to be realized within the neural network architecture and training methods. Thus, in order to reduce the frequency of one or more of the classes, studies use different approaches. One variation of the U-Net uses a shared encoder and two decoders [160] to detect clustered nuclei in glioma images. The first decoder is used for boundary
segmentation, the second for the distance map prediction of the interior, and a third forward convolution network is trained as a fusion layer. [161] uses stacked U-Nets and a hybrid loss function to address the class imbalance in a multi-output road extraction task. As a substitute for changing the structure of the U-Net, the focus is on the loss functions. There are a number of loss functions that can successfully minimise the disproportion between classes.

Focal Loss [145] is a popular function based on Cross Entropy that tries to reduce the class imbalance. [162] uses it together with the IoU loss to solve a segmentation task involving 80 different classes. [163] segments kidney organoids using Focal loss and [164] uses it as part of a model that counts cancer cells from zebrafish organoids. Other loss functions which look at the entire image include the Focal Tversky loss [146]. While this loss has not yet been applied in organoids datasets, it obtained a decent performance in the segmentation of brain tumors [165] or biomarkers in cases of bladder cancer [166].

The current study adopts the idea of double U-Nets to solve the overlapping problem in organoids. It presents a comparison between two network architectures with two outputs: the binary mask channel and a multi-class overlapping channel. As the class imbalance was a significant issue in the dataset, different combinations of loss functions are used and their results are compared.

5.3 METHODS

5.3.1 Data

The data was collected by the University Medical Centre Groningen (UMCG), the Netherlands, to study different metabolic diseases. Digital microscopes are used to take high-resolution CZI images of three different overlapping organoids cultures. These cultures contain organoids that are derived from the same group of cells. Each 3D culture consists of 14 stacks, and every stack represents a horizontal slice in the culture. One example of such a stack is given in Figure 5.1. As the majority of the organoids, especially overlapping ones, are intelligible in the middle stacks, stacks numbers 6, 7 and 8 were selected to generate the training dataset. Two other stacks from the remaining cultures were set aside and used for testing.

The true mask channel is generated using the OrganelX system [108], which detects, localizes and segments organoids given an input CZI file. The OrganelX system specifies the boundary of each detected organoid object. As parts of automatic segmentation were not accurate, limited manual work was performed
to fix some of the boundaries coordinates. However, this is a tedious process and due to the blurriness and the high level of overlapping, a significant number of organoids, especially large ones, were missed by the Mask-RCNN [167] used in the system. This represents one of the main issues encountered in the training process.

To create the true overlapping channel, the number of segmented organoid objects for each pixel were counted. This can be realized because the OrganelX system provides the boundaries of each detected object. All overlaps are treated as the same class, regardless of the number of organoids that are overlapping. As a result, three classes were defined: 0 for background, 1 for organoids, and 2 for overlaps. The limitations of the Mask-RCNN resulted in many missed overlaps.
All the previously discussed issues, together with the nature of the data created a serious imbalance in the frequency of classes. Table 5.1 shows the distribution of the different classes in the train and test sets. In the training dataset, overlaps account, on average, for only 1% in an image, while the frequency of the background class is 65% higher. The test set consists of images where the sum of mask and overlap pixels account for at least half of the total. Consequently, there are 8 times more overlaps than in the train set and the mask is also more recurrent, as it accounts for almost 50%, on average. The class imbalance is a common issue in segmentation tasks and solutions including down-sampling the data to match the lowest frequency class can lead to a lack of diversity in the dataset [168]. This is why, the current study proposes a combination of different losses, which includes class weights.

The initial culture images have a high resolution, with a size of 3830 by 2900 pixels. As this is incredibly large for a DL network, a sliding window with a step size of 60 pixels is used over the image to create smaller crops of 320 by 320 pixels. The edges of the image are ignored, as they do not contain any organoids. During training, the images are shrunk further with a scaling factor of 0.5. The reason for this is that images with a lower resolution reduce the training time considerably. After preprocessing and before augmentation, the total number of crops is 2,722.

To introduce diversity in the dataset, image augmentation techniques are used to generate additional examples. The first augmentation method used is rotation. The ground truth images are rotated with 90°, 180° and 270°. The second method is affine transformations, including a combination of positive and negative shearing with factors of −0.6 and 0.6, as well as translation on the x and y axes with a value of 1000. An affine transformation preserves parallel lines and creates images that continue to look realistic.

After the data augmentation, the train set contains 21,776 grey-scale crops. One random crop is shown in Figure 5.2. The leftmost crop is the ground truth image, followed by the true mask channel and the true overlapping channel. The mask channel is binary, so it contains two classes, the background, encoded as 0 and the organoids, encoded as 1. In the overlapping channel, there are three
classes. Apart from the background and the foreground, any kind of overlap in the organoids is encoded using value 2.

Figure 5.2: A random crop consisting of the ground truth image, the true mask and the true overlap with the corresponding encoding values: 0 - background, 1 - organoid, 2 - overlap.

5.3.2 Model Architectures

The U-Net model proposed by [141] serves as a starting point for the architectures used in this chapter. The current study focuses on the overlapping channel of the segmentation process, which is a significantly more intricate task, due to the low frequency of examples. The encoder-decoder architecture is preserved, but two main additions are included: the second decoder in the case of the double network and the residual-atrous skip connections. Depictions of the two architectures -- the simple network with one decoder and the double network with two decoders -- can be seen in Figures 5.3 and 5.4, respectively.

The encoder follows the structure of a convolutional neural network. It contains four down-convolution blocks, where each block is a series of two $3 \times 3$ padded convolution layers, each followed by a 2D normalization layer and a ReLU activation function. The role of the convolution layers is to increase the number of channels in the image. The input is a grey-scale $320 \times 320$ pixels image ($160 \times 160$ after scaling) with only one channel and the number of channels doubles with each block of the encoder. The normalization layer stabilizes the gradients in the learning process and the activation function reduces the exponential growth of the weights. At the end of the down block, a $2 \times 2$ max-pooling layer is applied, which reduces the size of the image. The role of the encoder is to squeeze the relevant information into a bottleneck and create a representation of the input. The input image that had one channel originally is encoded to a feature vector of 1024 values at end of the encoder. Figure 5.5 shows how the contracting path encodes information, starting with low-level features at the beginning of the
5.3 Methods

Figure 5.3: The simple U-net architecture with one encoder and one decoder. The two outputs — the mask channel and the overlapping channel — are generated in the end.

Figure 5.4: The double U-Net architecture with a shared encoder and two decoders, each responsible for a segmentation — mask or overlapping — channel.
encoder and finishing with higher-level features.

The decoder is an expanding path that places the extracted features into context and groups the pixels under classes. In the case of the simple U-Net architecture, as the input travels through the decoder, the image is up-sampled using $2 \times 2$ up-sampling layers, a series of double convolution layers and ReLU activation functions, in a symmetrical manner to the encoder. The number of feature maps is decreased in the final step, and a $1 \times 1$ convolution is applied to map the features to the final classes. The output of the network constitutes of pairs of images, the binary mask channel and the three-class overlapping channel. In the case of the double U-Net, it has one encoder, but two decoders, one responsible for each channel. As the features of both channels share similarities — the mask part in the two outputs is almost identical — it is reasonable to believe that the branches can be encoded together. While the simple architecture assumes that they can also be decoded together, the double version separates the ways in which the channels are reconstructed. The motivation for this is that the overlaps are areas that are very alike to the organoids themselves, and by separating the features, the aim is to reduce the number of overlaps falsely labelled as organoids. Figure 5.4 shows the two decoders, which are equivalent in terms of structure. The only difference is in the final number of classes of the output: two for the binary mask and three for the overlapping mask.

The role of skip connections is to take information from each of the levels of the encoder and copy them directly to the decoder by skipping the bottleneck. This way, the loss of features is reduced. While the conventional U-Net concatenated the feature maps directly, [157] suggests that this does not account for the semantic gap between the encoder and the decoder. By following the authors’ advice, the current model integrates residual-atrous blocks, which are applied to the feature maps of the encoder. Accordingly, $3 \times 3$ and $1 \times 1$ convolution layers are applied to the features of the encoder to reduce information loss. The weights are summed together element-wise and an atrous block with two parallel dilated convolution layers is then applied. The convolution layers are $3 \times 3$ and use dilation rates of $2$ and $4$. The dilation rate ensures that more spatial information is encoded and that the blurriness of the image is minimised [157]. The resulting
5.3 Methods

features are finally concatenated to the corresponding level of the decoder. In the case of the double architecture, the same feature maps are copied in the two decoders.

5.3.3 Experimental Design

This study investigates four different additions to the traditional U-Net and compared a total of 10 models. These four contributions are the residual-atrous skip connections, the architecture of the model, the combination of loss functions and the scheduler.

The 21,776 input images are divided into a training and a validation set. 90% of the data is used for training and 10% for validation. The F1 score was used as an evaluation metric for the validation set (see section 5.4.1). The batch size is set to 1 indicating that only one random crop is presented to the network at a time, followed by an update of weights based on the loss. A wide variety of losses, such as Cross Entropy, Focal Loss, Dice Loss, or Focal Tversky Loss are used throughout the experiments. The optimizer used during training was a first-order optimization algorithm – RMSprop with a momentum factor of 0.9. After computing the gradients based on the loss, the optimizer updates the weights. RMSprop is a technique for mini-batch learning that deals with the vanishing or explosion of gradients by normalizing the gradients using a squared moving average. The goal of the normalization step is to balance the momentum, by decreasing the step size for large gradients and increasing the step for small gradients. The majority of the models are trained for 2 epochs, but in order to assess the performance of a longer training time, two models are trained for 3 epochs. The starting learning rate was 1 x 10^-5 for all the models, but in some, a scheduler was used, while in others the decreasing step was performed manually. All models are tested on 88 unseen images from two different organoids cultures. Additionally, transfer learning is used to initialize the weights of the model with pre-trained weights from a U-Net trained on 11 grayscale images of the Carvana dataset.

5.3.3.1 Residual - Atrous Skip Connections

In this first experiment, the aim is to compare the performance of the model – especially the F1 score – when it uses traditional skip connections versus enhanced residual-atrous skip connections [157]. Two comparisons are presented, involving 4 models that were trained for 2 epochs, using the ReduceOnPlateau learning rate scheduler. The first comparison looks at the effect of the residual-atrous skip connections on a simple architecture. In this case, the weights are

1 The data can be found at https://www.kaggle.com/c/carvana-image-masking-challenge
updated based on a combination of the Cross-Entropy Loss and the Dice Loss. Later on, the same experiment is performed on 2 double U-Nets, which use the Focal Loss and the Focal Tversky Loss. These loss combinations are selected based on multiple experiments with different model configurations. As the results during training (section 5.4.2.1) indicate that applying the residual-atrous block in the skip connection improves the validation F1 score and decreases the loss in both scenarios, this addition was kept in the future experiments.

5.3.3.2 Loss Functions

Another set of experiments is carried out to test the effect of the loss function. Three different combinations of losses are used on a simple U-Net model with residual-atrous skip connections. The first combination is the classical Cross Entropy + Dice Loss (CE + DL), which is very popular in semantic segmentation tasks, followed by Focal Loss + Dice Loss (FL + DL) and Focal Loss + Focal Tversky Loss (FL + FTL). Several studies [138], [156] use Cross Entropy or Categorical Cross Entropy as a low-level loss function. Cross Entropy is a binary loss function that measures the performance of a classification model, by computing the pixel-wise difference, as shown in Equation 5.1, where \( p \in [0, 1] \) is the probability for the true class. Similarly, Categorical Cross Entropy is used when more than two classes are involved, as this is the case in the overlapping channel. However, for very imbalanced datasets, the loss tends to be dominated by the most frequent class. This is why a more viable solution to solve the overlapping problem is Focal Loss [145]. This function uses the extra \(-\alpha(1 - p)^\gamma\) balanced modulating factor (Equation 5.2), which reduces the loss contribution of frequent examples so that the network can focus on the more difficult classes. The parameter \( \gamma \) adjusts the rate at which loss is reduced for well-classified examples and the factor \( \alpha \) balances the importance of positive and negative examples. In the current experiments, \( \alpha \) is set to 1 and the function is initialized with the weights of the training class (Table 5.1) to achieve the desired balancing effect. In this work, the \( \gamma \) factor is set to 2.

\[
CE(p) = -\log(p) \quad (5.1)
\]

\[
FL(p) = -\alpha(1 - p)^\gamma \log(p) \quad (5.2)
\]

The Dice Loss (Equation 5.3) is widely used in biomedical segmentation tasks as a second loss function. It is defined as 1 – DS, where the F1 score (DS) indicates the amount of overlap between the true (mask/overlap) image and the predicted image and is equivalent to the F1 score. A perfect overlap indicates a DS equal to 1 and so a loss that equals 0. This loss, however, does not address the class imbalance problem. To address this issue, the Focal Tversky Loss [146] is used instead of the Dice loss. It is a generalization of the Dice loss but uses two extra
hyper-parameters which modulate the number of false positives (FP) and false negatives (FN) (Equation 5.4). In order to penalize the false negatives for the less frequent classes (mask and overlap), the $\beta$ factor is set to 0.7 and $\alpha$ to 0.3. The terms, TP and TN indicate the true positive and true negative, respectively.

$$DL = 1 - \frac{2 \times TP}{TP + FP + TP + FN}$$

(5.3)

$$FTL = 1 - \frac{TP}{TP + \alpha FP + \beta FN}$$

(5.4)

The experiments conducted in this subsection study the effect of loss functions on the simple U-Net. Three models are trained in total.

### 5.3.3.3 Model Architecture

The third set of experiments analyses the effect of model architectures. All models were trained for 2 epochs, a double U-Net with a shared encoder and two decoders (Figure 5.4) and a simple U-Net with a single encoder and decoder (Figure 5.3). By using only one decoder, the simple architecture decodes the features concurrently for the mask and overlapping channels and builds the outputs in parallel. On the other hand, the double architecture involves two branches, one responsible for each channel. Both model architectures are compared using the three combinations of loss functions, introduced in the previous section.

### 5.3.3.4 Learning Rate and Scheduler

The last experiment looks at the effect of the scheduler. While the ReduceOnPlateau scheduler is used in all previous setups, the goal of this current analysis is to show that it does indeed improve the evaluation $F_1$ score, compared to a manual approach. The role of the learning rate is to modulate the step size when moving towards the minimum of the loss function. In order to achieve a balance between exploration and exploitation, the initial learning rate starts high but decreases with time. The starting learning rate used throughout this study is $10^{-5}$. Two methods to adjust the learning rate are compared, one is by manually decreasing its value after each epoch with a factor of $10^{-1}$ and an automatic one using a scheduler that decreases the learning rate with the same $10^{-1}$ factor when the evaluation score stops improving. In other words, when the $F_1$ score function stops increasing monotonically, the learning rate is reduced. No plateau movements are allowed. Moreover, to allow the model to further explore the state space at the beginning of the training, the scheduler is frozen for the first two epochs. To preserve the equivalence, the manual decrease was also performed only at the beginning of the second epoch. The models were trained for a total of 3 epochs.
5.4 RESULTS

5.4.1 Evaluation Metrics

Several metrics are used to evaluate the performance of the model, both at the pixel and image levels. All metrics are computed per each class, not including the background and then averaged. The most trivial metric which looks at the individual pixels is the accuracy, defined in Equation 5.5, followed by precision (5.6) and recall (5.7). A perfect precision score is an indicator that there are no false positives, so no background pixels miss-classified as mask, for example. On the other hand, a perfect recall score shows that there are no false negatives (FNs), so no missing organoids. Due to the imperfect nature of the true masks, the focus is on the minimisation of FNs.

\[ \text{ACC} = \frac{1}{\text{cls}-1} \sum_{i=1}^{\text{cls}} \frac{TP + TN}{TP + TN + FP + FN} \]  

(5.5)

\[ \text{Precision} = \frac{1}{\text{cls}-1} \sum_{i=1}^{\text{cls}} \frac{TP}{TP + FP} \]  

(5.6)

\[ \text{Recall} = \frac{1}{\text{cls}-1} \sum_{i=1}^{\text{cls}} \frac{TP}{TP + FN} \]  

(5.7)

The F1 score is a commonly used metric in classification tasks and it is applied during the evaluation step in training, as well as in testing. The average F1 score is computed based on the number of classes \( \text{cls} \) (Equation 5.8). It is important to note that class 0 (the background) is purposely left out of the computation due to its high frequency, which would have altered the results. Therefore, in the mask channel, the score is computed only between the mask pixels in the true image and the mask pixels in the predicted image. The same applies to the overlapping channel, but in this case, the average is computed between the mask and the overlap parts.

\[ F1 = \frac{1}{\text{cls}-1} \sum_{i=1}^{\text{cls}} 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]  

(5.8)

The Jaccard Index (JI) is also used as a metric (Equation 5.9), equivalently to the F1 score. While the JI measures image similarity comparable to the F1 score, when averaged on multiple examples, the values differ and generally, the JI scores are lower than the corresponding F1 score, although they remain positively correlated.

\[ JI = \frac{1}{\text{cls}-1} \sum_{i=1}^{\text{cls}} \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \]  

(5.9)
\[ JJ = \frac{1}{\text{cls} - 1} \sum_{i=1}^{\text{cls}} \frac{TP}{TP + FN + FP} \]  

(5.9)

5.4.2 Experimental Results

This Section presents the results of all sets of experiments discussed in Section 5.3, both during training and testing. Table 5.2 provides an overview of the 10 final models. The name of every model follows the pattern of Model_Loss_Connection_Addition, where the first term indicates the model type, while the second and the third terms indicate the loss function and the type of skip connection. The last term provides additional information if necessary. The first subsection discusses the residual-atrous skip connections. Section 5.4.2.2 presents a comparison of different combinations of losses, followed by an experiment involving the two architectures (Section 5.4.2.3) and finally a description of the scheduler in section 5.4.2.4.

Table 5.2: An overview of the 10 models with respect to the skip connections, model architecture type, loss functions, scheduler and the number of training epochs. The res-atrous entry indicates the use of residual-atrous skip connections, as opposed to regular connections. The model architecture has two types: simple U-Net or double U-Net. The loss function includes a combination of the following: Cross Entropy (CE), Focal loss (FL), Dice loss (DL) and Focal Tversky loss (FTL). The name of every model follows the pattern of Model_Loss_Connection_Addition, where the first term indicates the model type, while the second and the third terms indicate the loss function and the type of skip connection. The last term provides additional information if necessary.

<table>
<thead>
<tr>
<th>Name</th>
<th>Skip connections</th>
<th>Architecture</th>
<th>Loss</th>
<th>Scheduler</th>
<th>Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_C+D_A</td>
<td>res-atrous</td>
<td>simple</td>
<td>CE + DL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
<tr>
<td>S_C+D_R</td>
<td>regular</td>
<td>simple</td>
<td>CE + DL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
<tr>
<td>S_F+D_A</td>
<td>res-atrous</td>
<td>simple</td>
<td>FL + DL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
<tr>
<td>S_F+FT_A</td>
<td>res-atrous</td>
<td>simple</td>
<td>FL + FTL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
<tr>
<td>D_F+FT_A</td>
<td>res-atrous</td>
<td>double</td>
<td>FL + FTL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
<tr>
<td>D_F+FT_A_M</td>
<td>res-atrous</td>
<td>double</td>
<td>FL + FTL</td>
<td>no - Manual</td>
<td>3</td>
</tr>
<tr>
<td>D_F+FT_A_3</td>
<td>res-atrous</td>
<td>double</td>
<td>FL + FTL</td>
<td>ReduceOnPlateau</td>
<td>3</td>
</tr>
<tr>
<td>D_F+D_A</td>
<td>res-atrous</td>
<td>double</td>
<td>FL + DL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
<tr>
<td>D_C+D_A</td>
<td>res-atrous</td>
<td>double</td>
<td>CE + DL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
<tr>
<td>D_F+FT_R</td>
<td>regular</td>
<td>double</td>
<td>FL + FTL</td>
<td>ReduceOnPlateau</td>
<td>2</td>
</tr>
</tbody>
</table>
A COMPARISON OF DIFFERENT U-NET MODELS FOR SEGMENTATION OF OVERLAPPING ORGANOIDS

Table 5.3: Testing results of the 10 models for the overlapping channel based on several evaluation metrics.

<table>
<thead>
<tr>
<th>Name</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
<th>Jaccard Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_C+D_A</td>
<td>0.82</td>
<td>0.51</td>
<td>0.32</td>
<td>0.35</td>
<td>0.27</td>
</tr>
<tr>
<td>S_C+D_R</td>
<td>0.85</td>
<td>0.54</td>
<td>0.42</td>
<td>0.42</td>
<td>0.34</td>
</tr>
<tr>
<td>S_F+D_A</td>
<td>0.83</td>
<td>0.50</td>
<td>0.37</td>
<td>0.38</td>
<td>0.30</td>
</tr>
<tr>
<td>S_F+FT_A</td>
<td>0.84</td>
<td>0.50</td>
<td>0.45</td>
<td>0.43</td>
<td>0.34</td>
</tr>
<tr>
<td>D_F+FT_A</td>
<td>0.83</td>
<td>0.50</td>
<td>0.38</td>
<td>0.39</td>
<td>30</td>
</tr>
<tr>
<td>D_F+FT_A_M</td>
<td>0.84</td>
<td>0.51</td>
<td>0.37</td>
<td>0.38</td>
<td>30</td>
</tr>
<tr>
<td>D_F+FT_A_3</td>
<td>0.83</td>
<td>0.50</td>
<td>0.35</td>
<td>0.36</td>
<td>0.28</td>
</tr>
<tr>
<td>D_F+D_A</td>
<td>0.83</td>
<td>0.51</td>
<td>0.40</td>
<td>0.40</td>
<td>0.31</td>
</tr>
<tr>
<td>D_C+D_A</td>
<td>0.84</td>
<td>0.53</td>
<td>0.44</td>
<td>0.43</td>
<td>0.34</td>
</tr>
<tr>
<td>D_F+FT_R</td>
<td>0.84</td>
<td>0.52</td>
<td>0.41</td>
<td>0.41</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Table 5.4: Testing results of the 10 models for the mask channel based on several evaluation metrics.

<table>
<thead>
<tr>
<th>Name</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
<th>Jaccard Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_C+D_A</td>
<td>0.74</td>
<td>0.88</td>
<td>0.63</td>
<td>0.72</td>
<td>0.58</td>
</tr>
<tr>
<td>S_C+D_R</td>
<td>0.81</td>
<td>0.87</td>
<td>0.79</td>
<td>0.82</td>
<td>0.71</td>
</tr>
<tr>
<td>S_F+D_A</td>
<td>0.76</td>
<td>0.86</td>
<td>0.71</td>
<td>0.76</td>
<td>0.63</td>
</tr>
<tr>
<td>S_F+FT_A</td>
<td>0.81</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.71</td>
</tr>
<tr>
<td>D_F+FT_A</td>
<td>0.77</td>
<td>0.86</td>
<td>0.71</td>
<td>0.77</td>
<td>0.64</td>
</tr>
<tr>
<td>D_F+FT_A_M</td>
<td>0.76</td>
<td>0.87</td>
<td>0.69</td>
<td>0.76</td>
<td>0.62</td>
</tr>
<tr>
<td>D_F+FT_A_3</td>
<td>0.74</td>
<td>0.86</td>
<td>0.66</td>
<td>0.74</td>
<td>0.59</td>
</tr>
<tr>
<td>D_F+D_A</td>
<td>0.78</td>
<td>0.85</td>
<td>0.75</td>
<td>0.79</td>
<td>0.66</td>
</tr>
<tr>
<td>D_C+D_A</td>
<td>0.81</td>
<td>0.85</td>
<td>0.82</td>
<td>0.83</td>
<td>0.72</td>
</tr>
<tr>
<td>D_F+FT_R</td>
<td>0.79</td>
<td>0.87</td>
<td>0.75</td>
<td>0.79</td>
<td>0.67</td>
</tr>
</tbody>
</table>

5.4.2.1 Residual - Atrous Skip Connections

The first experiment assists the effect of the skip connections type. In both studied scenarios (a simple architecture with CE + DL and a double architecture with FL + FTL) adding the residual-atrous skip connections increases the training final F1 score (Figure 5.6) and decreases the loss. While these results concern the overlapping channel, the mask channel has very similar behaviour. However, this result does not generalize in testing. Regardless of the architecture, adding the residual-atrous block to the features of the encoder before concatenation always decreases the testing F1 score both in the mask (Table 5.4) and overlap (Table 5.3).
The score decreases from 0.42 ($S_C+D_R$) to 0.35 ($S_C+D_A$) in the overlapping channel for a simple U-Net and from 0.41 ($D_F+FT_R$) to 0.39 ($D_F+FT_A$) in a double U-Net. In the case of the mask channel, the decrease is from 0.82 ($S_C+D_R$) to 0.72 ($S_C+D_A$) in a simple architecture and from 0.79 ($D_F+FT_R$) to 0.77 ($D_F+FT_A$) in a double architecture. For this comparison, the last value of the F1 score during training is used, together with the average F1 score obtained on the 88 images that comprise the test set.

Figure 5.6: An F1 score comparison between models using the residual-atrous skip connections versus the regular connections for the overlapping channel. The orange and red models use a simple architecture and the CE + DL loss combination, while the blue and pink models have a double architecture and a FL + FTL loss (Table 5.2).

5.4.2.2 **Loss Functions**

The second set of experiments investigates three different combinations of losses in the case of a simple U-Net. Figure 5.7 shows that using increasingly complex combinations of losses during training has a beneficial effect on the F1 score in the case of a simple U-Net model. Table 5.3 shows an increase of 0.08 and 0.05 in the F1 score for the test set, from 0.35 ($S_C+D_A$) to 0.38 ($S_F+D_A$) and to 0.43 ($S_F+FT_A$), attributed to the improved combination of losses. A similar increasing perspective can be observed in Table 5.4. The comparison of using different losses for the double U-Net is presented in section 5.4.2.3.
5.4.2.3 Model Architecture

To further study the effect of different losses on the architecture types, histograms 5.8 and 5.9 show the final F1 score achieved during training and the average F1 score during testing, respectively. The F1 score increases during training for the double architecture with more complex losses (D_F+FT_A), while the average F1 score on the test set is not high and decreases with complex losses. The F1 score for the simple U-Net also increases during training and this behaviour remains consistent in testing. In comparison to section 5.4.2.2, a decrease on the F1 score is observed when using complex losses for the double U-Net. Table 5.3 reports a 0.03 and 0.04 decrease in the F1 score for the test set, from 0.43 (D_F+FT_A) to 0.40 (S_F+D_A) and to 0.39 (S_C+D_A), respectively. In summary, the simple U-Net achieves the highest scores with complex losses, while the double U-Net performs best with simple losses.

5.4.2.4 Learning Rate and Scheduler

The last experiment studies the effect of the learning rate scheduler, as well as experiments with a larger number of training epochs. Empirical results show that using the scheduler (D_F+FT_A_3) improves performance during training and that training for additional time increases the F1 score significantly. In fact, the model D_F+FT_A_3 achieves the highest F1 score during training of 0.63
for the overlapping channel, and 0.85 for the mask channel. The loss function converged to 0.08 for the multi-class channel and 0.01 for the binary channel. The same models achieves a testing F1 score of 0.36 (Table 5.3). Nonetheless, manual change of the learning rate (D_F+FT_A_M) was slightly better with F1 score of 0.38, when evaluated on the test set, as can be seen in Table 5.3.

5.4.3 Discussion

Figure 5.10 presents the prediction results of all 10 models for 5 randomly selected images from the test set. Overall, the binary mask has a much more accurate segmentation than the overlapping mask (also seen in Tables 5.3 and 5.4), as it
Figure 5.10: The output of the 10 models (Table 5.2) on 5 random images from the test set. The best models are S_F+FT_A and D_C+D_A. The first one uses a simple architecture with a complex combination of losses FL + FTL, while the second one has a double architecture and simpler losses CE + DL.

is uncomplicated for the model to differentiate between 2 classes, compared to 3 in the multi-class task. Since the true annotations are not very detailed, some large organoids are missing from the predicted segmentation. Some networks also predict extra organoids that do appear in the original image but are not segmented in the true mask (for example, the top-left organoid in the first row). This contributed to a decrease in the F1 scores. The difference in class frequency between the train and the test set (Table 5.1) is also an explanation as to why the test results are generally lower than those obtained in training. The best two models in terms of the F1 score, as indicated in Tables 5.3 and 5.4 are S_F+FT_A and D_C+D_A. The first one uses a simple architecture with the complex combination of losses FL + FTL, while the second one has a double architecture and simpler losses CE + DL. One main point that can be accentuated here is that, while complexity can increase performance, too much complexity is detrimental.

5.5 CONCLUSION AND FUTURE WORK

The high level of overlapping in organoids represents a serious challenge in biomedical segmentation tasks. While several deep learning networks have achieved state-of-the-art performance when it comes to binary segmentation, overlapping objects are still an impediment to an accurate diagnosis of many diseases. The encoder-decoder architecture of the U-Net [141], as well as the skip connections represent useful additions, which resulted in high performance on
many datasets. The current study proposed a variation of the U-Net that solves the overlapping organoids task in a severely imbalanced dataset. Ten different models were trained and compared with respect to different additions, including the atrous skip-connections, model architecture, loss functions and scheduler. The novelty of the current work is by exploring and investigating ways to segment overlapping objects, especially on biomedical images. To the authors’ knowledge, there exists no such work in biomedical literature.

There is still a room for improvements by creating accurate and faithful true masks that include large organoids and overlaps. Moreover, with more balanced datasets, the network can be trained to differentiate between different levels of overlaps. In terms of the segmentation process, several studies obtained good results when separating the detection process of the boundaries from that of the center. [169] used single-path voting and mean-shift clustering for center localization and an interactive model for boundary detection. [160] had a double U-Net model with two decoders, for decoding the contours and the interiors separately. A different study [170] attempts to solve the overlapping problem by using active contours and the property that the intensities of touching nuclei are additive. Finally, the best models proposed in this chapter can also be used as part of an online platform, similar to that proposed by [171], that can generate binary and overlapping masks for new data, which can be used as training input for improved models.

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 the data with us.
Part III

APPLICATIONS