



Deep interactive learning for digital pathology

Master's thesis submitted in order to obtain the rank
of master in Computer Science

Author:
Ba LE

Academic supervisor:
Raphaël MARÉE

Abstract

In many biomedical applications, manual annotations of whole slide images take a tremendous amount of time. In the computer vision literature, semi-automatic tools using deep learning, known as deep interactive learning, have emerged to speed up the annotation process. These semi-automatic tools exploit the interactions of the annotators in various forms to produce the annotations more rapidly. In recent years, deep interactive learning seems to gain more attention for its performance. However, do the additional information provided by the annotators help to improve the results of automatic tools? An exploration in the literature was made, resulting in the finding of a promising architecture, named NuClick, which uses the scribbles of the annotators in combination with the images to produce decent annotations more quickly. In this thesis, results of the conducted experiments on various datasets show that the additional information provided by the scribbles improve drastically the performance of the segmentation for tissues, such as bronchi, glands, or infiltrations. However, this interactive approach fails to produce accurate segmentation for more complex tissues, such as tumours or inflammations. Also, results indicate that the quality of the scribbles highly influences the produced segmentation. Therefore, care should be taken when the annotators scribble the objects of interest. These results tend to support the benefit that can be gain from the interactions of the annotators, although this thesis shows that there is room for improvements with these semi-automatic tools.

Acknowledgements

First and foremost, I would like to sincerely thank my academic supervisor Raphaël Marée for his guidance, advice, proofreading, and helpful feedback throughout the realisation of this thesis.

Then, I would like to express my gratitude to Romain Mormont for his small feedback and suggestions at the early stage of the thesis.

I am grateful for the interesting discussions with Pierre Geurts and also for the precious advice about some implementation details, which allow a huge time gain during the realisation of the experiments.

Next, I would also like to thank Ruben Ulysse for his suggestions and valuable comments about the writing of this thesis.

A special thanks to the Alan Cluster team for giving me access to their GPUs. Without them, the numerous conducted experiments would have not been possible.

Finally, I would like to thank my friends and family for their support, especially my twin brother.

Contents

1	Introduction	1
2	Deep learning and state of the art	5
2.1	Deep learning	5
2.1.1	Supervised learning	5
2.1.2	Neural network	5
2.1.3	Training a neural network	5
2.1.4	Convolutional neural network	6
2.1.5	Residual block	8
2.1.6	U-Net architecture	8
2.2	State of the art	9
2.2.1	Feedback-based methods	9
2.2.2	Click-based methods	13
2.2.3	Bounding box-based methods	16
2.2.4	Contours-based methods	17
2.3	Discussion	19
2.3.1	Results of the approaches	19
2.3.2	Architecture choice	21
3	Methodology	22
3.1	Overview	22
3.2	Data acquisition	22
3.2.1	Dataset structure	23
3.2.2	Acquisition	23
3.2.3	Data processing	25
3.2.4	Data splitting	25
3.3	Segmentation model: NuClick	26
3.3.1	Inclusion and exclusion map	26
3.3.2	Loss function	29
3.3.3	Post-processing	30
3.4	Evaluation metrics	31
3.4.1	Intersection over the union	31
3.4.2	Dice coefficient	32
3.4.3	Hausdorff distance	33
3.5	Implementation details	33
4	Experiments and results	35
4.1	Datasets	35
4.1.1	CHALLENGE-CAMELYON16-TRAIN	36
4.1.2	CHALLENGE-GLAS-2015	36
4.1.3	CHU-ANAPATH-NST-DL	36

4.1.4	ULG-LBTD-NEO04	36
4.1.5	ULG-LBTD-NEO13 (3)	36
4.2	Replication of the original study	42
4.3	Experiments protocol	43
4.3.1	Datasets	43
4.3.2	Model training	43
4.3.3	Model evaluation	43
4.3.4	Assessment standard	43
4.4	Annotations analysis	44
4.4.1	Quantity analysis	44
4.4.2	Quality analysis	56
4.5	Robustness analysis	66
4.5.1	Bronchus	66
4.5.2	Gland	66
4.5.3	Inflammation	66
4.6	Model analysis	68
4.6.1	U-Net comparison	68
4.6.2	Absence of signal maps	68
4.6.3	Automatic NuClick architecture	68
4.6.4	Results	68
4.7	Discussion	71
5	Conclusion and perspectives	73
5.1	Perspectives	73
5.1.1	Improvements	73
5.1.2	Integration to Cytomine	74
	List of Figures	75
	List of Tables	78
	Bibliography	79
A	Neural network architecture	81
A.1	NuClick architecture	81
A.2	U-Net architecture	81
B	Quantity analysis	82
B.1	Bronchus experiments	83
B.2	Gland experiments	84
B.3	Inflammation experiments	87
B.4	Infiltration experiments	88
B.5	Tumour experiments	89
C	Quality analysis	90
C.1	Illustrations of the scribbles	90
C.2	Performance	94

Chapter 1

Introduction

Pathology is the science that studies the causes and effects of diseases or illnesses through the investigation of tissues, bodily fluids, organs, or in autopsy. In the medical sector, pathologists play a vital role in diagnosing human diseases and finding a treatment accordingly. The process of diagnosis is done through the examination of microscope slides. However, numerous drawbacks come with this process, such as the need for a microscope, at most one slide can be examined at a time, only limited analyses can be performed, storing the slides is a laborious task, and many more. A subfield, named digital pathology, has emerged to alleviate most of the aforementioned issues. More precisely, digital pathology refers to the process of digitising the microscope slides. With these digitised slides, called whole slide images, there is no need for a microscope anymore, an access to a computer is sufficient enough to view the slides at any time more easily. Several slides can be analysed and viewed at the same time and various analyses can be performed simultaneously. These advantages speed up the process of examinations and diagnoses done by pathologists. An example of whole slide image is shown in Figure 1.1.

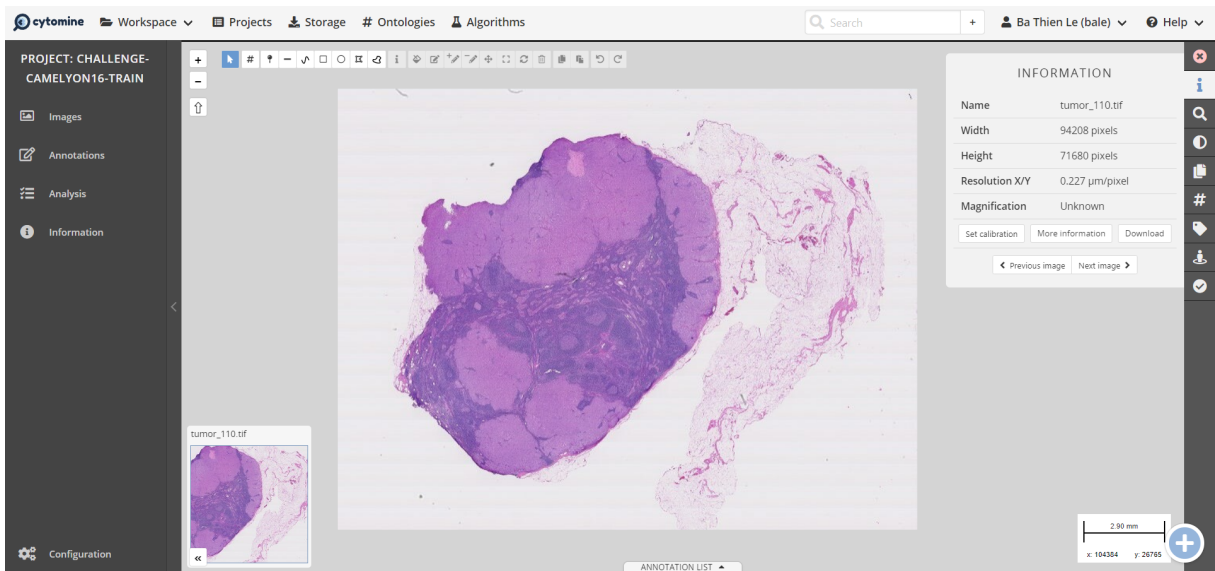


Figure 1.1: Illustration of a whole slide image of a sentinel lymph node shown in the Cytomine web interface.

After the examination of whole slide images, regions of interest have to be annotated so that further analysis can be performed later. The task of annotating regions of interest is a tremendous, time-consuming, and error-prone process for human annotators. It usually requires experts to annotate medical images which can result in expensive costs. To reduce the effort of the annotators, automatic annotation tools and algorithms using deep learning have emerged

to speed up this process. However, these automatic tools do not yet produce very accurate annotations that fully satisfy the experts. To this end, semi-automatic annotations can be designed, where the interactions of the human annotators are exploited to provide more accurate annotations. Typically, an algorithm or a tool produces an initial annotation of the object of interest. This initial annotation is then reviewed by an expert to correct the mistakes and to produce the final annotation [Marée et al., 2014].

One type of annotations that is widely used on whole slide images is the semantic segmentation. In short, it is the task of partitioning a given image into different regions. Each of these regions is then given a semantic category. In this thesis, there are only two categories, also known as binary semantic segmentation, namely the foreground or object of interest, and the background representing all the other pixels in the image. This principle is going to be applied to multiple images and object types, as illustrated in Figure 1.2 and Figure 1.3.

When an annotator has a limited number of annotations and wants to have more annotations without spending excessive time on the annotation process, this thesis aims at reducing the workload of the annotator by speeding up the process using a semi-automatic annotation tool. It also aims at determining whether the interactions of the annotators provide effective information or not. Illustrations of the different tissues used in this thesis are shown in Figure 1.3

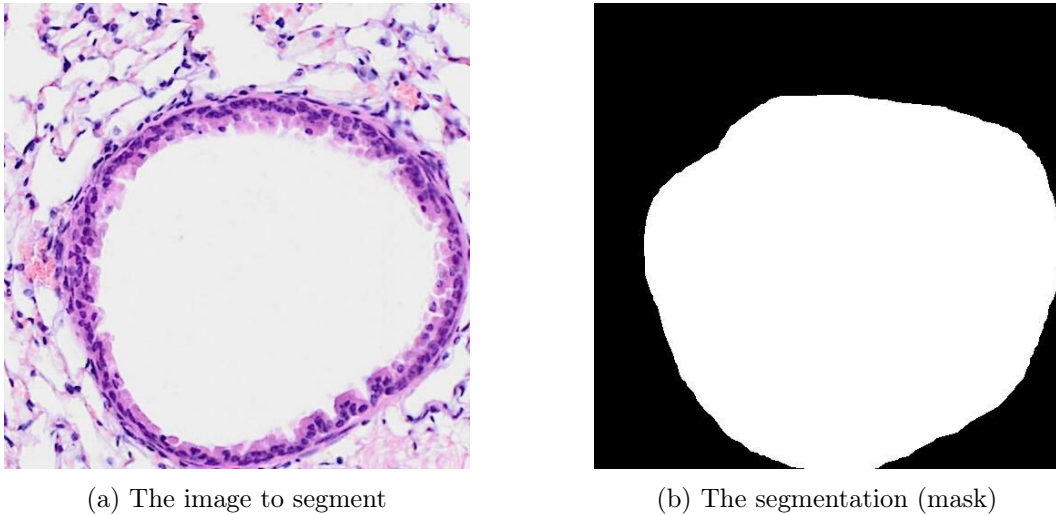


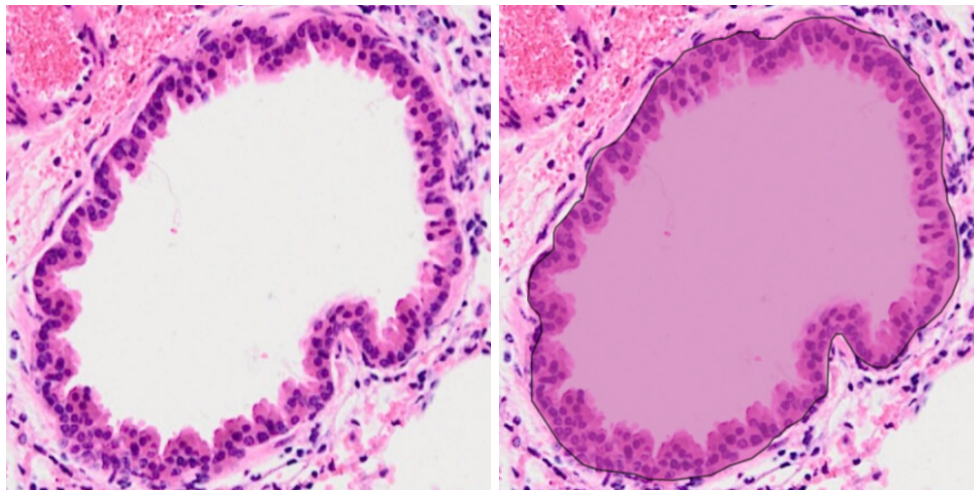
Figure 1.2: Example of the segmentation of a bronchus, where the foreground corresponds to the bronchus and the background to all the other pixels in the image.

To store and access these whole slide images easily, various web-based platforms have emerged to cope with the needs of fast and intuitive access. One of these platforms is Cytomine. Basically, it is a web-based application that allows collaborative analysis of multi-gigapixel images [Marée et al., 2016]. It is open-source and is composed of three main entities:

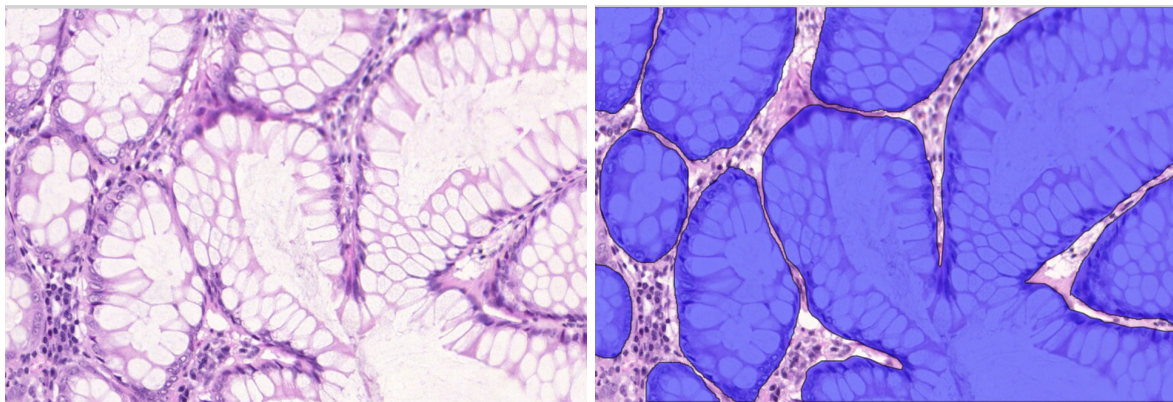
1. **Open Source Repository:** “Open-source rich internet application for collaborative analysis of multi-gigapixel images using machine learning” [Cytomine Corporation SA, 2021]. This entity is the main open source software, i.e., [Cytomine](#) with its documentation available at their [website](#).
2. **Open Company:** the company [Cytomine](#) that contributes to the open-source project, is in charge of promoting it, coordinating the open-source community, and providing products and services related to the open-source software.
3. **Open Research:** a research and development department in machine learning, image bioinformatics, and big data, [Cytomine ULiège R&D](#) [Marée et al., 2016]. The department contributes to the open-source project. It is located at the [Montefiore Institute \(University of Liège\)](#), Belgium. This thesis was conducted within this department, with the R&D team.

The Cytomine web user interface, shown in Figure 1.1, provides features that allow users to visualise the whole slide images and the annotations of these images, to select the annotations of a particular user, and many more. After the selection of the desired annotations, users can download a CSV, pdf, or an Excel file containing the complete information about the annotations, such as the project id, the id of the annotation, the filename of the whole slide image, the term of the annotation, its coordinates, etc. Cytomine also offers an API and Python and Java clients to import/export data.

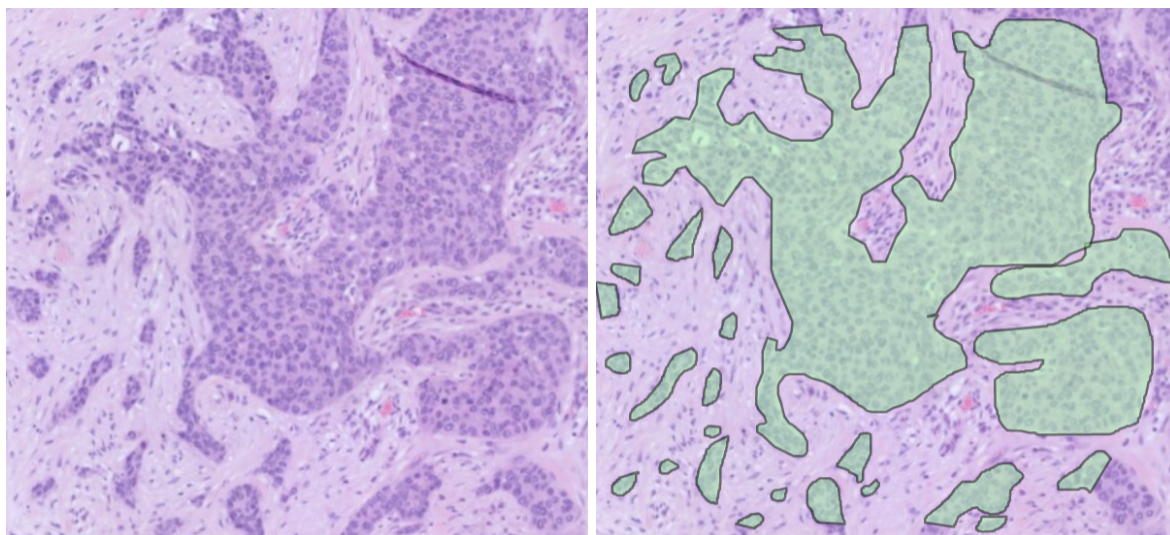
This Master Thesis report is organised as follows. First, chapter 2 presents some concepts related to deep learning and a review of the literature is made about semi-automatic learning. Then, chapter 3 describes the methodology developed. More precisely, the different steps of the methodology are explained: the acquisition of the datasets, the neural network used along with its specificities, the description of the metrics used for the performance assessment, and the implementation details. After that, chapter 4 details the various conducted experiments with the developed methodology and their results are discussed. Finally, chapter 5 concludes this thesis and presents future perspectives.



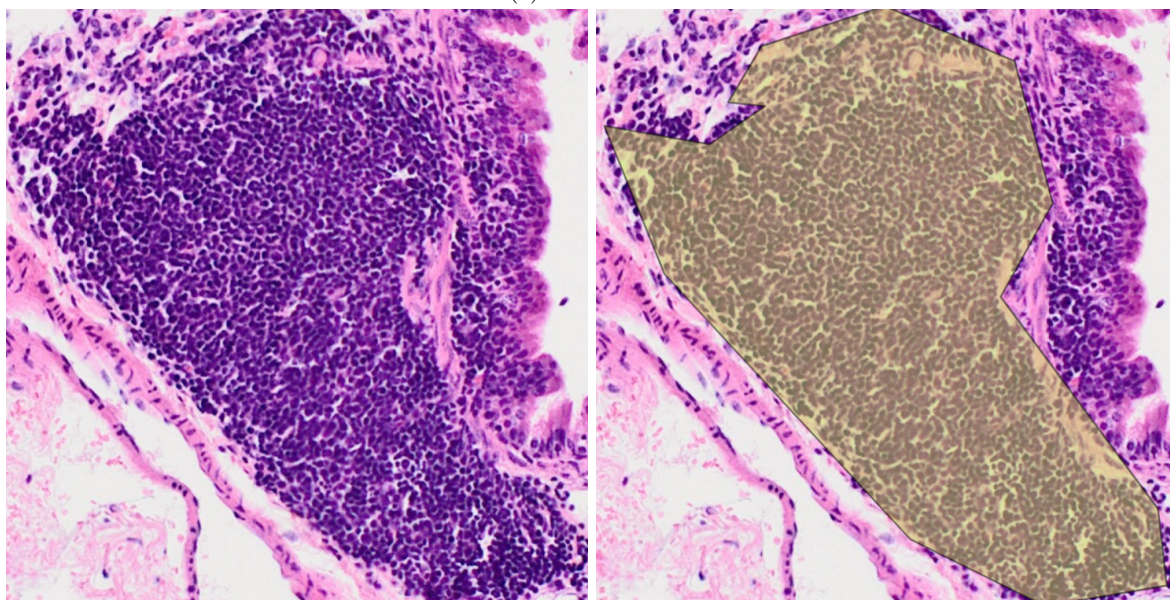
(a) Bronchus



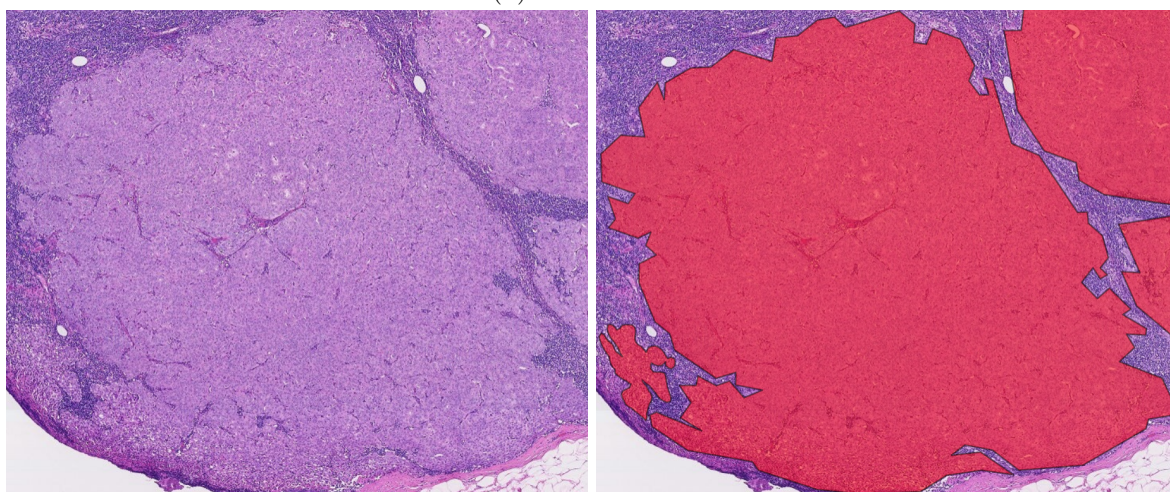
(b) Gland



(c) Infiltration



(d) Inflammation



(e) Tumour

Figure 1.3: The type of tissues used in this thesis with their corresponding segmentation mask.

Chapter 2

Deep learning and state of the art

In this chapter, section 2.1 reviews the necessary theoretical background in order to understand the thesis in general. Then, section 2.2 presents state-of-the-art approaches that are centred around the semi-automatic segmentation approach. Finally, section 2.3 concludes this chapter by motivating the selection of one specific approach.

2.1 Deep learning

In this section, some theoretical background is given to better understand all the subsequent chapters.

2.1.1 Supervised learning

The task of supervised learning is to predict targets, also called labels, given some inputs. The inputs are typically denoted as \mathbf{x} and the targets as \mathbf{y} . A collection of these input and target pairs is called a dataset, $\{(\mathbf{x}_i, y_i)\}_{i=0}^{n-1}$, where n represented the number of pairs. To be more precise, supervised learning aims to produce a model that approximates at best the targets given some inputs. In the context of binary semantic segmentation, the inputs \mathbf{x} are generally images represented by a matrix of shape $H \times W \times C$, where H and W denote, respectively, the height and the width of the images, and C denotes the channels of the image. Usually, the number of channels are 3 representing the encoding of an image in the RGB colour space. Regarding the targets y , they are binary images, also called masks or segmentations, represented by a matrix of shape $H \times W$, where H and W denote, respectively, the height and the width of the mask. The targets are also called ground truth masks in this context.

2.1.2 Neural network

The simplest neural network in deep learning is called a multilayer perceptron (MLP). It consists of several layers of neurons, also known as perceptrons. Each layer is connected to the previous and the next layer. For the first layer, it receives the input \mathbf{x} and the last layer produces the desired output \mathbf{y} . The layers between the input and output layer are more commonly addressed as hidden layers. Graphically, the neural network is usually represented as a graph where the nodes are the neurons and the edges are the connections between these nodes. Each of these edges holds what is called a **weight** that is tuned for the network to produce the desired output. A simple multilayer perceptron is illustrated in Figure 2.1.

2.1.3 Training a neural network

To train a neural network for producing the desired output, the weights of the neural network are tuned until the desired output is achieved given an input. For instance, the input is an image

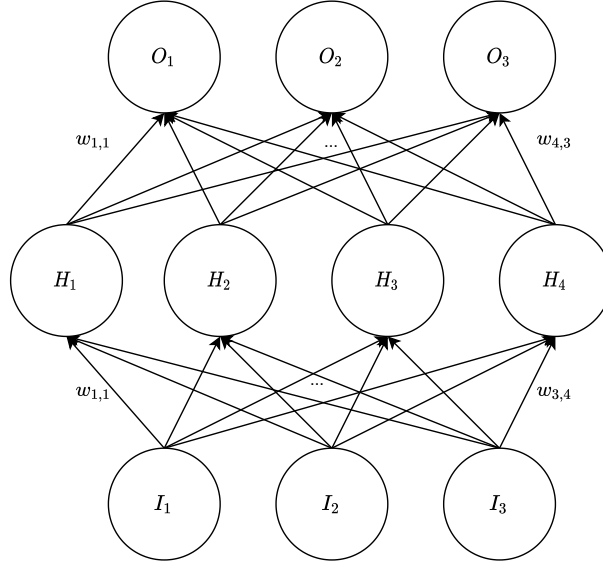


Figure 2.1: A multilayer perceptron with 3 layers of neurons.

containing a gland and the desired output is the segmentation mask of the gland. First, the loss function is introduced, which is needed for the training. Then, the training procedure is explained in details.

Loss function

The loss function evaluates how well a model performs on a dataset for a given task during the training phase. It measures the distance between the real and predicted value of a target. The objective is to minimise this loss function by training the model on the dataset for an arbitrary number of epochs. The loss is generally a non-negative number that represents the distance. The lower this number is, the better the predicted values are, where a loss of 0 represents a perfect prediction of the target.

Training

A classical training procedure is presented in Algorithm 1. The first step is to determine the number of epochs to use. Basically, an **epoch** is when the entire dataset is passed forward and backwards through the neural network once. Passing the entire dataset at once is not possible due to memory issues. Therefore, small groups of the dataset, more commonly called a **batch**, is passed until the entire dataset is passed. Most of the time, one epoch is not enough to update the weights because the neural network architectures are more complex than a simple multilayer perceptron. For each epoch, batches of input and target pairs are taken from the dataset. Then, the neural network predicts the presumable target, called predictions or outputs. The loss function is used to evaluate the quality of the predictions with respect to the targets. Then, the weights of the neural network are tuned with the help of a method called an **optimiser**, which tries to minimise the loss. An example of optimisers is the stochastic gradient descent. This process is repeated for each epoch until the specified number of epochs is reached.

2.1.4 Convolutional neural network

In semantic segmentation, a simple multilayer perceptron is unable to produce decent segmentation masks. A better and more complex architecture for dealing with images is the convolutional neural network (CNN). More precisely, a convolutional neural network consists of several convolutional blocks. One of these blocks is consists of a convolutional layer in place of the simple

Algorithm 1 Training procedure

Input: dataset, epochs

```
1: model  $\leftarrow$  NEURALNETWORK()
2: optimiser  $\leftarrow$  OPTIMISER()
3: criterion  $\leftarrow$  LOSSFUNCTION()
4: for epoch in epochs do
5:   for inputs, targets in dataset do
6:     predictions  $\leftarrow$  model(inputs)
7:     loss  $\leftarrow$  criterion(predictions, targets)
8:     loss.backward()
9:     optimiser.step()
10:  end for
11: end for
12: return model
```

neuron layer, followed by a batch normalisation, and ends with an activation function. These three notions are going to be explained in the subsequent sections.

Convolutional layer

The fundamental operation that a convolutional layer uses is called a convolution¹. For a 3D tensor, e.g., a coloured image, $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ and a convolutional kernel $\mathbf{k} \in \mathbb{R}^{C \times h \times w}$, the discrete convolution $\mathbf{x} \circledast \mathbf{k}$ produces a 2D tensor of size $(H - h + 1) \times (W - w + 1)$ such that

$$\sum_{c=0}^{C-1} (\mathbf{x}_c \circledast \mathbf{u}_c)[i, j] = \sum_{c=0}^{C-1} \sum_{m=0}^{h-1} \sum_{n=0}^{w-1} \mathbf{x}_{c,m+i,n+j} \mathbf{u}_{c,m,n} \quad (2.1)$$

The final output tensor $\mathbf{o} \in \mathbb{R}^{(H-h+1) \times (W-w+1)}$, called output feature map, is computed by introducing a bias term $\mathbf{b} \in \mathbb{R}^{(H-h+1) \times (W-w+1)}$ such that

$$\mathbf{o} = \mathbf{b} + \mathbf{x} \circledast \mathbf{k} \quad (2.2)$$

where the bias term \mathbf{b} and the convolutional kernel \mathbf{k} are shared parameters to learn.

Batch normalisation

During the training phase, the input distribution is subject to change due to the change in the network parameters. In this case, the hidden layers try to adapt to the new distribution. The consequence of this phenomenon causes a slow down in the learning process, i.e., it will take longer to converge to a global minimum. This phenomenon is called an *internal covariate shift*. To cope with this issue, the batch normalisation [Ioffe and Szegedy, 2015] was introduced. Batch normalisation consists in computing the mean and variance for each batch, which is used for normalising the features by shifting and scaling them.

Activation function

The motivation of an activation function is to introduce non-linearity into the output of the neuron. In the convolutional block, the rectified linear unit, better known as ReLU, activation function is usually used:

$$\text{ReLU}(x) = \max(0, x) \quad (2.3)$$

¹It is more commonly known as the cross-correlation operator.

2.1.5 Residual block

More complex blocks can be designed to solve various issues. One of them is called a residual block. The residual blocks were first introduced in the ResNet architecture [He et al., 2015] to solve the issue known as the degradation problem. Basically, stacking identity layers, i.e., a layer that simply maps inputs to outputs, to the network causes a degradation in performance. Figure 2.2 illustrates the concept of a residual block. The motivation to add this residual block is that it allows the design of deeper networks without the risk of having the vanishing gradient effect.

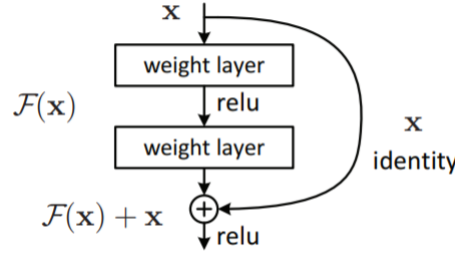


Figure 2.2: Residual block (Source: [He et al., 2015]).

2.1.6 U-Net architecture

A well known convolutional neural network architecture is U-Net [Ronneberger et al., 2015], which has proven to be very efficient for various medical segmentation tasks. Its neural network architecture is shown in Figure 2.3. It consists of two phases, named the contraction and the expansion phases. As can be seen in Figure 2.3, on the left part, it consists of the contraction phase. First, the input image is fed into the network. By going all the way down through the layer, the input image's height and width are reduced gradually and the feature map's size grows gradually. On the right part of the architecture, it consists of the expansion phase, where the input image's height and width are growing gradually until they reach the specified sizes and the feature map decreases in size progressively until it also reaches the specified size.

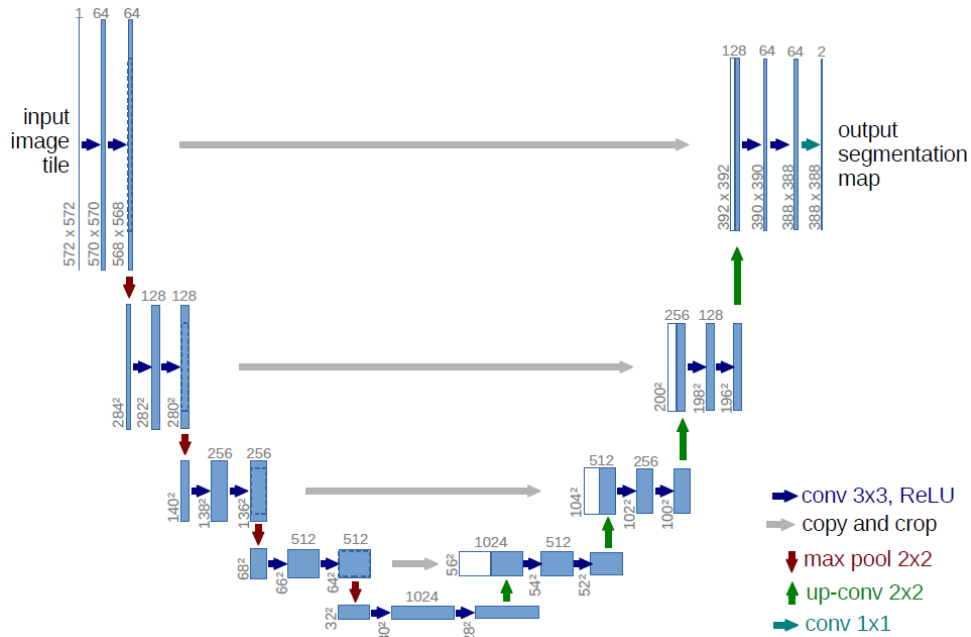


Figure 2.3: U-Net neural network architecture (Source: [Ronneberger et al., 2015]).

2.2 State of the art

In this section, various state-of-the-art approaches regarding semi-automatic and interactive learning are reviewed. Although this thesis focuses on biomedical images, papers outside this domain are also reviewed to identify ideas that could be applied to this field. Different methods are exploited to add the interactions of the annotator in their approach. In subsection 2.2.1, feedbacks from the annotator are used on the initial segmentation to improve the quality of the network for future segmentation. In subsection 2.2.2, some methods incorporate directly the interactions of annotators, called guiding signals, which are basically clicks provided by the annotators. Next, subsection 2.2.3 presents methods where the annotator provides a bounding box around the object of interest on the image, which is used to segment the object. Lastly, methods that focus more on the contour of the objects are presented in subsection 2.2.4.

2.2.1 Feedback-based methods

HistomicsML2

The paper *HistomicsML2: Interactive classification of whole-slide imaging data for cancer researchers* [Lee et al., 2021] presents a complete approach of the segmentation task as shown in Figure 2.4. The first step of the approach is to perform a superpixel segmentation on the whole slide image, which is then used to extract patches of tissue regions as illustrated by (A) in the overview. These extracted patches are fed to the first network to produce feature maps. The second step consists in producing a segmentation by a second network that uses these feature maps. In the second step, the annotator gives feedbacks about the produced segmentation to the network to improve the predictions, iteratively. These feedbacks are the annotations of the regions that are not well segmented.

The architecture of HistomicsML2 is composed of two parts. The first one is a pretrained VGG16 [Simonyan and Zisserman, 2014], which consists of 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers. The authors truncated this network to extract feature maps of size 4096 after the first fully connected layer. The second part uses a multilayer network for superpixel classification, i.e., a neural network with three layers using ReLU activation function, dropout of 30 %, and the output layer uses the sigmoid activation function for class prediction.

Two datasets were used in their validation study, containing lymphocyte infiltration, namely triple-negative breast carcinomas (BRCA) and primary cutaneous melanoma (SKCM). In short, BRCA is a kind of breast cancer and SKCM is a kind of skin cancer.

To evaluate their methods, the accuracy and the area under the curve (AUC) were used, which are defined as

$$\text{Accuracy} = \frac{TP + TN}{P + N} \quad (2.4)$$

where TP and TN denotes, respectively, the number of true positive and true negative pixels predicted, P the number of predicted positive pixels, and N the number of predicted negative pixels and the AUC was calculated using the receiver operating characteristic (ROC) curve:

$$\text{TPR} = \frac{TP}{P}, \quad \text{FPR} = \frac{FP}{N} \quad (2.5)$$

where TPR, FPR, and FP denote, respectively, the true positive rate, the false positive rate, and the false positive predicted pixels.

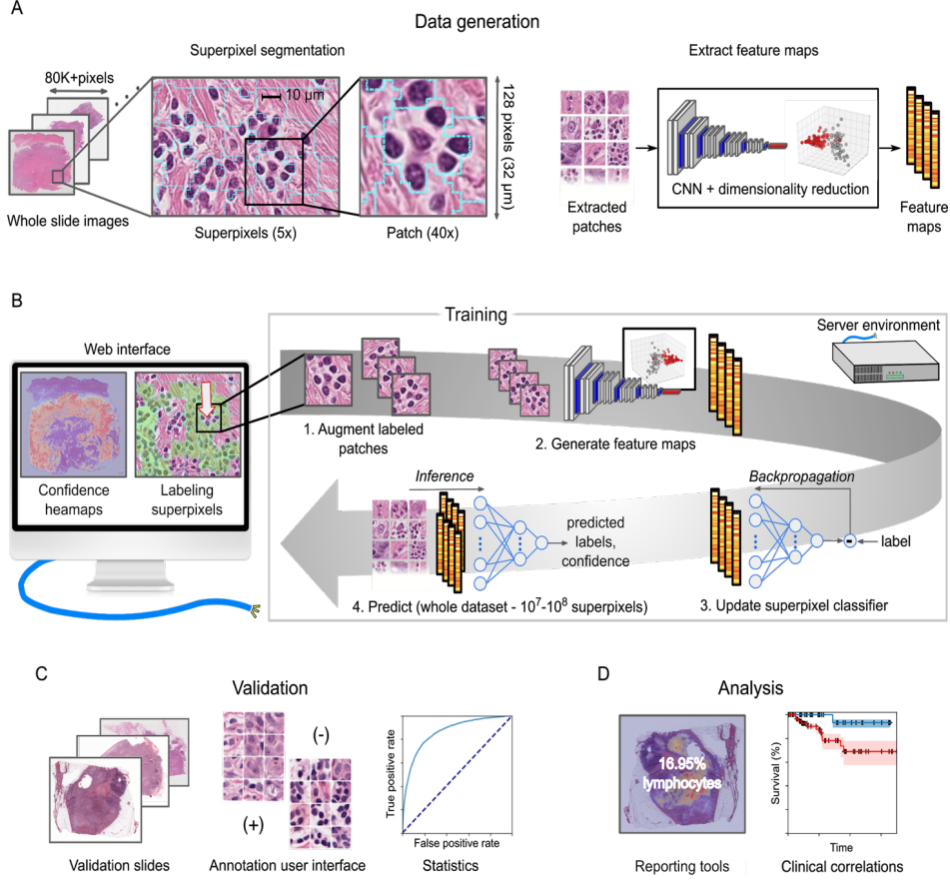


Figure 2.4: Complete pipeline of their software (Source: [Lee et al., 2021]).

DiaL

In the paper *Deep Interactive Learning: An Efficient Labelling Approach for Deep Learning-Based Osteosarcoma Treatment Response Assessment* [Ho et al., 2020], an initial segmentation is produced by the network. The annotator reviews and corrects regions that are not well segmented, which is used to improve the network at the next iteration of the training procedure. This process is repeated until a satisfactory segmentation is achieved. The entire process is illustrated in Figure 2.5.

The neural network architecture employed in this approach is a deep multi-magnification network (DMMN) [Ho et al., 2021] and is depicted in Figure 2.6. In short, the network is a concatenation of several U-Net architectures [Ronneberger et al., 2015]. More precisely, it is composed of three U-Nets, i.e., U-Net-20 \times , U-Net-10 \times , and U-Net-5 \times . Their corresponding input takes whole slide images with a 20 \times , 10 \times , and 5 \times magnifications, respectively. A CONV_BLOCK is composed of two sequences of a convolutional layer followed by a ReLU activation function, a CONV_TR is similar to CONV_BLOCK but contains transposed convolutional layer instead of a convolutional layer. And CONV_FINAL contains a convolutional layer that outputs the number of classes needed, in the case of a binary segmentation, there is only 2 classes, namely the background pixels and the foreground pixels representing the segmented object.

To assess the performance of their model, they have used the error rate as the evaluation metric.

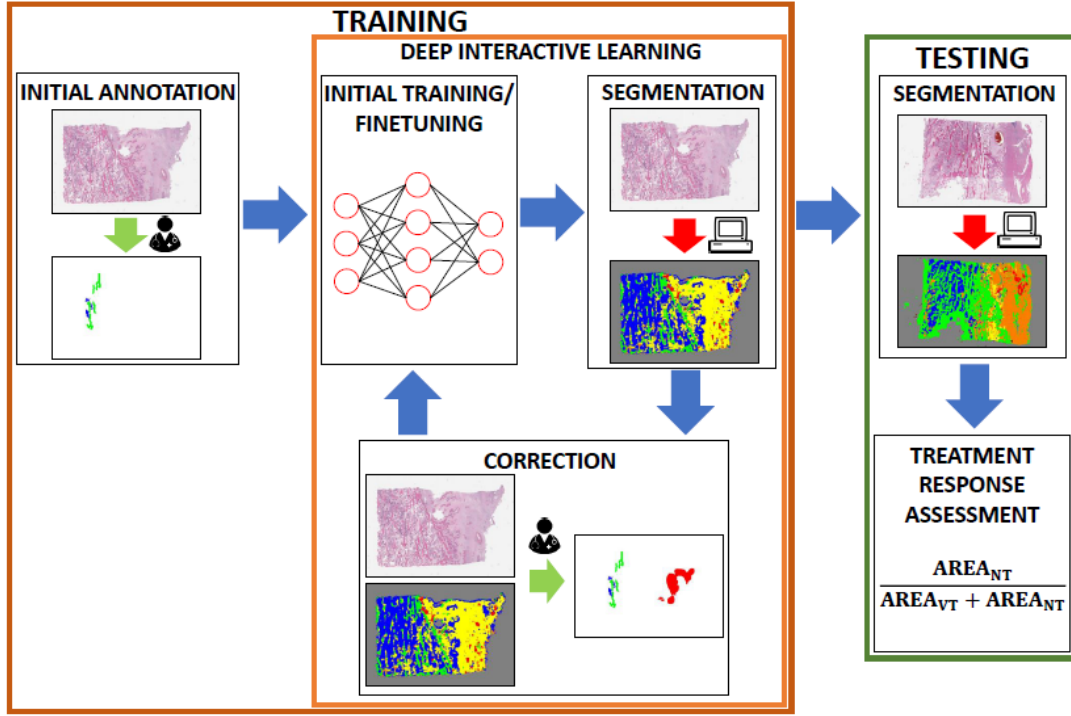


Figure 2.5: Complete pipeline of the framework (Source: [Ho et al., 2020]).

DeepIGeos

The approach proposed in the paper *DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation* [Wang et al., 2019] is in the same spirit as the two other previous approaches, which is that an initial segmentation is produced by a neural network, on which the annotator corrects bad segmented regions. However, these corrections along the initial segmentation are fed into a second neural network to produce the final segmentation, whereas the previous approaches only use the corrections and not the initial segmentation.

The first neural network used for the initial segmentation is called P-Net and is depicted in Figure 2.7 and the second neural network is called R-Net. These two neural networks share the same network architecture except for the difference in input dimensions. More precisely, P-Net and R-Net are inspired from VGG16 [Simonyan and Zisserman, 2014], i.e., the first 13 convolutional layers are grouped into 5 blocks with the first two blocks composed of 2 layers only and the rest of 3 layers.

The authors worked with two types of datasets, namely placenta segmentation from fetal magnetic resonance imaging (MRI) and brain tumour segmentation from fluid-attenuated inversion recovery (FLAIR). Regarding the first dataset, they have collected clinical T2-weighted MRI of 25 pregnant women in the second trimester. As for the second dataset, they have used the brain tumour image segmentation challenge of 2015 (BRATS).

To evaluate the performance of their two-stages model, the Dice coefficient and the average symmetric surface distance (ASSD) were used. The second metric is defined as follows

$$ASSD = \frac{1}{|\mathcal{S}_a| + |\mathcal{S}_b|} \left(\sum_{i \in \mathcal{S}_a} d(i, \mathcal{S}_b) + \sum_{i \in \mathcal{S}_b} d(i, \mathcal{S}_a) \right) \quad (2.6)$$

where $|\cdot|$ denotes the number of elements, \mathcal{S}_a and \mathcal{S}_b denotes, respectively, the set of surface points of the predicted segmentation and its ground truth, and $d(i, \mathcal{S}_b)$ denotes the shortest Euclidean distance between i and \mathcal{S}_b . Basically, the lower the ASSD value the better the performance of the model. The Dice coefficient is going to be explained in details in subsection 3.4.2, page 32.

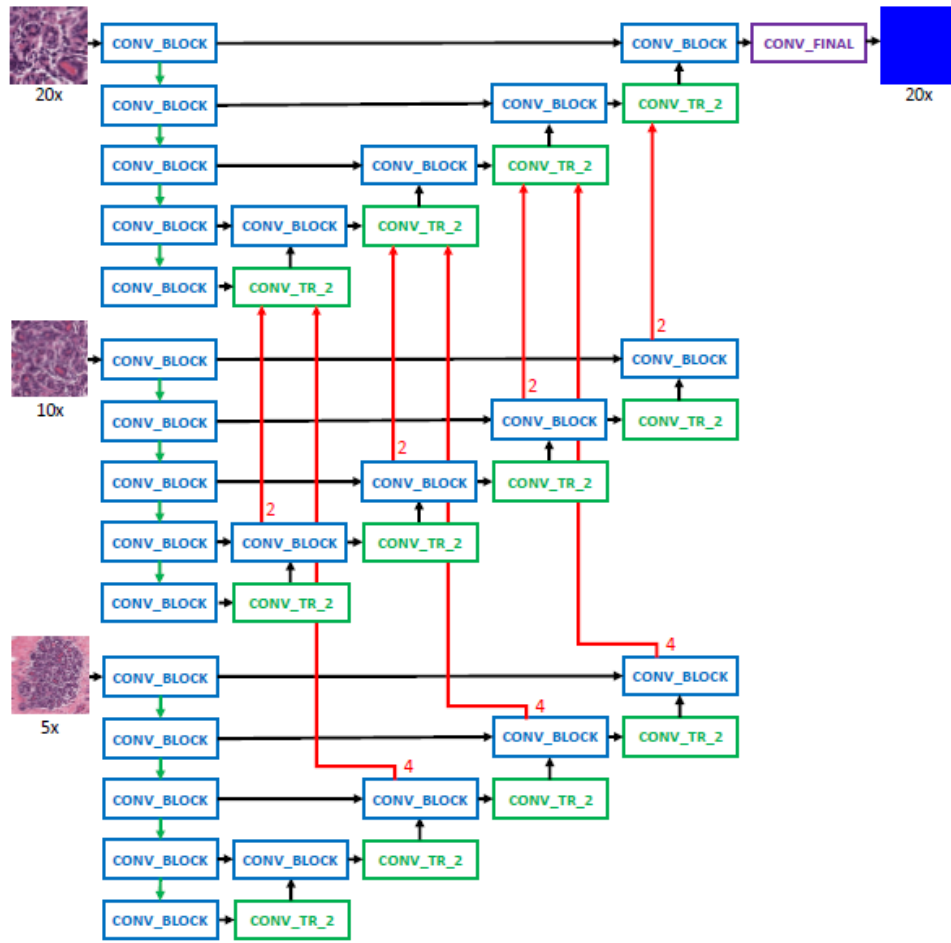


Figure 2.6: Deep Multi-Magnification Network architecture (Source: [Ho et al., 2021]).

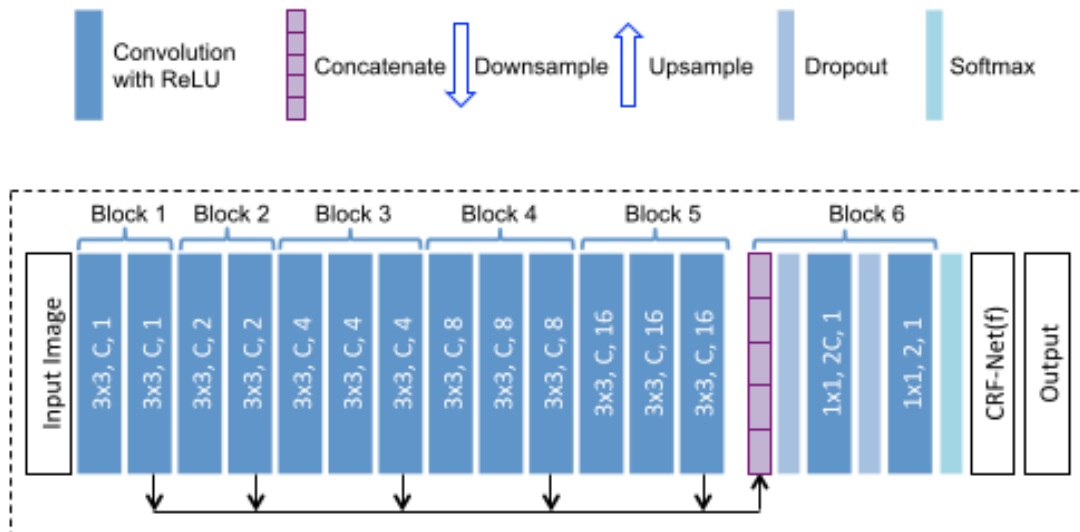


Figure 2.7: P-Net neural network architecture (Source: [Wang et al., 2019]).

2.2.2 Click-based methods

NuClick

The approach proposed in *NuClick: a deep learning framework for interactive segmentation of microscopy images* [Jahanifar et al., 2019; Alemi Koohbanani et al., 2020] is to scribble objects of interest. These scribbles serve as additional signals and are concatenated to the image, which is then fed to the network to produce the segmentation of the scribbled objects of interest. They are divided into two categories, namely inclusion and exclusion maps. In short, when an annotator scribbles the object of interest, an inclusion map is created, which stores the scribble in white pixels in a binary map. Frequently, on a whole slide image, numerous objects of interest close nearby are scribbled by the annotator, another map is also created to store all these scribbles except the scribble present in the inclusion map. Therefore, for each scribble, there is an inclusion map that includes the scribble and an exclusion map that comprises the other scribbles. The purpose of these maps is to avoid closely scribbled objects being segmented by the network as a unique object.

The authors propose a convolutional neural network model, named NuClick, based on the U-Net architecture [Ronneberger et al., 2015]. As illustrated in Figure 2.8, NuClick is composed of several convolutional blocks, residual blocks, and multi-scale convolutional blocks. First, using residual block allows building deeper and more complex network without the risk of vanishing gradient. Then, the motivation to use multi-scale convolutional blocks is to allow the network to segment both large and small objects.

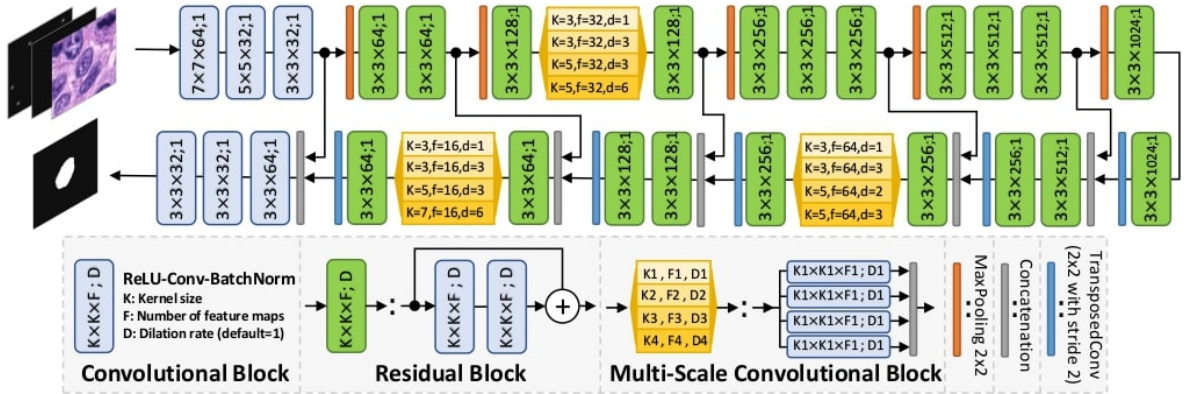


Figure 2.8: NuClick neural network architecture (Source: [Alemi Koohbanani et al., 2020]).

In this work, the authors focus on three fundamental objects in pathology, i.e., nuclei, cells, and glands. On the one hand, for nuclei and cells, one click inside each object is sufficient for NuClick to produce a precise segmentation. On the other hand, since glands are more complex objects, one click inside the object is not enough for the model to yield an accurate segmentation. Therefore, instead of a single click, a scribble is given as a guiding signal for the model to produce an accurate segmentation of the gland.

To put in practice their approach, for the nuclei, they have used the [MonuSeg](#) and [CPM](#) datasets which contain, respectively, 30 and 32 H&E images. Then, concerning the cells, they have synthesised a dataset of 2,689 images consisting of touching white blood cells (WBCs). Finally, concerning the glands, the [GlaS2015](#) and the CRAG datasets were used. The former dataset, namely the gland segmentation in colon histology images (GlaS) dataset, consists of 165 images of stages T3 or T4 colorectal adenocarcinoma images. The latter dataset, CRAG, consists of 213 colorectal adenocarcinoma images. Colorectal adenocarcinoma is a kind of colon cancer.

Lastly, to validate their approach, various metrics reported in the literature have been used. For nuclei and cells, the Aggregated Jaccard Index (AJI), the Dice coefficient, the Hausdorff distance, the Detection Quality (DQ), the SQ, and the Panoptic Quality (PQ) have been employed.

For gland segmentation, the F1-score, the $Dice_{obj}$ coefficient, and the Hausdorff distance were used.

FCNN

In the paper *Interactive segmentation of medical images through fully convolutional neural networks* [Sakinis et al., 2019], the annotator is asked to click within the objects of interest, the so-called *foreground clicks*, and to click wherever the annotator feels that the network is going to be segmenting falsely a region that does not contain the object of interest, called the *background clicks*. In short, the foreground clicks guide the network to focus the segmentation towards these clicks while avoiding area containing background clicks. These guidance signals are used with the images to train the neural network.

The neural network architecture, proposed in this paper, is also inspired by the U-Net architecture [Ronneberger et al., 2015] with several modifications and is depicted in Figure 2.9. The contraction and expansion parts are composed of four convolutional blocks instead of three. Each block is composed of twice the sequence of a convolution, followed by a batch normalisation, and ends with a ReLU activation function.



Figure 2.9: The FCNN architecture (Source: [Sakinis et al., 2019]).

The authors of this work focus on computed tomography (CT) images of the abdomen. Two datasets were used, namely the BCV dataset released during the *multi-atlas labelling beyond the cranial vault* challenge organised by the MICCAI society in 2015 and the MSD dataset released during the *medical segmentation decathlon challenge* also held by the MICCAI society in 2018. The former dataset consists of 30 CT volumes with the corresponding ground truth segmentations and 20 test cases, whereas the latter consists of MRI and CT volumes that are relevant to 10 different segmentation tasks, where only two tasks related to the abdomen were selected.

To assess the performance of their approach, the authors have used the Dice coefficient, the Hausdorff distance, and the mean absolute distance (MAD) metrics.

RefineNet

The paper *Interactive deep refinement network for medical image segmentation* [Kitrungsakul et al., 2020] presents an approach composed of two stages. In the first stage, an initial segmentation of the object of interest is produced by a backbone network. With this initial segmentation, the annotator is asked to provide foreground and background clicks within the objects of interest and regions that do not correspond to the objects, in the same spirit as presented in FCNN [Sakinis et al., 2019]. In the second stage, these guidance signals, also known as seed points, are used with the initial segmentation to compose the input of a second network, the so-called refinement network. This network takes into account the signals provided by the annotators to correct and refine the initial segmentation to produce the final segmentation. The two networks are combined to form the complete architecture called RefineNet as illustrated in Figure 2.10.

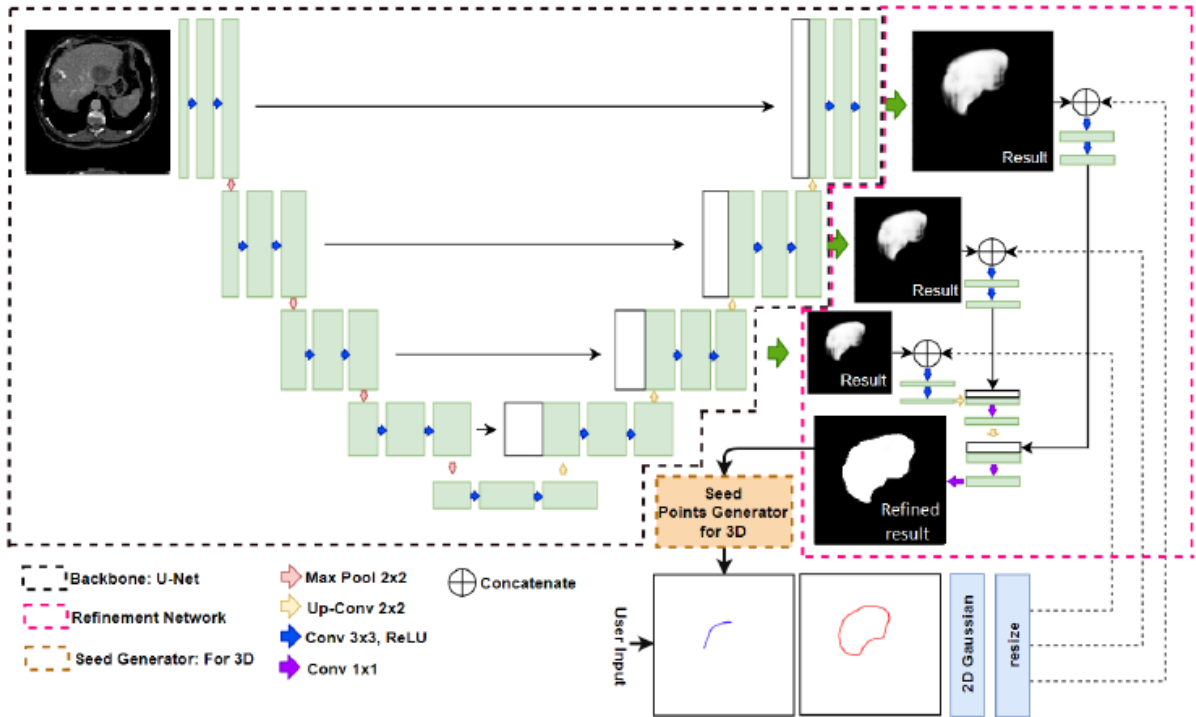


Figure 2.10: The architecture of the network (Source: [Kitrungsakul et al., 2020]).

As previously mentioned, their architecture consists of two parts. The first one consists of the backbone network that produces an initial segmentation. Basically, a **backbone network** extracts features from an input image. The U-Net architecture [Ronneberger et al., 2015] is again used as their backbone network. Concerning the second network, they used the concept of a pyramidal network to refine the initial segmentation.

To validate their approach, the Dice coefficient, the sensitivity, and the positive predicted value (PPV) were used as metric evaluations. Regarding the dataset used, only the 3D Image Reconstruction for Comparison of Algorithm Database ([IRCAD](#)) dataset was employed, which consists of liver segmentations.

2.2.3 Bounding box-based methods

BIFSeg

In this paper *Interactive medical image segmentation using deep learning with image-specific fine tuning* [Wang et al., 2017], the annotator is asked to provide a bounding box around the object of interest. This bounding box is extracted from the image and is fed to the network to produce the segmentation. Then, the annotator reviews this segmentation and provides its correction to the segmentation via scribbles, which is used to update the weight of the neural network to take into account the provided feedbacks. The proposed approach, named BIFSeg, is depicted in Figure 2.11. It is worth mentioning that this approach could indeed also be categorised as a click-based method since it also involves clicks in form of scribbles from the annotator.

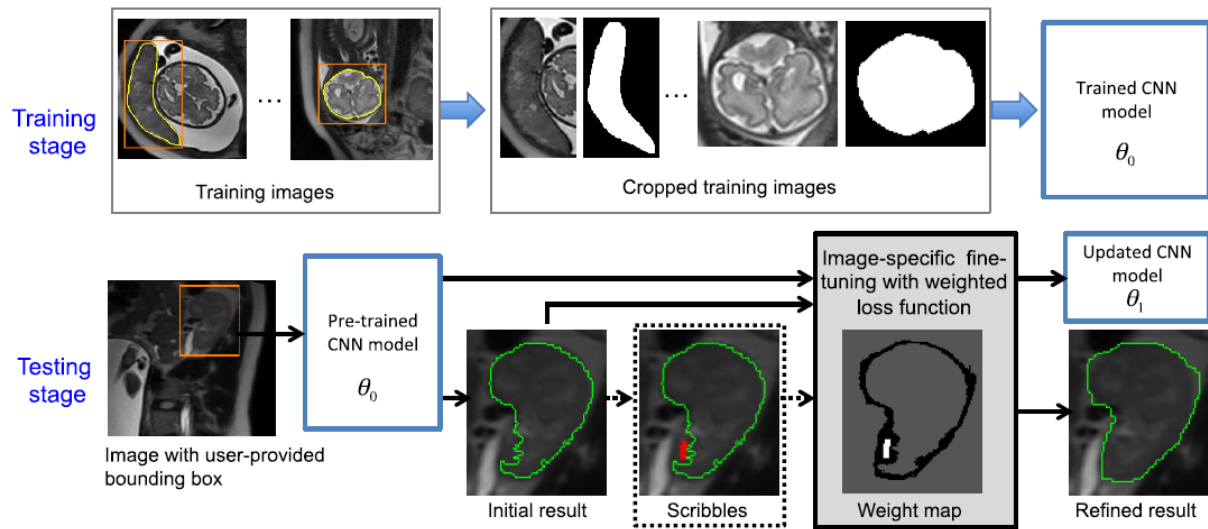


Figure 2.11: Complete pipeline of the framework (Source: [Wang et al., 2017]).

The neural network used in this approach is the P-Net architecture [Wang et al., 2019] presented in section 2.2.1, page 11, and illustrated in Figure 2.7, page 12.

In this paper, the authors focus on two applications, namely 2D segmentation of multiple organs from fetal magnetic resonance slices and 3D segmentation of brain tumour. The dataset, containing multiple organs from fetal MRI, is a private dataset that they have acquired themselves using a Single-shot Fast Spin Echo (SSFSE) method. Stacks of T2-weighted MR images were taken from 18 patients. This dataset was split at the patient level, i.e., the image of 10, 2, and 6 patients were used for the training, validation, and testing sets, respectively. Regarding the second dataset, which is about 3D segmentation of brain tumour, they have used the brain tumour image segmentation challenge of 2015 (BRATS) for their evaluation.

Finally, to validate their approach, only the Dice coefficient was used as the evaluation metric. They have also compared their method against other existing approaches, namely FCN, U-Net [Ronneberger et al., 2015], and HighRes3DNet for 3D segmentation.

2.2.4 Contours-based methods

Click carving

Traditionally, the interactive segmentation takes the inputs of the users as a starting point for the model to segment the region of interests. In this paper *Click Carving: interactive object segmentation in images and videos with point clicks* [Jain and Grauman, 2019], the authors present a novel approach in the interactive segmentation task. Instead of waiting for the users' inputs, a network first generates thousand of segmentation for a given image. Then, the annotator chooses the most accurate segmentation so that the network focuses on these specific segmentations. This process is repeated until the user is satisfied with the segmentation. The three main steps of this approach are described:

1. The first step consists in generating foreground proposals for a given image. The authors use state-of-the-art region proposal generation algorithms to generate 1,000 possible segmentations of the objects of interest, called the foreground regions. The authors have tried several region proposal algorithms. One that generates accurate region proposal that they have used is the multi-scale combinatorial grouping (MCG) algorithm.
2. The second step involves the user feedbacks, in the form of clicks, to rank the different region proposals generated by the MCG algorithm. To avoid the user of scanning through all the 1,000 region proposals, the clicks are used as criteria to rank the regions. More precisely, the user clicks on the contour of the object of interest and the algorithm first finds all the region proposals that intersect with the clicked points and ranked them higher. This process is repeated to have a final ranking and the user is presented the top k proposals having the most points.
3. The third step is an extension of their work where they incorporate negative feedbacks of the user. This step is similar to the previous one. This time, the user is asked to provide negative points, i.e., clicks on background regions. These regions refer to parts of the image that does not contain the objects of interest. The clicked points are then used to re-ranked the region proposals.

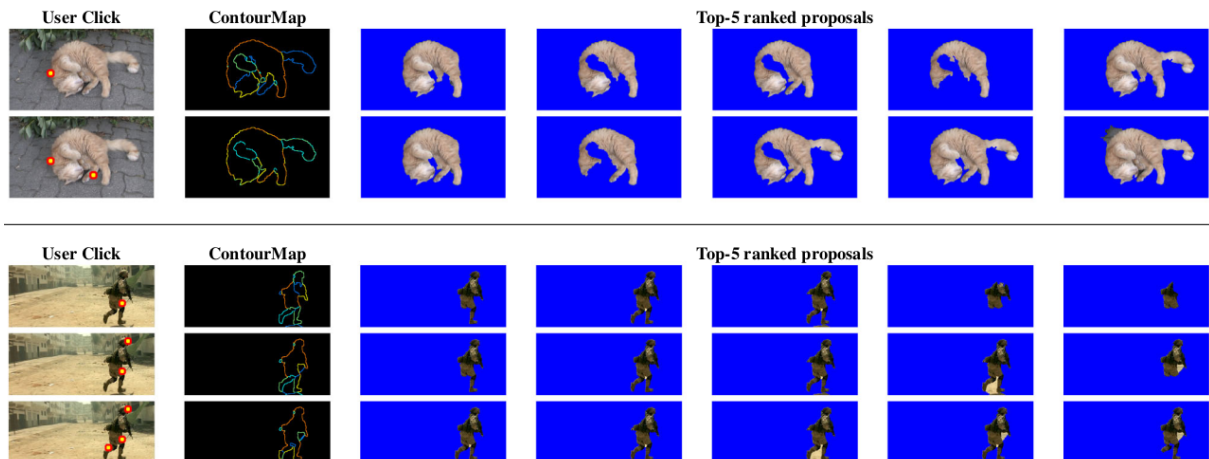


Figure 2.12: Examples using the Click Carving approach (Source: [Jain and Grauman, 2019]).

To evaluate their approach, the authors have used the Intersection over the Union metric. A total of 6 datasets were used, namely SegTrack v2, VSB100, iVideoSeg, MSRC, CMU-Cornell iCoseg, and Interactive Image Segmentation (IIS). These datasets contain numerous types of objects, such as vehicles, people, plants, animals, and many more.

Curve GCN

In this approach, named Curve-GCN, proposed in the paper *Fast interactive object annotation with Curve-GCN* [Ling et al., 2019], the annotator first provides a bounding around the object of interest. Then, a crop based on this bounding box is extracted from the image and an initial segmentation of the object is produced. Habitually, the produced segmentation from the network is given in the form of a binary image, where the foreground, i.e., white pixels represents the object of interest, and the background, i.e., all the other pixels are in black. However, the network in this approach returns a polygon delineating the object of interest. Then, the annotator can move vertices of the polygon to correct the initial segmentation. This approach can also be categorised as a bounding box-based method, however the focus is more placed on the interaction with the vertices of the polygon.

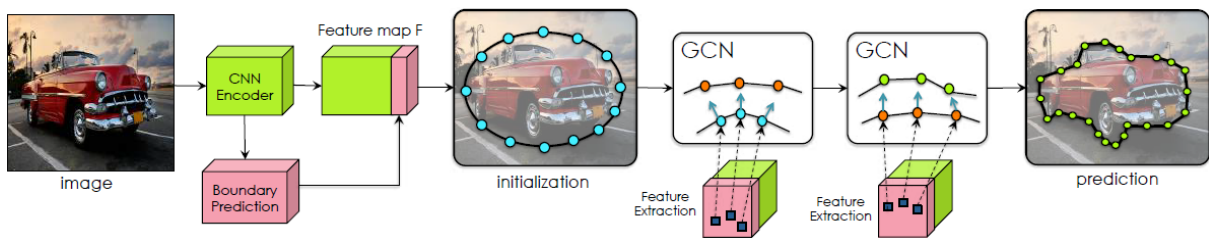


Figure 2.13: Illustration of the initial segmentation produced by Curve-GCN (Source: [Ling et al., 2019]).

The architecture of the neural network is composed of two parts as can be seen in Figure 2.13. The first part is a convolutional neural network, which produces a feature map to create a simple elliptical shape around the object of interest. The second part uses a multi-layer Graph Convolutional Network (GCN), which refines the initial elliptical shape to a more precise polygon delineating the object of interest. The produced polygon segmentation can then be corrected by the user.

To evaluate their approach, the authors have used several datasets, i.e., the Cityscapes dataset as the main benchmark to train and test their model, the KITTI dataset, ADE20K, Aerial Rooftop, Cardiac MR, and ssTEM. These datasets contain various types of objects, such as vehicles, traffic signs, and many more. The Intersection over the Union was used as the metric evaluation to assess the performance of their approach.

Deep snake

The paper *Deep snake for real-time instance segmentation* [Peng et al., 2020] presents a novel approach based on snake algorithms. In short, snake algorithms try to find the contour of an object of interest with the help of additional information, such as the interactions of the user. Their method is divided into two parts. The first one focuses on predicting bounding boxes given an input image. The latter part first creates a diamond contour and tries to deform it into the boundary of the object of interest. An overview of the pipeline is presented in Figure 2.14. The steps of the method are described:

1. A object detector, i.e., a neural network, first produces a bounding box around the object of interest. This bounding box is used to produce a diamond contour around the object. This diamond contour is then fed as input to the deep snake model and it outputs four offsets representing extreme points of the object.
2. The model then takes the diamond contour and the four extreme points and tries to deform the contour until it covers the object of interest. By deforming the diamond contour, a polygon shape is created. This step is repeated until the polygon represents the contour of the object of interest.

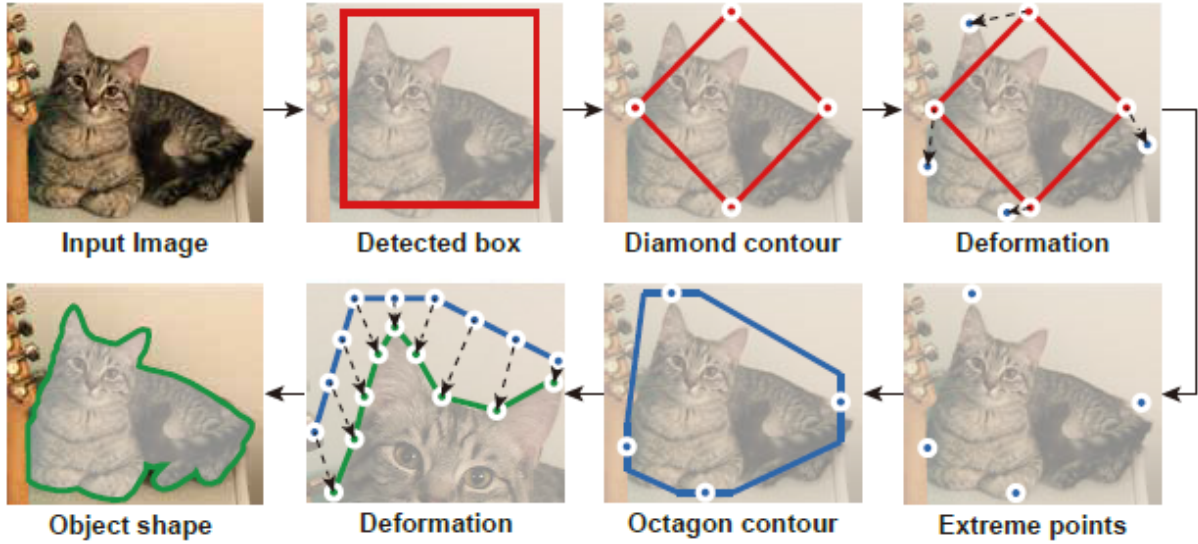


Figure 2.14: Illustration of an segmentation produced by the Deep Snake approach (Source [Peng et al., 2020]).

2.3 Discussion

In this section, a small discussion about the different state-of-the-art methods is done. First, the results stated by the different papers are going to be reviewed and compared against each other. Then, from the comparison, one of the methods is going to be used for conducting several experiments presented in chapter 4.

2.3.1 Results of the approaches

Table 2.1 reports the performance achieved by the different state-of-the-art approaches. Comparable results, such as Click Carving and Curve-GCN, have more or less the same performance. The same observation can be made about DeepIGeos, NuClick, FCNN, RefineNet, and BIFSeg for the performance expressed with the Dice coefficient. Most methods using datasets related to the biomedical sector seem to achieve better performance than methods using other type of datasets, such as a dataset containing vehicles.

Method	Dataset	Type	Metric	Value
HistomicsML	BRCA	BRCA	Accuracy (%)	89.9
	Digital Slide Archive	SKCM		92.4
DiaL	Private dataset	Osteosarcoma	Error rate (%)	20.0
DeepIGeos	Private dataset	2D Placenta	Dice (%)	89.31 ± 5.33
			ASSD (pixels)	1.22 ± 0.55
NuClick	MonuSeg	Nucleus	AJI	0.834
			Dice	0.912
			SQ	0.839
			PQ	0.838
			Hausdorff	4.05
	WBC	Cell	AJI	0.954
			Dice	0.983
			SQ	0.958
			PQ	0.958
			Hausdorff	7.45
	GlaS2015 TestA	Gland	F1	1.000
			Dice _{obj}	0.956
			Hausdorff	15
			F1	1.000
			Dice _{obj}	0.951
	GlaS2015 TestB		Hausdorff	21
FCNN			BCV	Abdomen
	Hausdorff	7.369		
	MAD	0.253		
RefineNet	Slice Liver	Liver	Dice	0.937
			Sensitivity	0.941
			PPV	0.918
BIFSeg	Private dataset	Placenta	Dice (%)	91.93 ± 2.79
		Fetal brain		95.58 ± 1.94
		Fetal lungs		91.71 ± 3.18
		Maternal kidneys		89.37 ± 2.31
Click Carving	Segtrack-v2	Person, vehicles, animals, scenery, and many more	IoU	78.77
	VS100			69.63
	iVideoSeg			79.53
	MSRC Dataset			82.44
	iCoseg dataset			82.13
	IIS dataset			76.47
Curve GCN	Cityscapes	Vehicles	IoU	80.19

Table 2.1: The performance stated by each of the reviewed approaches.

2.3.2 Architecture choice

Regarding the choice made for the architecture used in this thesis, three related criteria and an additional criterion are designed to choose the architecture:

1. Are the datasets used in the paper public?
2. Are the datasets related to the biomedical sector?
3. Can the results be reproduced?

(Bonus) Is the code source available?

The first criterion is to see whether the datasets can be used for replicating the experiments performed in the paper. Then, the second criterion is to select the datasets in the biomedical sector because it shares more similarity with the datasets used in this thesis. Finally, the third criterion is to see whether the implementation of the selected approaches is possible. It is mainly because some approaches use substantial resources that a master student does not necessarily have at hand. In the case where several approaches meet the requirements, one additional constraint is added, i.e., the availability of the source code.

Table 2.2 reports visually whether the requirements are met or not. Two of the approaches meet the requirements, i.e., HistomicsML2 and NuClick. Therefore, an investigation of the two methods was made. The final selection is **NuClick**, because of several reasons. First, during the investigation, it was easier to understand and to test their code, whereas HistomicsML2 had numerous issues in the testing of the available code. In short, the issues encountered by the testing concerned the input images and the resources. First, only images of SVS format, which is used by medical scanners, are accepted. These images are then converted to a pyramidal structure of TIFF format. In our case, the images were already in TIFF format without the pyramidal structure, so the first issue is encountered here. Luckily, a test image was already provided by HistomicsML2, so the first issue was more or less solved for the testing purpose. Then, the next step of the approach was to perform a superpixel segmentation of the TIFF image. The authors stated about 40 minutes for the superpixel segmentation, whereas our time was about 4 hours for this step. Finally, the next step, features extraction, was not feasible because the program exited with the Out of Memory Exception. The conclusion is that this approach is too resource hungry and too long for a real-time use case.

Then, comparing approximately the performance of both methods, NuClick seems to achieve an F1 score of 1.000 with the glands meaning that it has managed to produce a very accurate segmentation, whereas HistomicsML2 only achieves 92.4 of accuracy. The results of NuClick are going to be further discussed when replicating their work in section 4.2, page 42.

Method	Criterion 1	Criterion 2	Criterion 3	Bonus
HistomicsML2	✓	✓	✓	✓
Dial	✗	✓	✗	✗
DeepIGeos	✗	✓	✗	✗
NuClick	✓	✓	✓	✓
FCNN	✓	✓	✓	✗
RefineNet	✓	✓	✓	✗
BIFSeg	✗	✓	✗	✗
Click Carving	✓	✗	✓	✗
Curve-GCN	✓	✗	✓	✗

Table 2.2: Selection of the architecture based on the criteria.

Chapter 3

Methodology

This chapter presents the methodology developed to conduct the experiments. It draws heavily on the methodology presented with the NuClick architecture [Alemi Koohbanani et al., 2020]. First, section 3.1 shows an overview to get a general idea of the complete annotation process. Then, section 3.2 explains the acquisition of the dataset from the Cytomine web user interface. In section 3.3, the neural network, used in this thesis, is going to be explained along with its specificities. After that, section 3.4 presents the various metrics used in the assessment of the performance. This chapter ends with section 3.5, where the implementation details are described.

3.1 Overview

A simplified overview of the methodology is presented in Figure 3.1. Illustrations of the dataset generation, the training procedure, and testing procedure are depicted in Figure 3.2, Figure 3.3, and Figure 3.4, respectively.

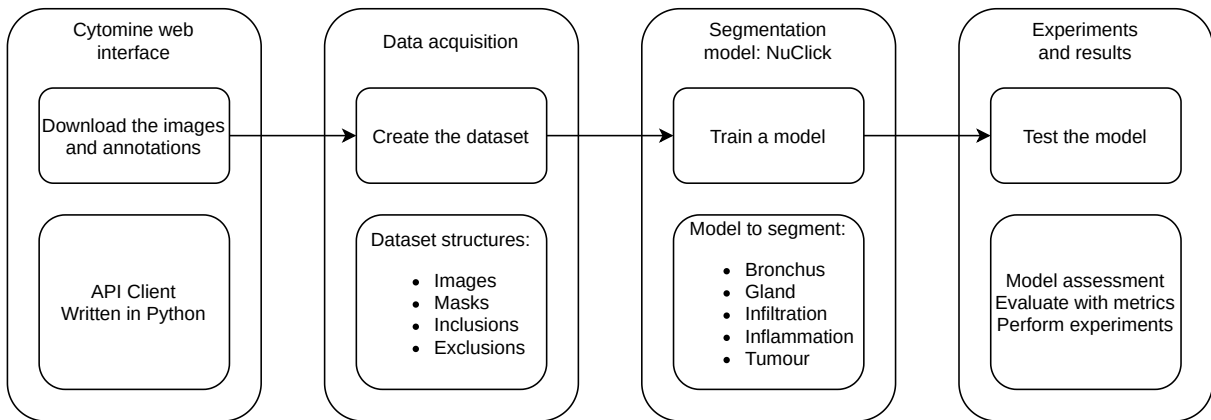


Figure 3.1: Simplified overview of the methodology.

3.2 Data acquisition

This section aims at describing the process of the data acquisition. There are three main phases in the data acquisition, namely the acquisition, the processing, and the splitting phase, as illustrated in Figure 3.1. First, the dataset structure is explained followed by the three aforementioned phases.

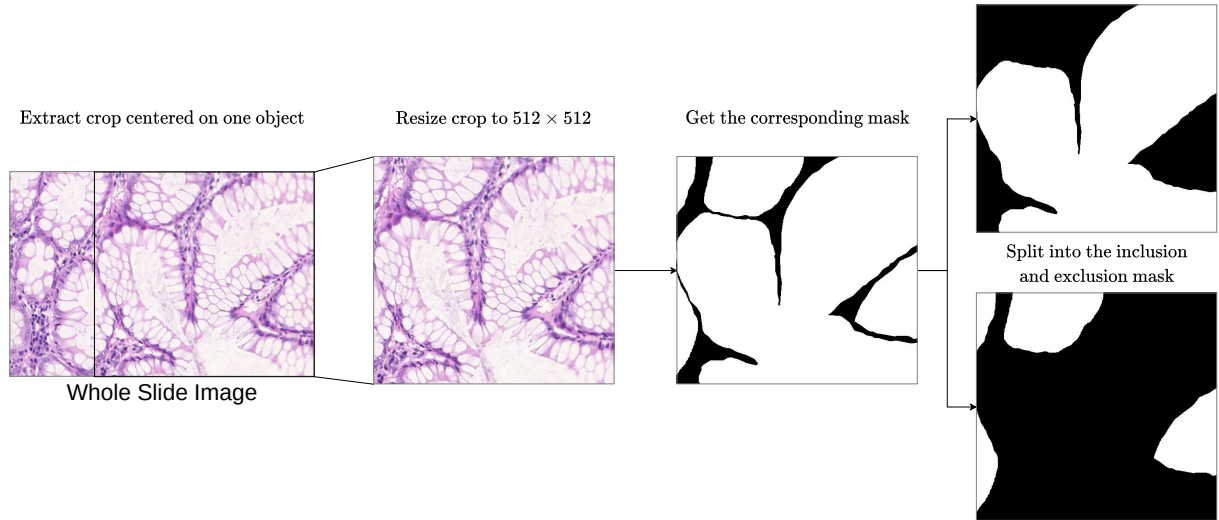


Figure 3.2: Illustration of the creation of the dataset from the raw downloaded Cytomine data. This process is done for each whole slide image in the dataset.

3.2.1 Dataset structure

A traditional segmentation dataset is composed of the images and the binary mask of objects present in these images. However, in this thesis, a dataset is composed of the aforementioned images and masks with two additional components called inclusion and exclusion masks. In short, the inclusion is the binary mask of only one object in the images. In the case where there is more than one binary mask in the image, the exclusion is all the other binary masks except the one in the inclusion. In Figure 3.2, the two images on the right represent, on the upper part the inclusion mask and on the lower part the exclusion mask, respectively. The purpose of these two supplementary components is going to be explained in subsection 3.3.1.

3.2.2 Acquisition

The datasets are acquired from Cytomine [Marée et al., 2016] using its Python client, which is available on [GitHub](#). The generation of the dataset involves the following steps:

1. **Annotation term selection:** generally, a whole slide image contains annotations of several different terms, e.g., bronchus, tumour, etc. The first step of the acquisition is to select a specific term.
2. **Users selection:** after the term selection, the annotations from specific users are selected. The main motivation is because some users have tried out a Cytomine feature that allows the user to annotate the image. This results in a false annotation that might be downloaded in the dataset.
3. **Dataset download:** following the previous steps, for each annotation, a crop centred on the annotation in the whole slide image is extracted using its coordinates. Likewise, the corresponding binary mask is extracted as well. After the extraction, the inclusion and exclusion binary masks are created based on the binary mask crop. If there is only the mask of the desired annotation, the exclusion mask is black. The size of the crop is 512×512 if the binary mask is smaller. Otherwise, the dimensions are of the size of the binary mask.

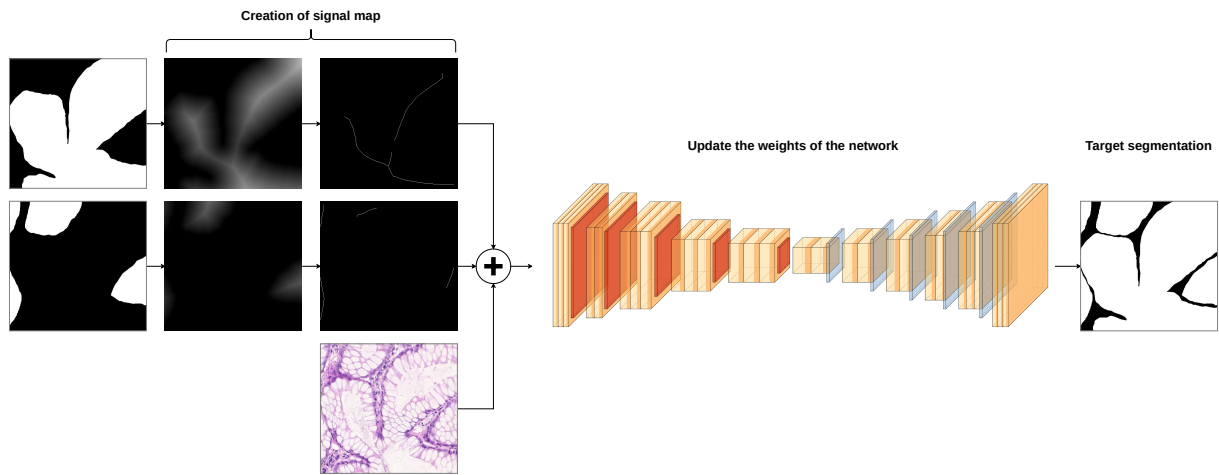


Figure 3.3: Illustration of the training procedure.

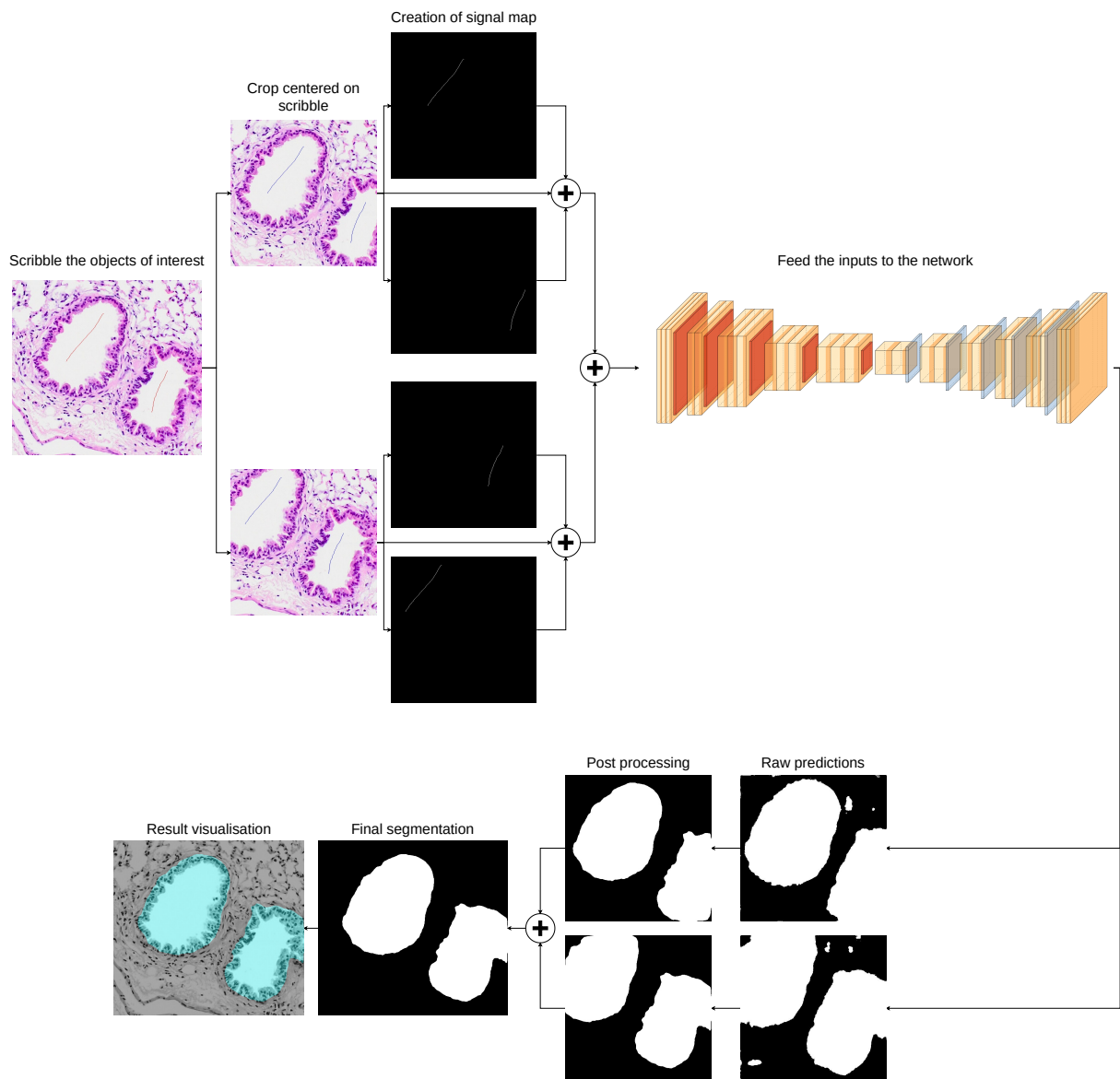


Figure 3.4: Illustration of the testing procedure.

3.2.3 Data processing

After the acquisition phase, some irregular annotations can occur in the downloaded dataset. An example of irregular annotation is the annotation of only one pixel. Three steps of examination are done to remove these undesirable annotations:

1. **Empty annotation:** occasionally, an annotation that has no binary mask, i.e., the crop containing binary mask is completely black, can occur. This first step aims at detecting this kind of annotations and removing them.
2. **White annotation:** opposite to the first step, a crop centred on the binary mask can be composed of only white pixels, i.e., the whole image is a binary mask. This type of annotation can occur in the case where another bigger binary mask is superimposed. The second step focuses on removing them.
3. **Empty inclusion mask:** similar to the first step, the desired annotation is present on the binary mask but not in the inclusion mask. This step replaces the empty inclusion mask with the binary mask.

An optional step that can be performed is the manual removal of images done by the users.

3.2.4 Data splitting

Since the dataset is composed of crops centred on the annotations and not the whole slide images themselves, splitting the dataset is done with the attention that crops from the same whole slide image are split together. The main motivation behind this strict constraint is to avoid bias when evaluating the model on the testing set since the model could incorporate relevant information from the whole slide images used during the training. Two kind of splits are done, namely the so-called *three-way data splits* and the *k-fold cross-validation*. After the split is done, a CSV file is created containing information of the split similar to the CSV file generated by Cytomine. More specifically, it contains the ID of the annotation, the ID and the file name of the whole slide image, the term of the annotation, the user that created this annotation, and in which split the annotation is located.

Three-way data splits

The three-way data splits is more commonly known as the training, validation, and test sets:

1. **Training set:** this set is used for the learning phase, i.e., to adapt the parameters of the model for a given task. In this case, it is to segment a specific object.
2. **Validation set:** this set is used to fine-tune the parameters of the model. This set also prevents underfitting and overfitting.
3. **Testing sets:** this set is used for the performance assessment of the final model.

Taking into account the previously described constraint, about 80% is used for the training set and 20% for the testing set. From these 80% of the training set, about 20% is used for the validation set. For the conducted experiments, more precise values are going to be stated.

k-fold cross-validation

This split is conventionally used for the fine-tuning of the model's hyperparameters. About 80% is used for the training set and the remaining 20% for the test set. The training set is then divided evenly into k folds.

3.3 Segmentation model: NuClick

In this thesis, the model that is going to be used is called NuClick [Alemi Koohbanani et al., 2020], presented in section 2.2.2, page 13. The motivation behind the use of this architecture is because of the minimal interactions needed from the user for annotating the desired object. Previously stated, NuClick is based on a well-known architecture, which is U-Net [Ronneberger et al., 2015]. In short for NuClick, the user provides simple line strokes on the object of interest. These line strokes are used as supervisory signals along the image to guide the network towards segmenting the object of interest as illustrated in Figure 3.4.

The complete architecture of the neural network is presented in Figure 3.5. It is composed of three building blocks, namely *convolutional block*, *residual block*, and *multi-scale convolutional block*. The main building block of this architecture is the convolutional block. As shown in the legend of Figure 3.5, this block is composed of a convolutional layer followed by a batch normalisation layer and ends with a ReLU activation function. Then, a residual block is composed of two convolutional blocks. Finally, a multi-scale convolutional block allows to segment various objects, smalls or larges, that vary in scales [Alemi Koohbanani et al., 2020]. This block is composed of four convolutional blocks, where each of the blocks has different parameters for the kernel size and the dilation rate as shown in Figure 3.5. Let F be the size of the input feature map of this block, the input size of the four convolutional blocks are $F/4$, which produces four feature maps that are concatenated back together to constitute the output feature map of size F . The number of parameters of the model is given in Appendix A, page 81.

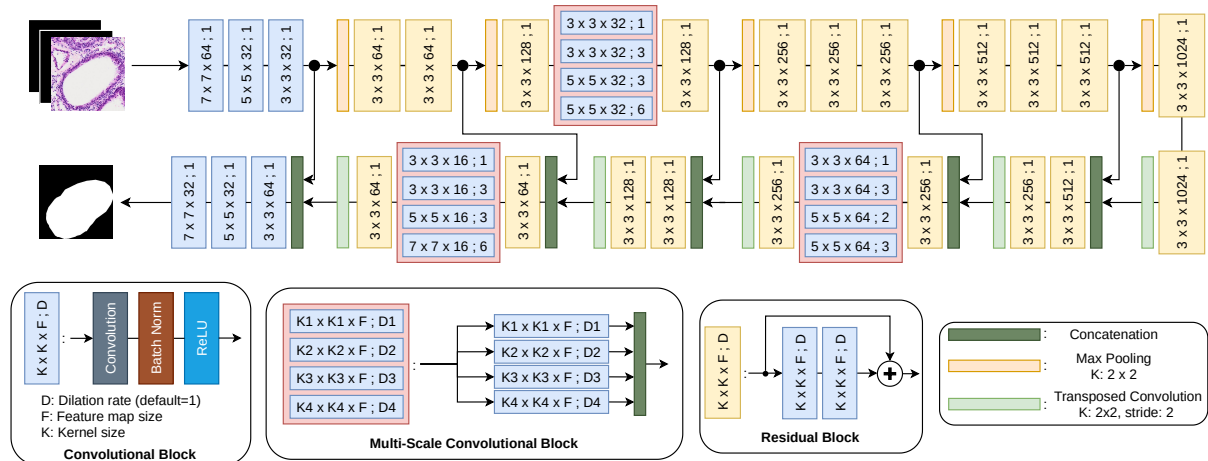


Figure 3.5: NuClick neural network architecture.

Since complex objects, such as bronchi, glands, etc, are usually large, the input shape of the network is $512 \times 512 \times 5$. The first two dimensions are, respectively, the height and the width of the input image and the last dimension is the number of channels. The channels are composed of the concatenation of the RGB channels with the so-called inclusion and exclusion mask that is going to be described in the subsequent section. The output shape of the network is 512×512 where the two dimensions represent the height and the width of the binary segmentation mask, respectively. The final layer uses a sigmoid activation function, which makes the values of the predicted mask range from 0 to 1.

3.3.1 Inclusion and exclusion map

As briefly explained previously, the number of input channels for NuClick is five, i.e., the concatenation of the RGB image with two additional channels called inclusion and exclusion map. These two maps incorporate the line strokes done by users to guide the network towards segmenting these regions of interest. The inclusion map contains one specific line stroke. In the case

the annotator scribbles several line strokes, the exclusion map contains all the other line strokes except the one that is present in the inclusion map. The purpose of having these two maps is to inform the network that other close objects are present and that the network should not segment nearby objects as a single mask. These maps are more generally called as signal map.

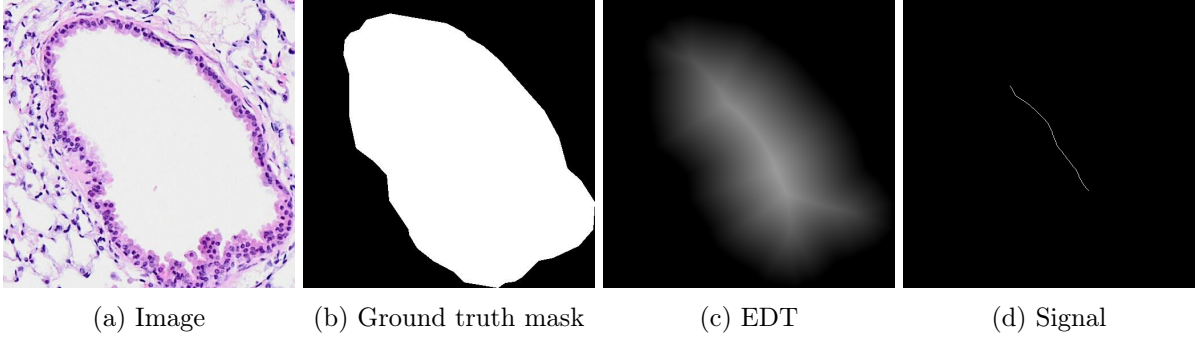


Figure 3.6: Creation process of the signal map.

Training phase

To train such networks, having to manually scribble line strokes on objects of interest for each image at each epoch is infeasible. An algorithm from the original paper based on the ground truth binary mask is used to mimic the line strokes done by the annotators. The process of creating these line strokes automatically, also called signals, is shown in Figure 3.6. This process is done in three steps:

1. The first step starts with the ground truth binary mask \mathcal{M} . An Euclidean distance transform (EDT) $D(x)$ is applied on the mask as shown in Figure 3.6c. It produces a distance map where each pixel is represented by the distance of this pixel with the closest pixel of the object boundaries.

$$D_{i,j}(\mathcal{M}) = \{\sqrt{(i - i_b)^2 + (j - j_b)^2} \mid (i, j) \in \mathcal{M}\} \quad (3.1)$$

where (i_b, j_b) represents the coordinate of the closest pixel to the boundaries with the pixel at position (i, j) .

2. The second step is to apply a threshold on the produced distance map \mathcal{D} . To compute the threshold τ , the mean μ and the standard deviation σ of the distance map is calculated. The threshold is then sampled uniformly at random in the interval $[0, \mu + \sigma]$:

$$\mathcal{D}_{i,j} = \begin{cases} 1 & \text{if } D_{i,j}(\mathcal{M}) > \tau \\ 0 & \text{otherwise.} \end{cases} \quad (3.2)$$

The purpose of the random threshold and not a fixed one is to allow the network to learn various input signals with the same annotation for robustness since it is intended to mimic the user's annotations. Signals with different values of τ are depicted in Figure 3.7.

3. The last step is to produce the inclusion map by computing the morphological skeleton of the distance map \mathcal{D} as illustrated in Figure 3.6d.

Regarding the creation of the inclusion and exclusion maps, the first step is to generate the signal from the ground truth mask as explained previously as shown in Figure 3.8b. For the example presented in Figure 3.8, there is a total of 3 inclusion and exclusion maps. The second step is to split the desired signal to the inclusion map (Figure 3.8c) and subtract this signal from the signal map to obtain the exclusion map (Figure 3.8d). This process is done for each of the signal present in the signal map. In the case where there is only one complex object such as in Figure 3.6a, the exclusion map is completely black.

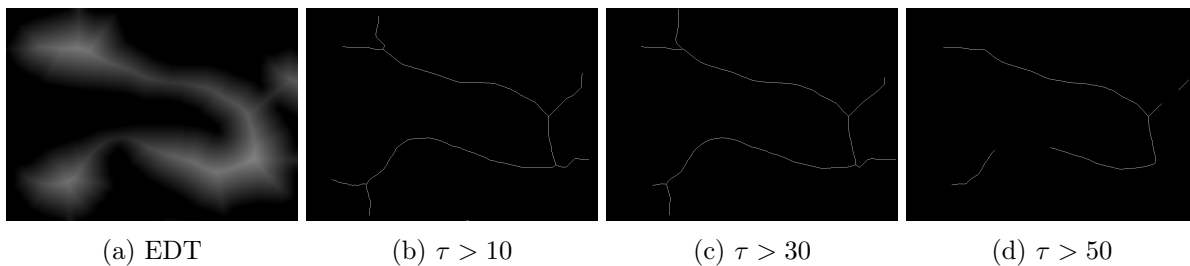


Figure 3.7: Signal map with different threshold value τ .

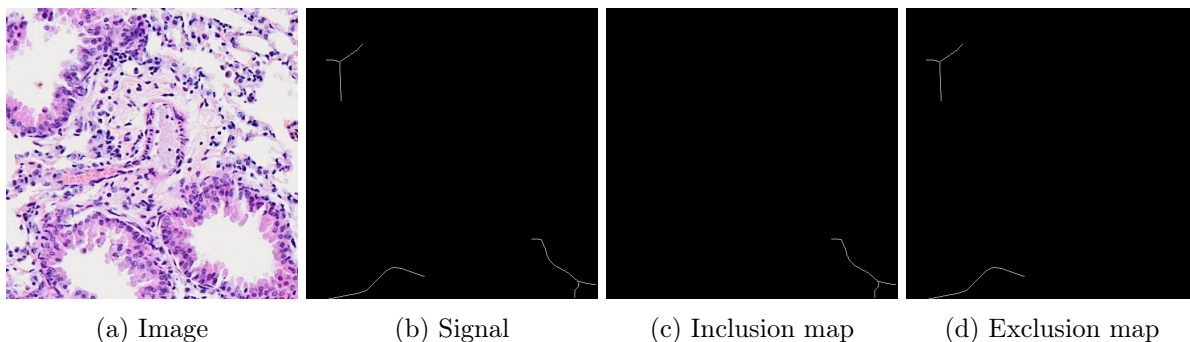


Figure 3.8: Inclusion and exclusion map.

Testing phase

In the testing phase, the scribbles of the user are stored. These scribbles are later used to build the inclusion and exclusions maps. The first step is to scribble a line stroke on the objects of interest. An example of scribbles on these objects done by the user is shown in green in Figure 3.9. From these scribbles, crops of size $512 \times 512 \times 3$ are extracted from the image, each of the crops is centred on one of the scribbles. In the example, there is a total of 5 crops. The inclusion map is created, which is the centred scribble. Similarly, the exclusion map is constructed from the other scribbles if they appear in the crop. After the creation of the maps, the concatenation of the crop with the inclusion and exclusion map produces the input of the network. Lastly, it is fed to the network to produce the segmentation of the scribbled objects.

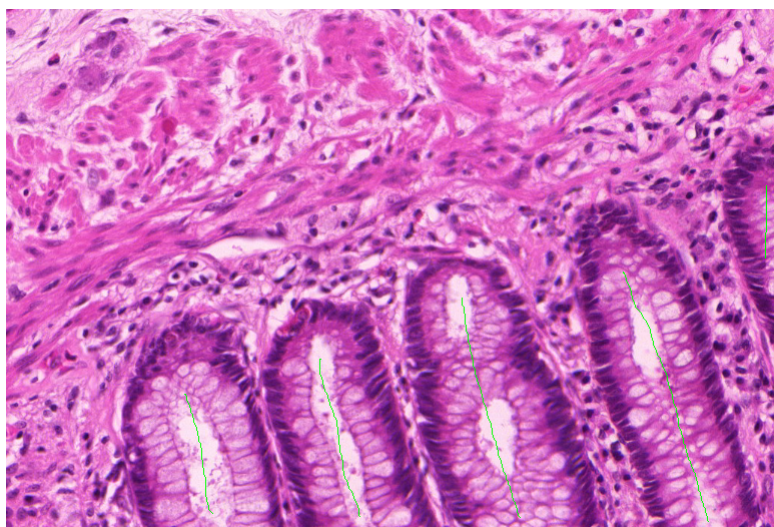


Figure 3.9: Scribble on the objects of interest, i.e., glands, done by a user.

3.3.2 Loss function

In this thesis, the loss function used to train the neural network is a combination of the soft dice loss and a weighted cross-entropy loss [Alemi Koohbanani et al., 2020]:

$$\mathcal{L} = 1 - \underbrace{\left(2 \sum_{i=1}^n p_i g_i + \varepsilon\right) / \left(\sum_{i=1}^n p_i + \sum_{i=1}^n g_i + \varepsilon\right)}_{\text{Soft dice loss}} - \underbrace{\frac{1}{n} \sum_{i=1}^n w_i (g_i \log p_i + (1 - g_i) \log(1 - p_i))}_{\text{Weighted cross entropy}} \quad (3.3)$$

where n is the total number of pixels in the image, p_i and g_i are, respectively, the value of the prediction and the ground truth for the i^{th} pixel in the image, w_i is the weight associated to pixel i , and ε is a small number to avoid numerical instabilities.

Soft dice loss

The dice loss has shown to control the class imbalance. Let P be the predicted mask and G the ground truth mask, the *soft dice loss* is computed as follows

$$\mathcal{L}_{dice}(P, G) = 1 - Dice(P, G) \quad (3.4)$$

where $Dice(P, G)$ is the *Sørensen-Dice coefficient* of P and G . This coefficient can be expressed as

$$Dice(P, G) = \frac{2|P \cap G|}{|P| + |G|} \quad (3.5)$$

where $|P|$ and $|G|$ are the number of elements in P and G , respectively, and $P \cap G$ is the intersection of the elements in P and G . In the context of segmentation, $|P|$ represents the sum of the non zero value in P and $P \cap G$ is the element-wise product of P and G . Equation 3.4 can therefore be reformulated as

$$\mathcal{L}_{dice}(P, G) = 1 - \left(2 \sum_{i=1}^n p_i g_i + \varepsilon\right) / \left(\sum_{i=1}^n p_i + \sum_{i=1}^n g_i + \varepsilon\right) \quad (3.6)$$

where n is the total number of pixels in the image, p_i is the predicted value for pixel i , g_i is the ground truth value for pixel i and ε is a term to avoid numerical instability of the division by zero, i.e., P and G are empty. By default, the value of ε is 1.

Weighted cross entropy

The *weighted cross entropy* is computed as follows

$$\mathcal{L}(p, g) = -w(g \log(p) + (1 - g) \log(1 - p)) \quad (3.7)$$

where p and g are respectively the predicted value and the ground truth value for a pixel and w the weight associated to this pixel. Similar to the soft dice loss, the generalisation to all pixels is defined as follows

$$\mathcal{L}_{wce}(P, G) = -\frac{1}{n} \sum_{i=1}^n w_i (g_i \log(p_i) + (1 - g_i) \log(1 - p_i)) \quad (3.8)$$

where n is the total number of pixels in the image, p_i and g_i are respectively the value of the prediction and ground truth for the i^{th} pixel in the image, w_i is the weight associated to pixel i . The weight \mathcal{W} is an adaptive weight map taken from [Alemi Koohbanani et al., 2020]. It means that for each pair of inputs P and G , the weight is based on the ground truth mask and is computed as follows

$$\mathcal{W} = \alpha^2 \mathbf{G} + \alpha \tilde{\mathbf{G}} + 1 \quad (3.9)$$

where α is an adaptive factor, \mathbf{G} is the ground truth mask, and $\tilde{\mathbf{G}}$ is its complement. The factor α is based on the ground truth mask and its complement and is computed as follows

$$\alpha = \max \left\{ \sum \tilde{\mathbf{G}} / \sum \mathbf{G}, 1 \right\} \quad (3.10)$$

3.3.3 Post-processing

The predicted mask of the neural network may include noises. The purpose of the post-processing is to further refine the initial predicted segmentation mask by removing as much noises as possible. The following steps are made for the post processing:

1. **Threshold the predicted values:** since the network predicts a mask where the values range from 0 to 1, a threshold is put to remove the false predicted pixels.

$$P_{i,j} = \begin{cases} 1 & \text{if } P_{i,j} > \xi \\ 0 & \text{otherwise.} \end{cases} \quad (3.11)$$

where $P_{i,j}$ is the value of the predicted pixel at position (i, j) in the image and ξ the threshold, which is set at 0.5.

2. **Small objects removal:** as mentioned previously, some noises can be predicted by error as illustrated by the green circles in Figure 3.10a. A minimal size in pixels is set so that masks that are below this specified size are removed. The result is shown in Figure 3.10b, it can be seen that the noises is removed. The threshold was applied beforehand, which has polished the contour of the masks and that can be clearly seen on the bottom left mask. The minimal size in this thesis is set at 100 pixels. Thus, masks under this threshold are removed.
3. **Holes filling:** another issue that can occur is the fact that the network predictions leave small holes as depicted in the rounded green rectangle in Figure 3.10c. A threshold is put on the area of the hole to be filled. It is important because some shapes of objects can incorporate holes in their binary mask. The resulting process can be seen in Figure 3.10d. The area of the hole in this thesis is set at 300.

The values for the threshold, the minimum size, and the area of the holes are going to be discussed in section 4.7, page 71. The aforementioned steps are performed for each predicted mask produced by the network. The final step is to merge all these cropped masks by overlapping them to form the final segmentation mask as shown in Figure 3.4, page 24.

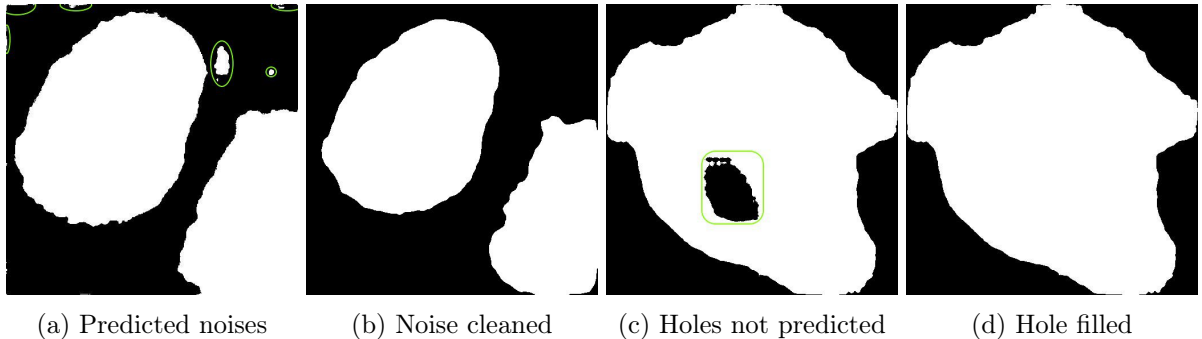


Figure 3.10: Example of the post-processing for an noisy prediction and a prediction with a hole.

3.4 Evaluation metrics

In this section, various metrics are going to be explained and used for the subsequent experiments for the performance assessment. The intersection over the union, the dice coefficient, and the Hausdorff distance are commonly used for segmentation task assessments.

3.4.1 Intersection over the union

The *intersection over the union* (IoU), also known as the *Jaccard index*, measures the similarity between two sets, as illustrated in Figure 3.11. It is defined as the intersection divided by the union of the two sets:

$$IoU(P, G) = \frac{|P \cap G|}{|P \cup G|} = \frac{|P \cap G|}{|P| + |G| - |P \cap G|} \quad (3.12)$$

In the context of the segmentation, let P be the predicted segmentation mask and G the ground truth mask, it can be computed as follows

$$IoU(P, G) = \frac{|P \cap G| + \varepsilon}{|P| + |G| - |P \cap G| + \varepsilon} \quad (3.13)$$

where $|\cdot|$ denotes the sum of the elements, \cap denotes the Hadamard product (element-wise product), and ε is equal to 1 to avoid division by zero. Therefore, it can be expressed as

$$IoU(P, G) = \left(\sum_{i=1}^n p_i g_i + \varepsilon \right) / \left(\sum_{i=1}^n p_i + \sum_{i=1}^n g_i - \sum_{i=1}^n p_i g_i + \varepsilon \right) \quad (3.14)$$

where p_i and g_i are, respectively, the value of the predicted pixel and the ground truth pixel at the i^{th} pixel.

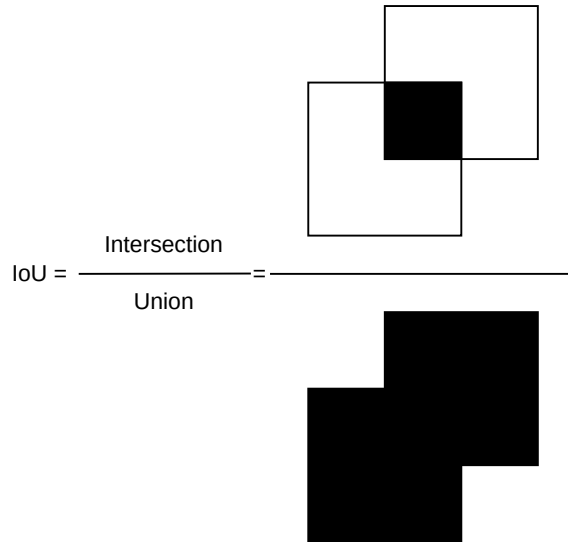


Figure 3.11: The intersection over the union (IoU).

The values of this metrics range from 0 to 1, where 0 means that there is no overlap between the predicted segmentation and its ground truth, and 1 means a perfect prediction of the ground truth mask. Thus, the higher this value, the better the segmentation is. The range of values for the Dice coefficient is exactly the same.

3.4.2 Dice coefficient

The *Sørensen-Dice coefficient*, also known as *Dice coefficient* or *F1 score*, measures the similarity between two sets, as shown in Figure 3.12. It is pretty similar to the intersection over the union. Let P be the predicted segmentation mask and G the ground truth mask,

$$Dice(P, G) = \frac{2|P \cap G|}{|P| + |G|} \quad (3.15)$$

Again, in the context of segmentation,

$$Dice(P, G) = \frac{2|P \cap G| + \varepsilon}{|P| + |G| + \varepsilon} \quad (3.16)$$

where $|\cdot|$ denotes the sum of the elements, \cap denotes the Hadamard product, and ε is a small number to avoid division by zero. It can be re-expressed as

$$Dice(P, G) = \left(2 \sum_{i=1}^n p_i g_i + \varepsilon \right) / \left(\sum_{i=1}^n p_i + \sum_{i=1}^n g_i + \varepsilon \right) \quad (3.17)$$

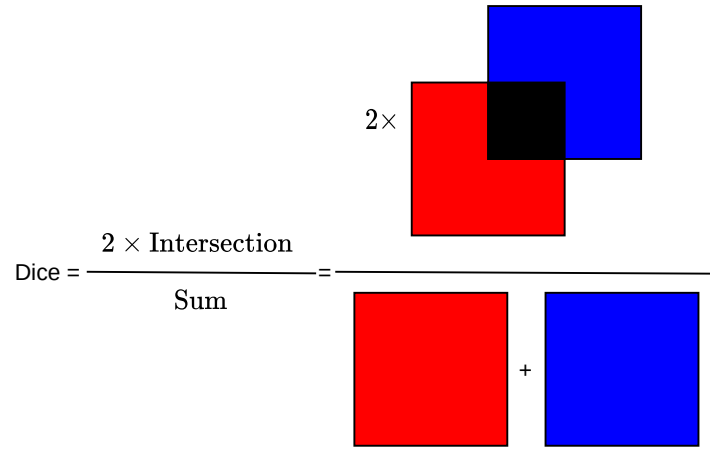


Figure 3.12: Dice coefficient.

IoU and Dice relationship

A relationship exists between the intersection over the union and the Dice coefficient. Let $|P \cap G| = a$ and $|P| + |G| = b$, the Dice coefficient and the IoU can be expressed as

$$Dice = 2\frac{a}{b}, \quad IoU = \frac{a}{b-a} \quad (3.18)$$

The relationship can be derived as follows

$$Dice = 2\frac{a}{b} = \frac{2\frac{a}{b-a}}{\frac{b}{b-a}} \quad (3.19)$$

$$= \frac{2IoU}{\frac{b+a-a}{b-a}} = \frac{2IoU}{\frac{a}{b-a} + \frac{b-a}{b-a}} \quad (3.20)$$

$$Dice(P, G) = \frac{2 IoU(P, G)}{IoU(P, G) + 1} \quad (3.21)$$

Therefore, only one of the two metrics is going to be used for the subsequent experiments, which is the intersection over the union, since the Dice coefficient is just a similar representation of the intersection over the union in the numerical sense.

3.4.3 Hausdorff distance

The Hausdorff distance [Karimi and Salcudean, 2019] measures how far two sets are from each other based on their boundary, as depicted in Figure 3.13. This metric measures the boundary-based accuracy between the segmented object and the ground truth mask object. Intuitively, the Hausdorff distance measures the longest distance between the segmented object and its ground truth at their boundaries. Thus, the smaller the gap, the higher the similarity between the two boundaries. To measure the longest distance from a segmented object X with its ground truth Y , the one-sided distance is computed as follows

$$hd(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\|_2 \quad (3.22)$$

and with the opposite side,

$$hd(Y, X) = \max_{y \in Y} \min_{x \in X} \|x - y\|_2 \quad (3.23)$$

The Hausdorff distance is defined as the maximum of the bidirectional distance:

$$HD(X, Y) = \max(hd(X, Y), hd(Y, X)) \quad (3.24)$$

This metric is expressed in term of pixels, where 0 means that the segmented object X perfectly matches the boundaries of the ground truth Y and higher values represent the longest distance. Thus, the lower the value, the better the segmentation is.

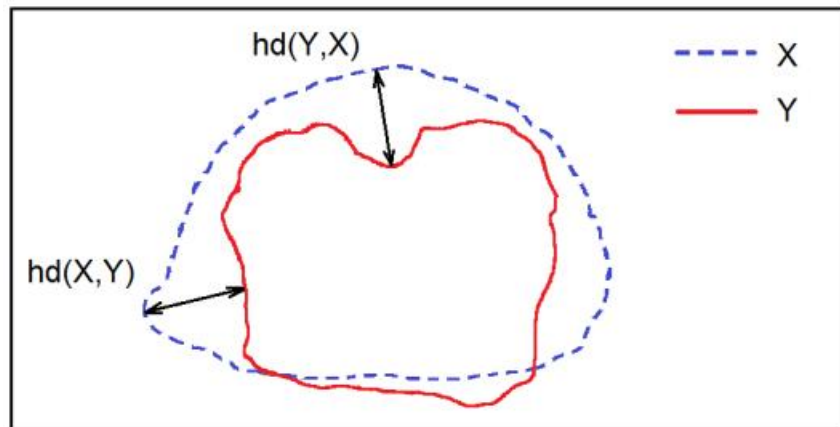


Figure 3.13: Hausdorff distance, where the dashed line is the predicted mask and the full line is the ground truth mask (Source: [Karimi and Salcudean, 2019]).

3.5 Implementation details

The code is available on Github at <https://github.com/bathienle/master-thesis-code.git>. It is inspired by the original implementation of NuClick. However, no code is reused from their implementation. The code developed in this thesis is done in the Python language. It heavily relies on the PyTorch framework and the NumPy library. For the data acquisition described in section 3.2, the Shapely, PIL, and Cytomine libraries are used. The documentation of the Cytomine python client is available on their website Cytomine ULiège R&D Documentation. Regarding the generation of the inclusion and exclusion map explained in subsection 3.3.1, to compute the Euclidean distance transform, the SciPy library is used. For the post-processing, the scikit-image is used for the morphological skeleton, the holes filling, and the removal of small objects. For some of the subsequent experiments, the OpenCV library is used. About 2,000 lines of codes were needed for this thesis. This number does not take into account the comment nor the blank lines.

Table 3.2 resumes the value used for the parameters in this thesis. The training of the models was done exclusively on the GPUs provided by the [Alan GPU cluster](#) at the University of Liège. The list of GPUs used during the thesis is shown in Table 3.1.

GPU	GTX 1080 Ti	RTX 2080 Ti	Quadro RTX 6000	Tesla V100
-----	-------------	-------------	-----------------	------------

Table 3.1: The list of GPUs from the Alan cluster used during the thesis.

Parameter	Value
Epochs	300
Batch size	16
Optimiser	Adam
Learning rate	3×10^{-3}
Weight decay	5×10^{-5}

Table 3.2: Summary of hyperparameters and other parameters used in this thesis.

A very small cross-validation search was done to find the optimiser and the values for the learning rate and the weight decay. The setup was to train a model to segment bronchus from the ULG-LBTD-NEO04 dataset¹. The tested optimiser and parameters are shown in Table 3.3. Each of the models was trained on 50 epochs and evaluated with the intersection over the union on a small test set. The best parameters for this small search was the Adam optimiser with a learning rate of 1×10^{-3} and a weight decay of 1×10^{-5} . From this small cross-validation search, the aforementioned parameters were used to train a model on the same dataset but with various epochs size, i.e, 100, 200, and 300 epochs. The results show that 300 epochs give slightly better performance than the 100 and 200 epochs. The same setting was used for determining the batch size, resulting in a batch size of 16.

Optimiser	Adam	SGD	✗	✗
Learning rate	1×10^{-2}	1×10^{-3}	1×10^{-4}	1×10^{-5}
Weight decay	1×10^{-2}	1×10^{-3}	1×10^{-4}	1×10^{-5}

Table 3.3: Hyperparameters tested in a cross-validation search.

A complete cross-validation search would take a tremendous amount of time. Therefore, for all the subsequent experiments, Table 3.2 was used for the training of the models. However due to a lack of time, this is the principal reason why specific fine-tuning for all the type objects used in this thesis is not done.

¹The dataset is going to be presented in the next chapter in more details.

Chapter 4

Experiments and results

In this chapter, various experiments are conducted and their results discussed. First, section 4.1 presents the different datasets used for the experiments. Then, section 4.2 tries to replicate the performance obtained in the original paper of NuClick. After that, section 4.3 presents the protocol used for all the experiments. Next, section 4.4 introduces the two major experiments that are related to the annotations. After that, a small robustness experiment is performed in section 4.5. Afterwards, experiments related to the model architecture in section 4.6 are performed. Lastly, this chapter ends with a discussion over the conducted experiments in section 4.7.

4.1 Datasets

In this section, the different datasets used for the subsequent experiments are described. First, a summary of the datasets and the type of objects used is shown in Table 4.1. The complex objects that the model has to segment are bronchi, inflammations, glands, infiltrations, and tumours. An overview of the size of the whole slide images in each of these datasets is reported in Table 4.2, which presents the size of the smallest to the biggest whole slide image and the level of magnification for these images.

Dataset	Bronchus	Inflammation	Gland	Infiltration	Tumour
CHALLENGE-CAMELYON16	X	X	X	X	2,545
CHALLENGE-GLAS-2015	X	X	1,538	X	X
CHU-ANAPATH-NST-DL	X	354	6,268	2,833	X
ULG-LBTD-NEO04	379	148	X	X	492
ULG-LBTD-NEO13 (3)	409	X	X	X	175

Table 4.1: Summary table for the types and datasets used. It shows the number of annotations for each type of object.

Dataset	Smallest	Largest	Magnification
CHALLENGE-CAMELYON16	$35,840 \times 45,056$	$111,104 \times 217,088$	$1\times$
CHALLENGE-GLAS-2015	430×567	522×775	$20\times$
CHU-ANAPATH-NST-DL	$18,416 \times 18,336$	$104,848 \times 52,848$	$20\times$
ULG-LBTD-NEO04	$14,848 \times 17,920$	$75,776 \times 37,888$	$10\times$
ULG-LBTD-NEO13 (3)	$19,456 \times 11,776$	$87,552 \times 38,400$	$10\times$

Table 4.2: The smallest to the largest dimension of the whole slide images in each dataset.

4.1.1 CHALLENGE-CAMELYON16-TRAIN

The CHALLENGE-CAMELYON16-TRAIN dataset comes from the *Cancer Metastases in Lymph Nodes Challenge 2016* (CAMELYON16) [Ehteshami Bejnordi et al., 2017]. It is composed of whole slide images containing metastases of lymph nodes of women with breast cancer. As the name suggests, this dataset is only composed of the training set of the challenge. It contains a total of 270 whole slide images of sentinel lymph node collected in the Radboud University Medical Centre and the University Medical Centre Utrecht, both centres are located in the Netherlands. The training dataset is composed of 160 normal slides and 110 slides containing metastases. From these whole slide images, there are about 2,600 annotations labelled as a tumour. Some whole slide images with tumour annotations are shown in Figure 4.1, page 37.

4.1.2 CHALLENGE-GLAS-2015

This dataset comes from the challenge *Gland Segmentation Challenge Contest* (GlaS) [Sirinukunwattana et al., 2015, 2016] organised by the University of Warwick in conjunction with the MICCAI Society and held in Munich, Germany. The data were acquired by pathologists at the University Hospitals Coventry and Warwickshire, the United Kingdom. The dataset consists of 165 whole slide images derived from 16 Hematoxylin and Eosin (H&E) stained images of stage T3 or T4 colorectal adenocarcinoma. In the 165 images, 85 of them were used as the training set and the remaining 80 for the testing set in the challenge setup. The test images were split into TestA and TestB, respectively. From these images, there are about 1,600 gland annotations. Examples of such images are illustrated in Figure 4.2, page 38.

4.1.3 CHU-ANAPATH-NST-DL

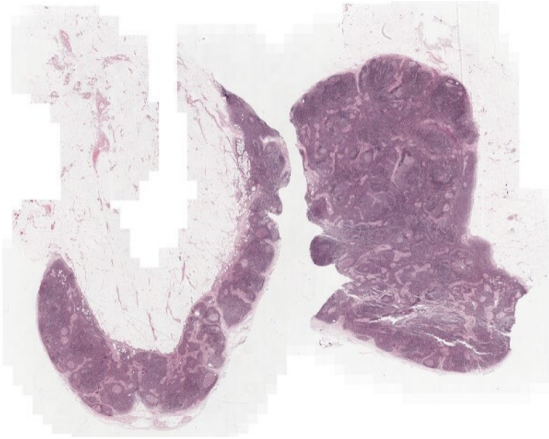
This dataset comes from the University Hospital Centre CHU of Liège, Belgium, more precisely from the Unit of Prof. Philippe Delvenne. It consists of 268 whole slide images with various complex objects, such as glands, infiltrations, inflammations, etc. There are 6,348 annotations of glands, 2,842 annotations of infiltrations, and 354 annotations of inflammations. One expert in the field of pathology, Michel Reginster, has annotated the whole slide images. One image of the dataset with different level of focus is shown in Figure 4.3, page 39.

4.1.4 ULG-LBTD-NEO04

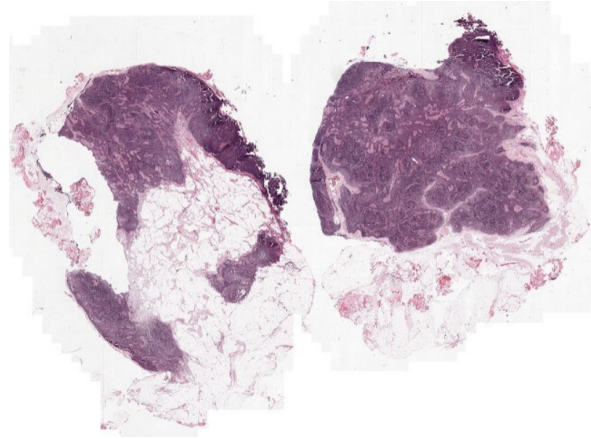
This dataset comes from the Laboratory of tumour and development biology (LBTD), more specifically from the Unit of Prof. Didier Cataldo, which is part of the GIGA research Institute within the GIGA-Cancer, located at the CHU Sart-Tilman in Liège, Belgium. It is mainly composed of annotations of bronchus and different types of tumours, such as adenocarcinoma, focal nodular hyperplasia, etc. More precisely, it consists of 126 whole slide images, with 384 annotations of bronchus and 492 tumour annotations. There are other annotations but are not listed because of their irrelevance in this context. Several expert pathologists, Didier Cataldo, Natacha Rocks, and Christine Fink, have annotated the whole slide images present in this dataset. An example whole slide image that comes from this is depicted in Figure 4.4, page 40.

4.1.5 ULG-LBTD-NEO13 (3)

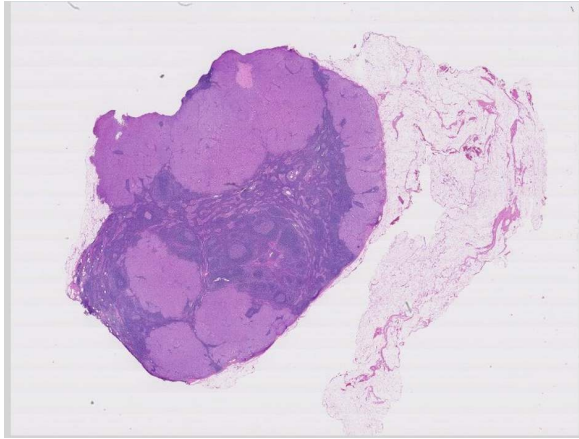
The provenance of the ULG-LBTD-NEO13 (3) dataset is the same as the ULG-LBTD-NEO04 and was annotated by the same experts. However, a slightly different preparation protocol was used, i.e., staining variations. This dataset is very similar to the ULG-LBTD-NEO04 dataset, it also contains the same type of annotations. More precisely, it consists of 414 annotations of bronchus, 175 tumour annotations and others non-relevant annotations. An illustration with different level of focus from this dataset is depicted in Figure 4.5, page 41.



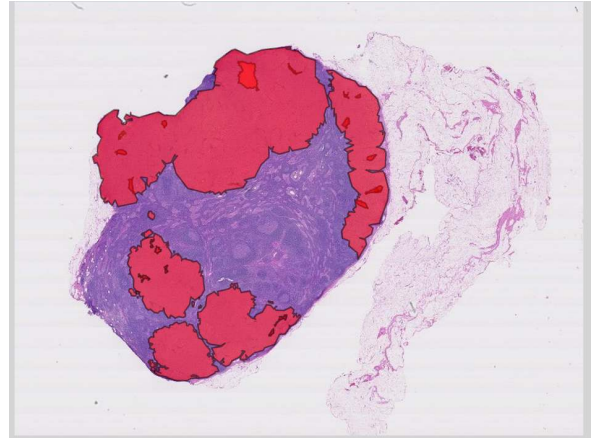
(a) Normal_009.tif (Magnification: 3.11 mm).



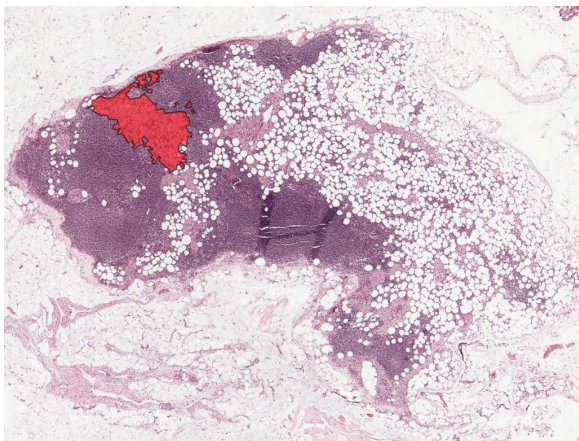
(b) Normal_032.tif (Magnification: 3.11 mm).



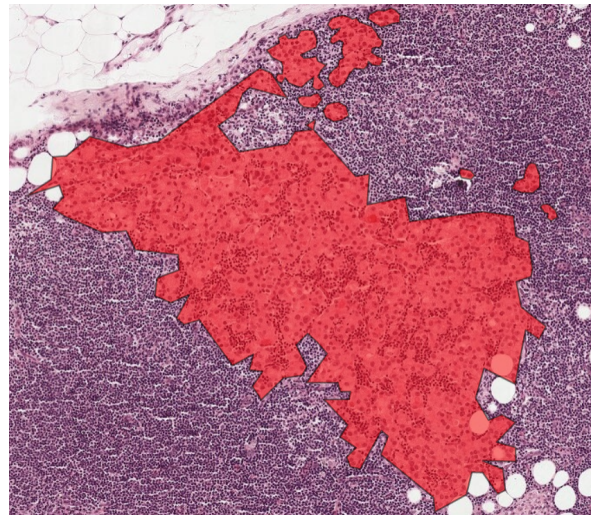
(c) Tumor_110.tif (Magnification: 2.90 mm).



(d) Annotations of tumour regions.

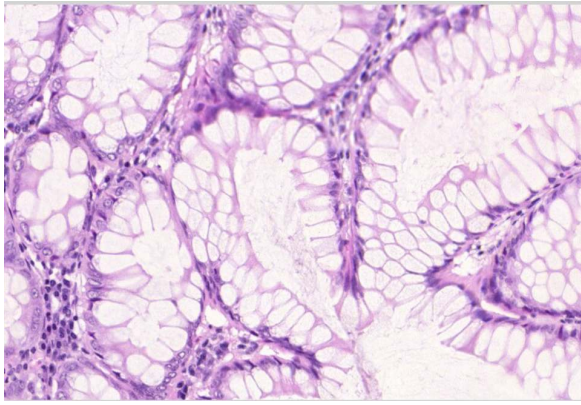


(e) Tumor_111.tif (Magnification: 778 μm).

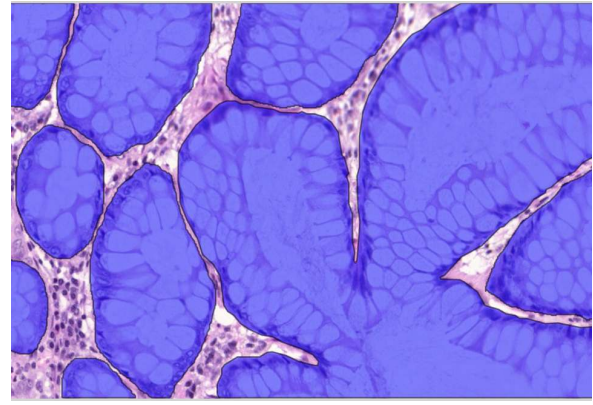


(f) Crop centred on an annotation of tumour.

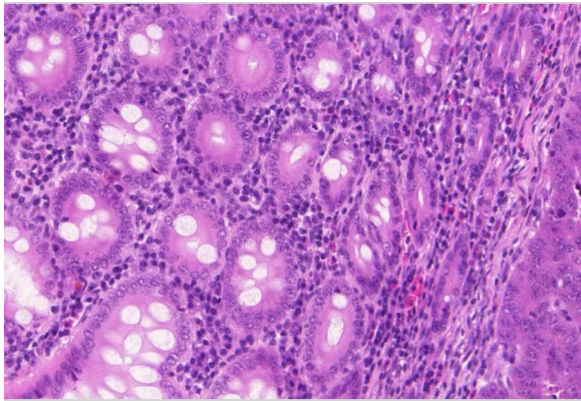
Figure 4.1: Example of whole slide images in the Camelyon16 dataset. The tumour regions can be seen in red.



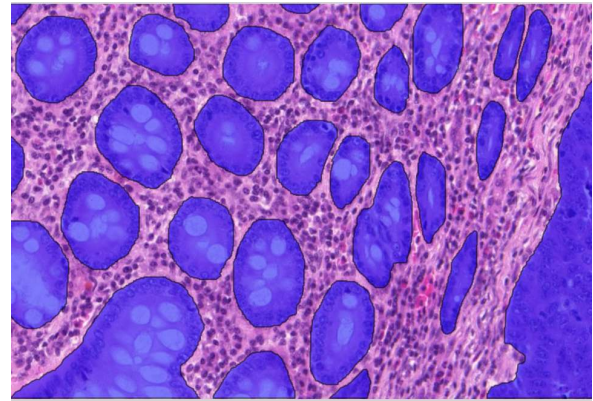
(a) testA_1.bmp



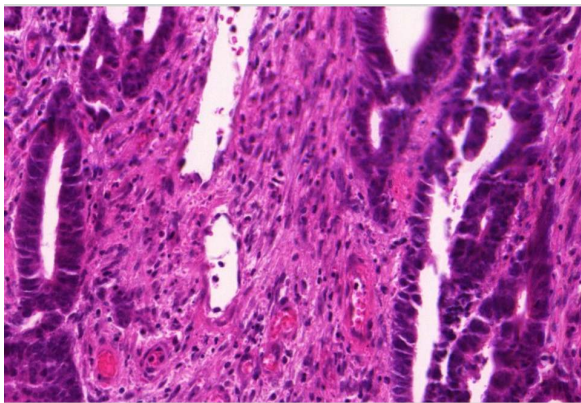
(b) Annotations of glands.



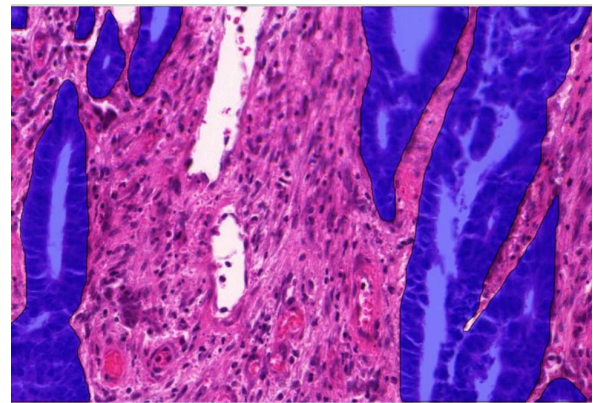
(c) testA_10.bmp



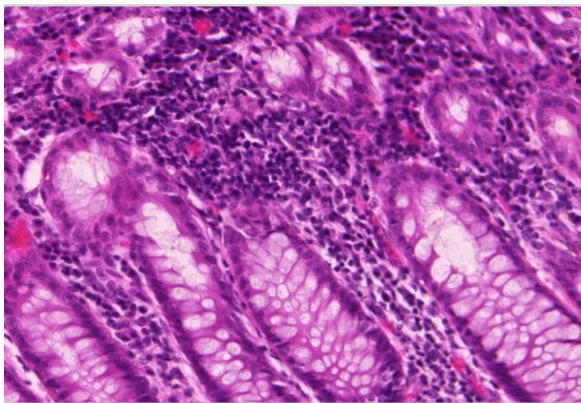
(d) Annotations of glands.



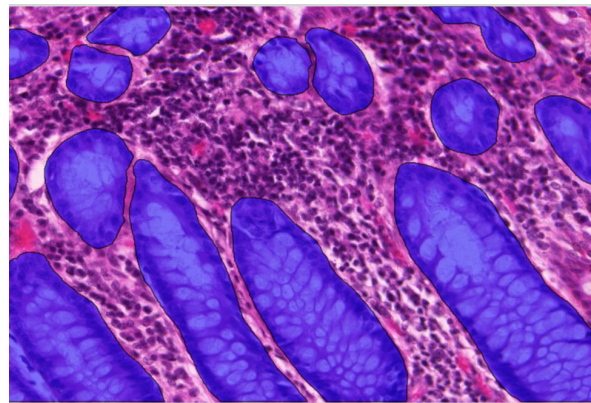
(e) testA_13.bmp



(f) Annotations of glands.

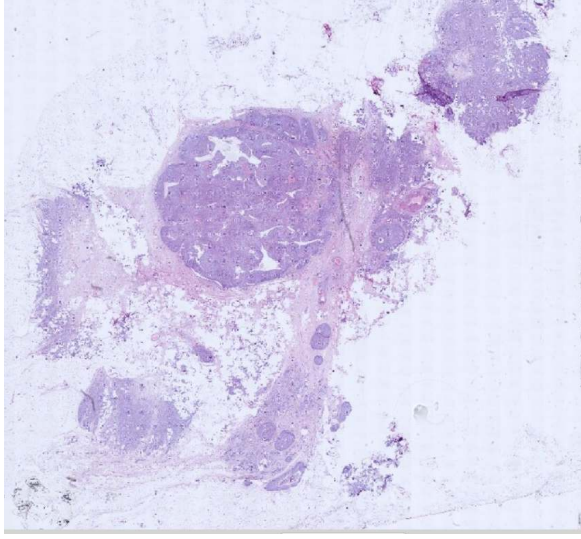


(g) testA_21.bmp

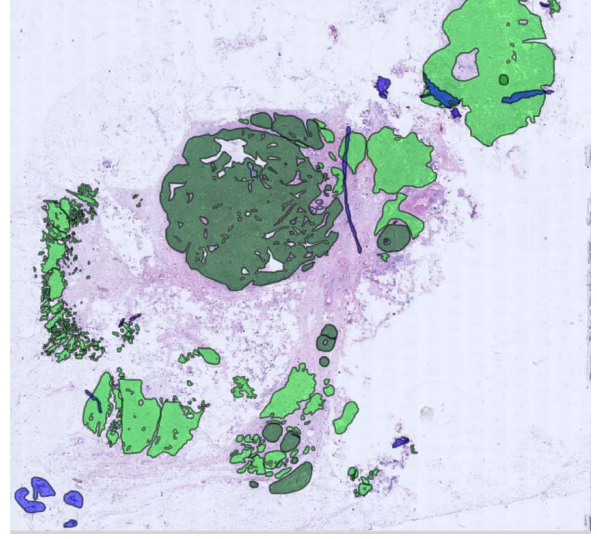


(h) Annotations of glands.

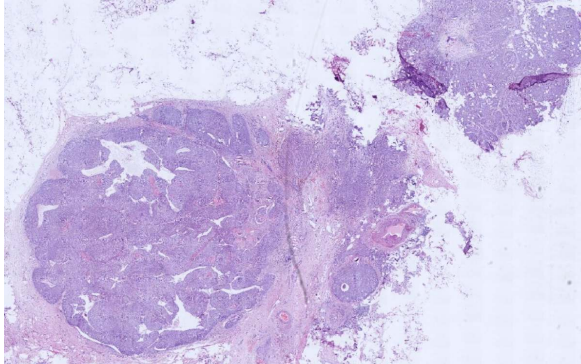
Figure 4.2: Example of images in the GlaS2015 dataset.



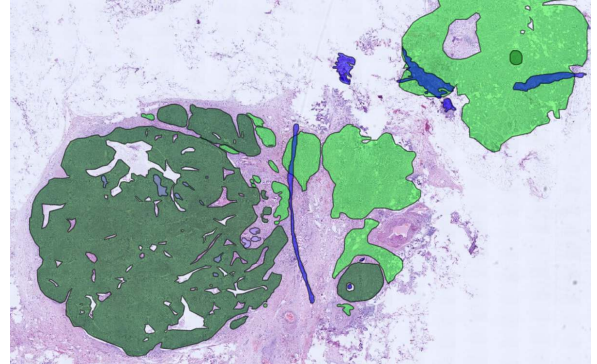
(a) Magnification $0.31\times$



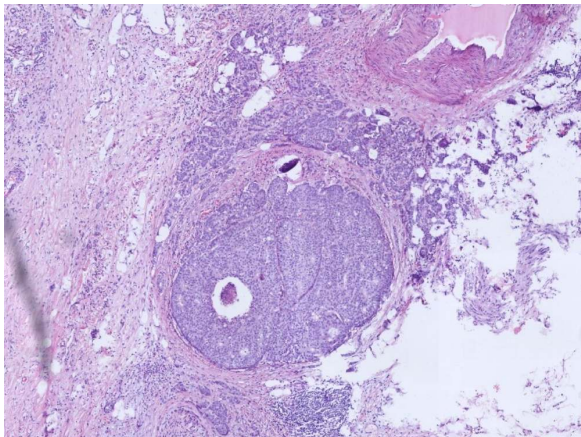
(b) The annotations



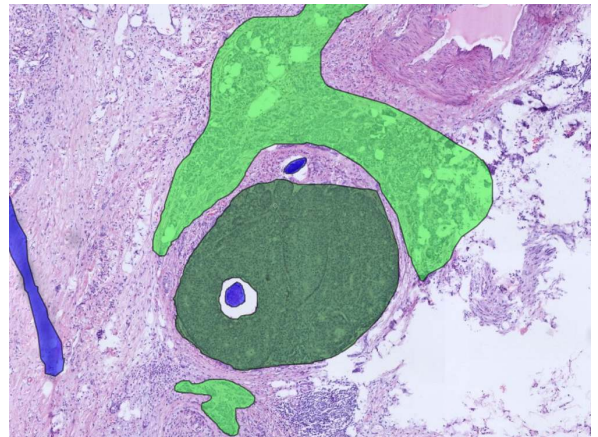
(c) Magnification $0.63\times$



(d) The annotations

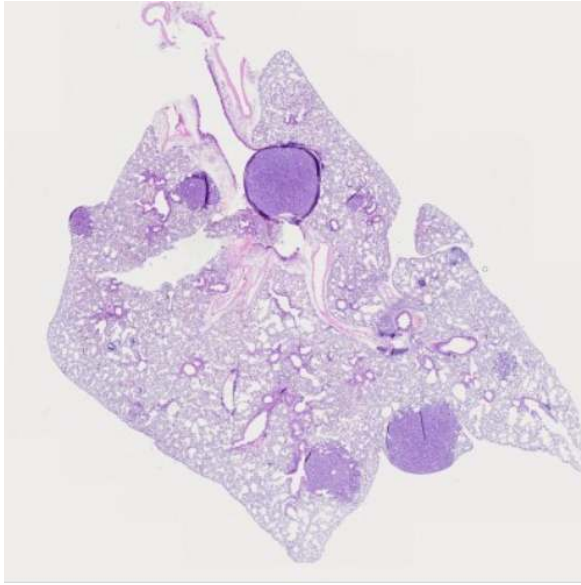


(e) Magnification $2.5\times$



(f) The annotations

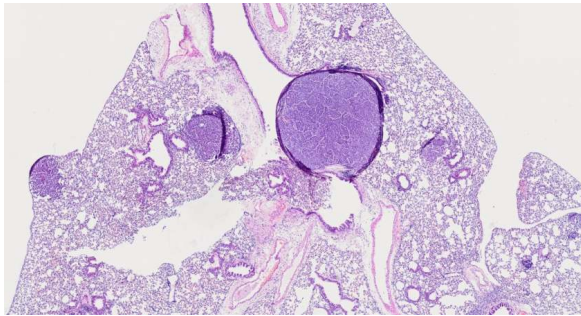
Figure 4.3: Example of a whole slide image at different magnifications in the CHU-ANAPATH-NST-DL dataset (Source: NEW_201708241740.tif). Infiltration can be seen in light green, In situ in dark green, and artefact in blue.



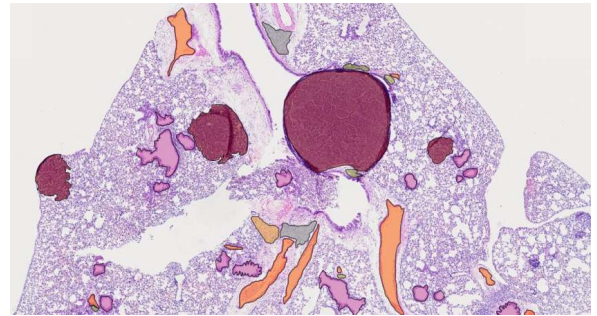
(a) Magnification $0.31\times$



(b) The annotations



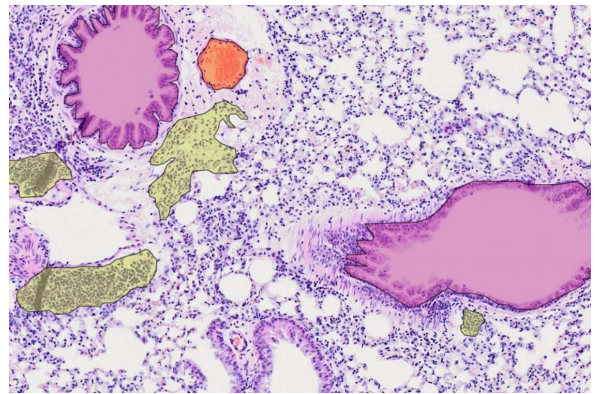
(c) Magnification $1.25\times$



(d) The annotations

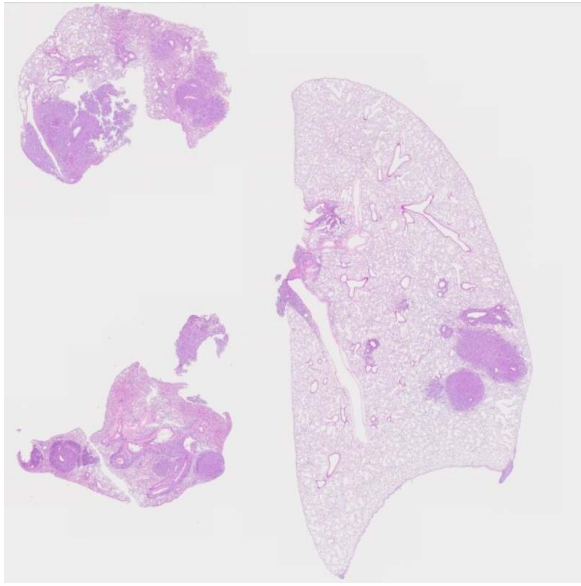


(e) Magnification $5\times$

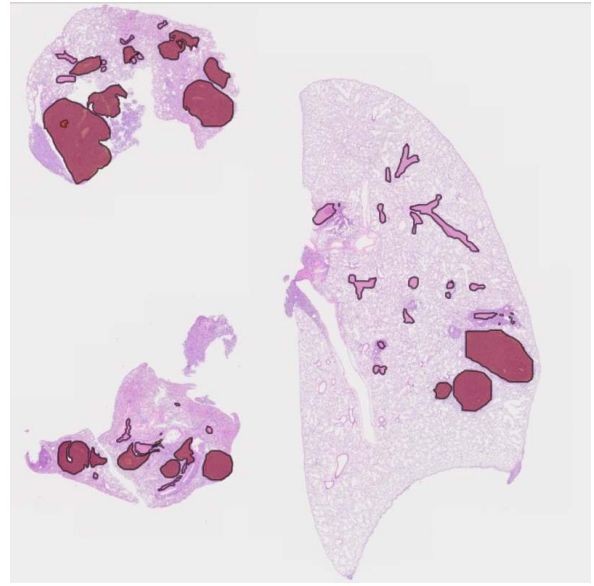


(f) The annotations

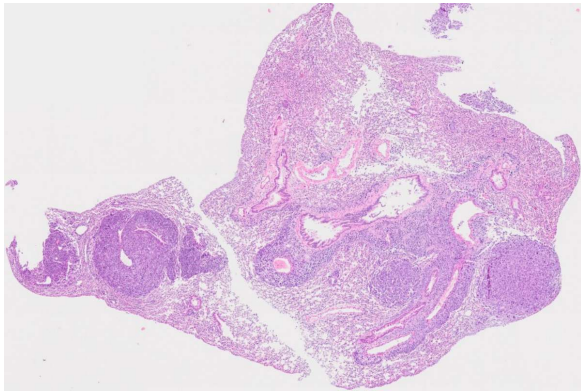
Figure 4.4: Example of a whole slide image at different magnifications in the ULG-LBTD-NEO04 dataset (Source: NEO4_CURCU_INH_8.20_01.tif). Bronchus annotations can be seen in purple, inflammation in yellow, and tumour in burgundy.



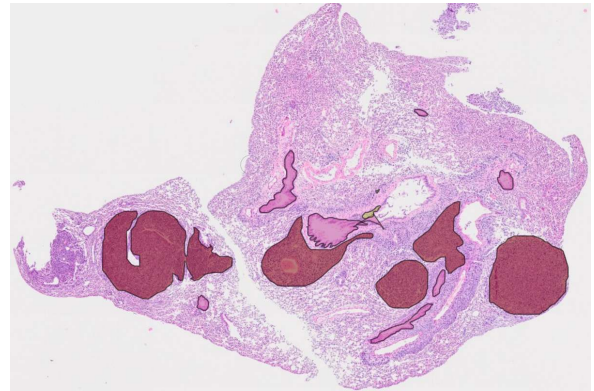
(a) Magnification $0.31\times$



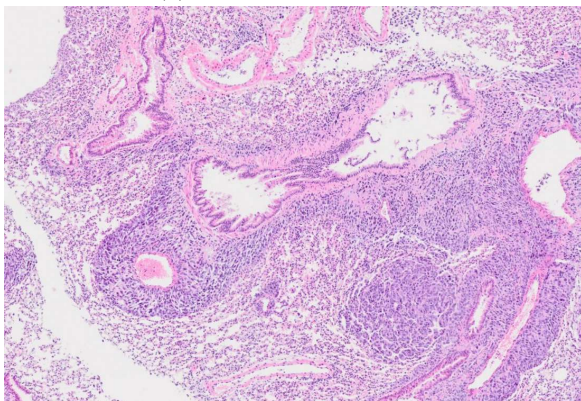
(b) The annotations



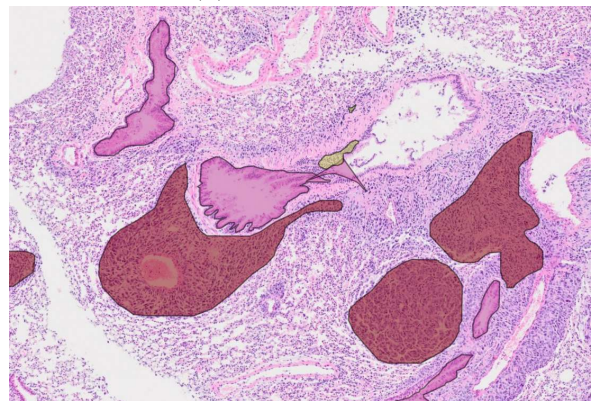
(c) Magnification $1.25\times$



(d) The annotations



(e) Magnification $2.5\times$



(f) The annotations

Figure 4.5: Example of a whole slide image at different magnifications in the ULG-LBTD-NEO13 (3) dataset (Source: NEO13_CNS_1.30_5_3_01.tif). In burgundy is shown tumour regions, bronchus in purple, and inflammation in yellow.

4.2 Replication of the original study

Before conducting any experiments, the first step is to try to replicate the performance achieved by NuClick in the original paper. For the replication, Table 4.3 reports the parameters used by the original implementation and Table 4.4 shows the dataset used. Only the experiments on the glands are performed.

Parameter	Value
Epochs	200
Batch size	16
Optimiser	Adam
Learning rate	3×10^{-3}
Weight decay	5×10^{-5}

Table 4.3: Parameters used by the original NuClick.

Dataset	Type	Train set	Val set	TestA	TestB
CHALLENGE-GLAS-2015	Gland	613	154	664	119

Table 4.4: The split used by the original NuClick.

Table 4.5 reports the performance of the original results achieved by NuClick and the performance obtained with the reimplemented version. As can be seen from the table, the reimplemented version of NuClick could not achieve the perfect score with the Dice coefficient. However, the reimplementation achieves a better Hausdorff distance in the two test sets. It is really surprising to see that the original performance achieves such a score with the Dice coefficient.

Model	TestA		TestB	
	Dice/F1 Score	Haus.	Dice/F1 Score	Haus.
Original NuClick	1.000	15	1.000	21
Replicated NuClick	0.9323	8	0.9329	13

Table 4.5: Performance comparison between the stated result from the NuClick paper and the reimplemented version.

4.3 Experiments protocol

In this section, the various settings used for all the subsequent experiments are defined.

4.3.1 Datasets

Since the experiments aims at reflecting as close as possible a real use case, not all the available data, shown in Table 4.1, page 35, are going to be used. Table 4.6 reports the number of annotations for each of the type that is going to be used for the experiments. To reflect at best a situation where an annotator possesses very little annotations, a maximum of about 500 annotations is put for the training set and 250 annotations for the validation set. However, no limit is applied on the test set to have a proper performance assessment.

Dataset	Type	Train set	Val set	Test set
ULG-LBTD-NEO04	Bronchus	242	65	72
ULG-LBTD-NEO13	Bronchus	262	73	74
CHALLENGE-GLAS-2015	Gland	517	250	304
CHU-ANAPATH-NST-DL	Gland	508	250	1,239
ULG-LBTD-NEO04	Inflammation	94	24	30
CHU-ANAPATH-NST-DL	Inflammation	226	59	69
CHU-ANAPATH-NST-DL	Infiltration	518	250	549
CHALLENGE-CAMELYON16	Tumour	494	250	498

Table 4.6: The split of annotations for each type of object in the various datasets.

4.3.2 Model training

Regarding the training of the models, Table 3.2, page 34, reports the hyperparameters that are going to be used for all the experiments. The training is performed with the help of the training and validation sets. The models are trained on GPUs as explained in section 3.5, page 33. The scribbles (inclusion and exclusion maps) are automatically generated during the training using the method described in section 3.3.1, page 27.

4.3.3 Model evaluation

The evaluation of a model segmenting a type of object is done on the corresponding test set, e.g., a model trained on ULG-LBTD-NEO04 to segment bronchus is tested on the ULG-LBTD-NEO04 bronchus test set. An exception is made for the robustness analysis in section 4.5, page 66, where the datasets used are going to be clearly stated. To evaluate a model, the metrics presented in section 3.4, page 31, are going to be used, namely the intersection over the union and the Hausdorff distance.

4.3.4 Assessment standard

In this thesis, the assessment standard to be considered as good performance are an intersection over the union greater or equal to a value of 0.7 and a Hausdorff distance lower or equal than a value of 20. The values are chosen to try to be the most balanced as possible. A too low value for the IoU would let some results considered as good although it is not. In contrast, a too high value will only let a small part of the results be considered as good. As for the Hausdorff distance, the smaller the value, the better the segmentation.

4.4 Annotations analysis

This section is dedicated to the analysis of the annotations in term of quality and quantity. The quantity analysis aims at determining the minimum number of annotations needed to train a model achieving satisfactory performances. On the other hand, the quality analysis assesses the *quality* of the annotations made by potential users on the objects of interest.

4.4.1 Quantity analysis

In this experiment, a model is going to be trained on a growing number of annotations starting from one annotation up to a certain number of annotations that is going to be precised for each type of object. Training a model by incrementing the number of annotations by one is infeasible because it requires an excessive amount of time. Therefore, the number of annotations grows by an arbitrarily step. The models are then evaluated on the test set with the metrics presented in section 3.4, page 31, namely the intersection over the union and the Hausdorff distance.

Regarding the step size used for the experiments, a reasonable size is computed to approximate a real feasible size of annotations. After numerous trials, a step size of approximately 24 annotations is chosen for the subsequent experiments. An exception is made for the inflammations, for which there are very few available annotations. Depending on the available number of annotations, the step size might slightly change to have more or less an evenly spaced number of annotations and about 20 models per type of object to segment. The exact number of annotations and models are going to be shown in the tables for each of the following experiment.

Bronchus

Figure 4.6 reports the IoU and the Hausdorff distance obtained for the two datasets containing annotations of bronchus, namely the ULG-LBTD-NEO04 and the ULG-LBTD-NEO13 (3) datasets. As a general observation for the figure regarding the IoU, the more annotations are added for the training, the better the performance, which is the expected behaviour. Notice that some sudden decrease in performance can be recognised, e.g., at about 75 annotations for the blue curve, or about 120 annotations for the orange curve. After the investigation of the dataset, the explanation of the decrease in performance is because the newly added images have more than one mask of bronchus. Usually, annotations of bronchus are only composed of one mask as shown in Figure 4.7a and Figure 4.7b. Therefore, the network tries to incorporate this information that several masks can appear, which explains the decrease.

Aiming at 0.7 IoU, the required number of annotations seems to be located approximately around 70 and 120 annotations with the consideration of the sudden decrease at 70 annotations. Needless to say, to achieve a higher IoU, more than 100 annotations are required as the minimum number of annotations. On the other hand, from the performance given by the Hausdorff distance, it can be noticed that the models trained on the ULG-LBTD-NEO04 dataset seem to have a larger Hausdorff distance than the other dataset. This is an unexpected result as both datasets share high similarity in images. After the investigation of the predicted segmentation in both datasets for the test set, poor predictions, illustrated in Figure 4.8, are sometimes encountered in the ULG-LBTD-NEO04 dataset, thus leading to a higher Hausdorff distance. Lastly, an illustration of a good prediction is shown in Figure 4.7.

Regarding the computation time, Table 4.7 presents the time taken to train the models with varying size of annotations. To achieve a good performance, it requires between 5 to 8 hours, which is already enormous in term of time.

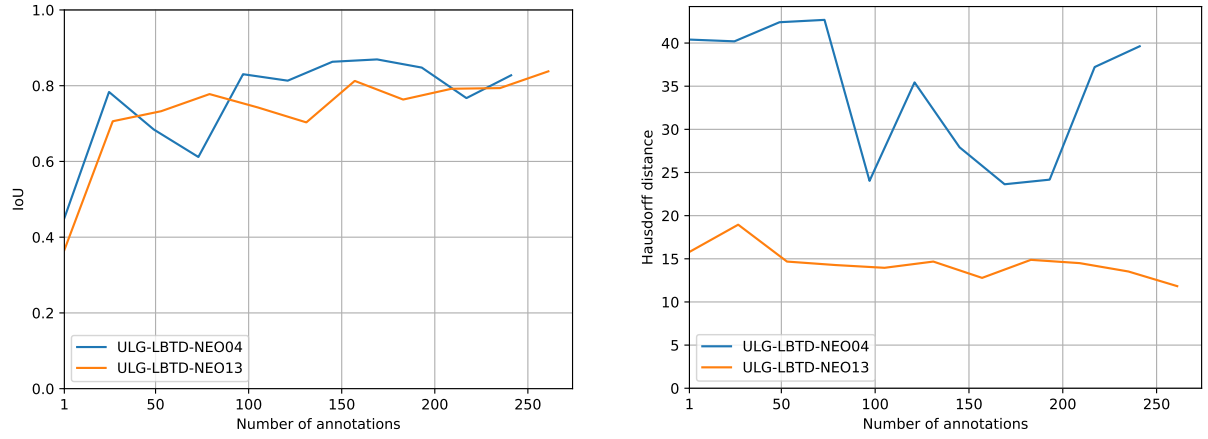


Figure 4.6: Performance for the bronchus. On the left is the performance using the intersection over the union metric and on the right is the Hausdorff distance.

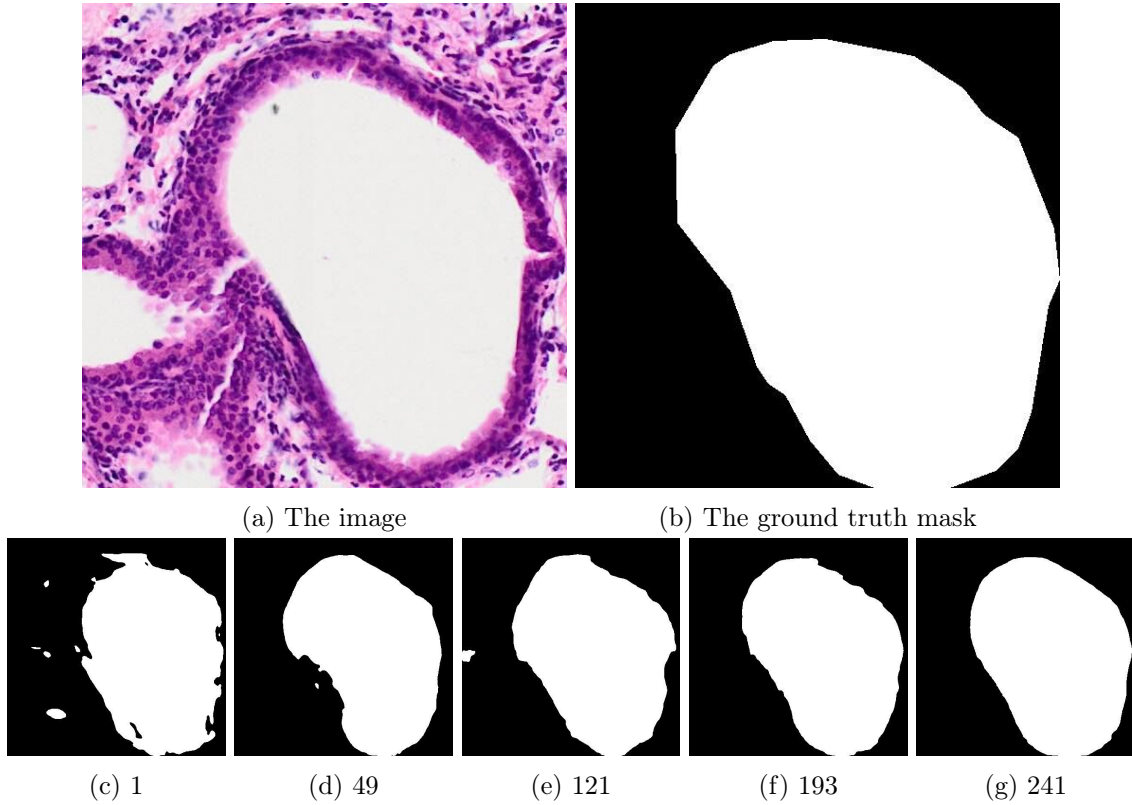


Figure 4.7: Example of a good segmentation of bronchus. The number from the subfigure (c) to (g) shows the number of annotations used for the training.

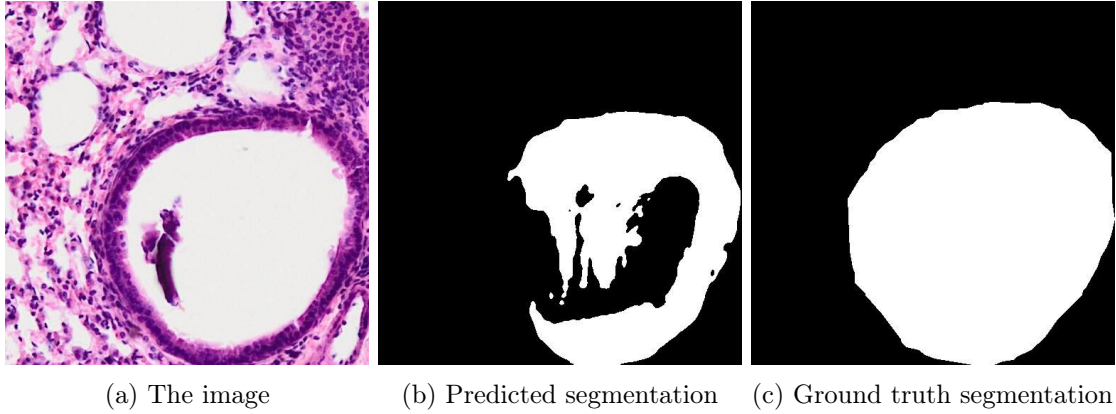
Number of annotations	Average time for one epoch	Total time
1	27s	2h15m24s
25	39s	3h16m42s
49	52s	4h20m28s
73	1m06s	5h28m36s
97	1m20s	6h40m34s
121	1m34s	7h49m35s
145	1m46s	8h50m54s
169	1m59s	9h54m47s
193	2m18s	11h31m12s
217	2m27s	12h13m43s
241	2m42s	13h20m58s

(a) ULG-LBTD-NEO04

Number of annotations	Average time for one epoch	Total time
1	33s	2h44m19s
25	47s	3h52m34s
49	55s	4h34m49s
73	1m12s	6h2m20s
97	1m20s	6h40m7s
121	1m33s	7h44m18s
145	1m53s	9h25m17s
169	1m58s	9h49m54s
193	2m11s	10h53m0s
217	2m33s	12h47m14s
241	2m34s	12h50m15s

(b) ULG-LBTD-NEO13 (3)

Table 4.7: Computation time regarding the trainings of the bronchus segmentation task.



(a) The image

(b) Predicted segmentation

(c) Ground truth segmentation

Figure 4.8: Poor bronchus segmentation generating a high Hausdorff distance value, i.e., 150.

Gland

Originally, this thesis bases itself on the original paper of NuClick [Aleml Koohbanani et al., 2020]. A replication study was done in section 4.2, page 42, to try to achieve the stated performance. In this experiment, the number of annotations used differs from the original dataset of the CHALLENGE-GLAS-2015 dataset. For instance, as described in subsection 4.1.2, page 36, the dataset is composed of a training set, and two test sets. Here, the annotations are shuffled so that some of the original testing images can be seen in the training set. Figure 4.9 reports the performance of the evaluation on the two datasets containing annotations of glands, namely the CHALLENGE-GLAS-2015 and the CHU-ANAPATH-NST-DL datasets. As can be seen from this table, the performance of the CHU-ANAPATH-NST-DL seems to achieve a higher performance in both metrics than the CHALLENGE-GLAS-2015.

Comparing the computation time reported in Table 4.8 with the one of the bronchus, a huge difference can be observed, for instance, training the model on one annotation takes about 2 hours for the bronchus, whereas it takes about 8 to 9 hours. This difference of time can be explained by the fact that the validation phase of the gland takes more times as the number of validation annotations are three times more than the one of the bronchus, as can be seen from Table 4.6, page 43. Lastly, an example of a good and a poor result are shown in Figure 4.10 and Figure 4.11, respectively.

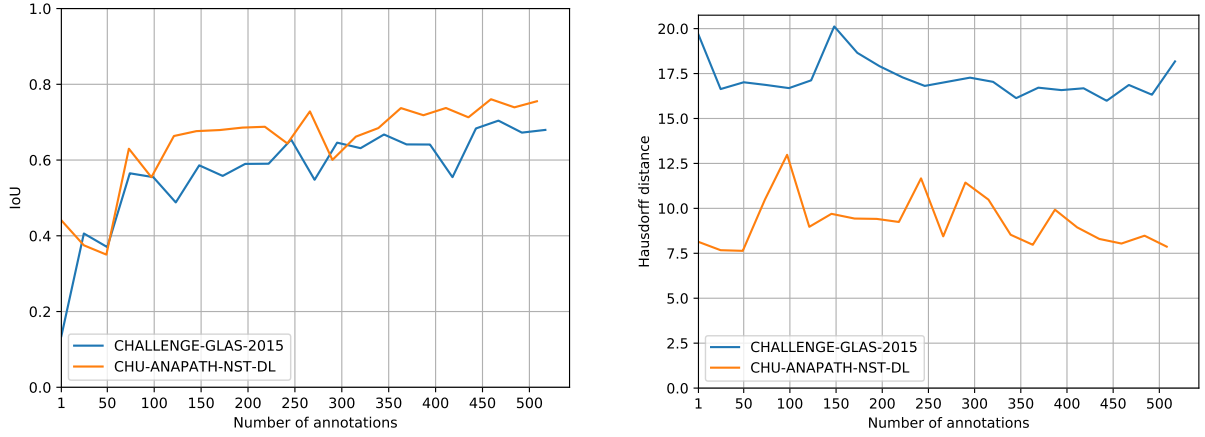


Figure 4.9: Performance for the gland. On the left is the performance using the intersection over the union metric and on the right is the Hausdorff distance.

Number of annotations	Average time for one epoch	Total time	Number of annotations	Average time for one epoch	Total time
1	1m46s	8h12m49s	266	4m13s	21h03m26s
25	2m05s	10h29m54s	290	4m01s	20h04m41s
49	2m10s	10h49m03s	315	4m08s	20h40m48s
73	2m36s	12h58m18s	339	4m35s	22h52m38s
97	2m44s	13h38m08s	363	5m10s	1d01h47m51s
121	2m55s	14h33m33s	387	4m54s	1d00h30m56s
145	2m56s	14h39m49s	411	4m49s	1d00h05m59s
170	3m27s	17h13m51s	435	5m27s	1d03h14m43s
194	3m47s	18h53m50s	459	5m53s	1d04h25m16s
218	3m48s	18h59m25s	484	5m50s	1d04h07m42s
242	3m31s	17h36m26s	508	5m16s	1d02h19m57s

(a) CHU-ANAPATH-NST-DL

Number of annotations	Average time for one epoch	Total Time	Number of annotations	Average time for one epoch	Total time
1	1m53s	9h24m43s	271	3m49s	19h5m36s
25	2m7s	10h34m17s	320	3m48s	19h1m56s
50	2m17s	11h27m20s	295	4m12s	20h57m59s
74	2m30s	12h28m47s	369	4m23s	21h55m10s
99	2m47s	13h54m7s	345	4m32s	22h41m28s
123	2m41s	13h25m28s	394	4m31s	22h34m51s
148	2m59s	14h53m40s	418	4m56s	1d00h38m8s
173	3m8s	15h40m42s	443	5m10s	1d01h51m39s
197	3m19s	16h33m46s	467	5m20s	1d02h40m30s
222	3m28s	17h21m58s	492	5m33s	1d03h45m46s
246	3m39s	18h15m0s	517	5m36s	1d03h57m34s

(b) CHALLENGE-GLAS-2015

Table 4.8: Computation time regarding the trainings of the gland segmentation task.

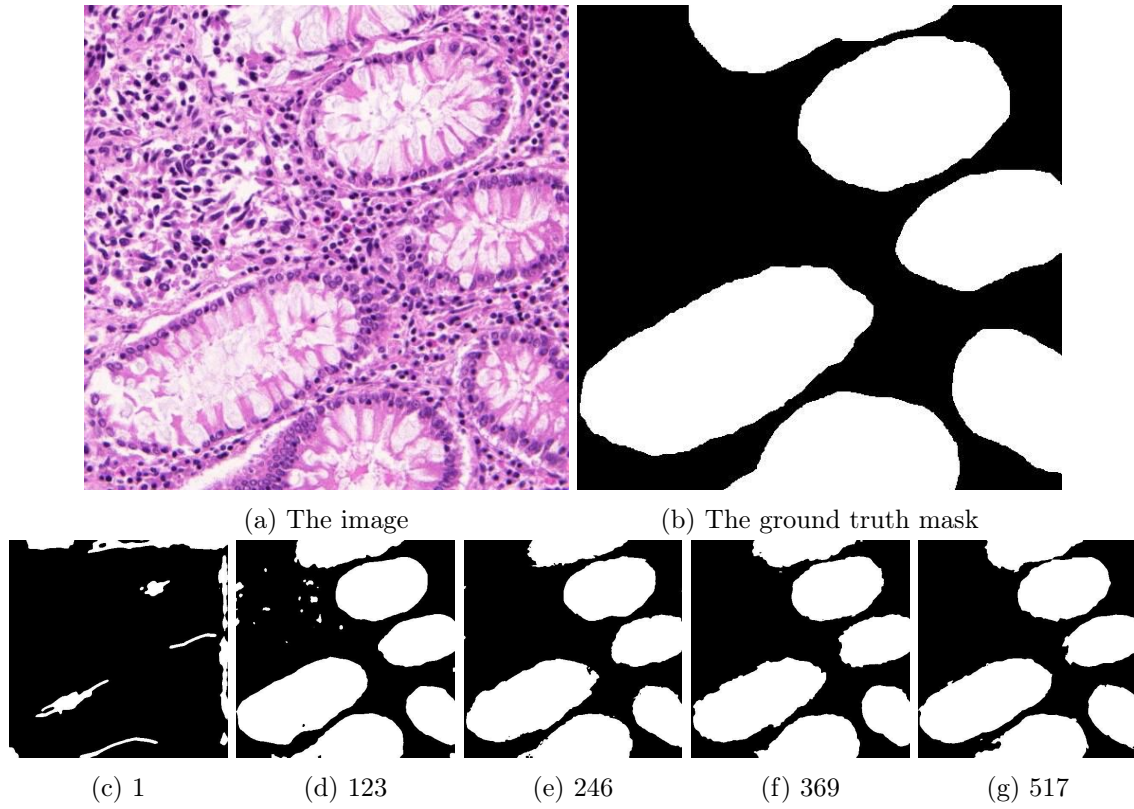


Figure 4.10: Example of a good segmentation of gland. The number from the subfigure (c) to (g) shows the number of annotations used for the training.

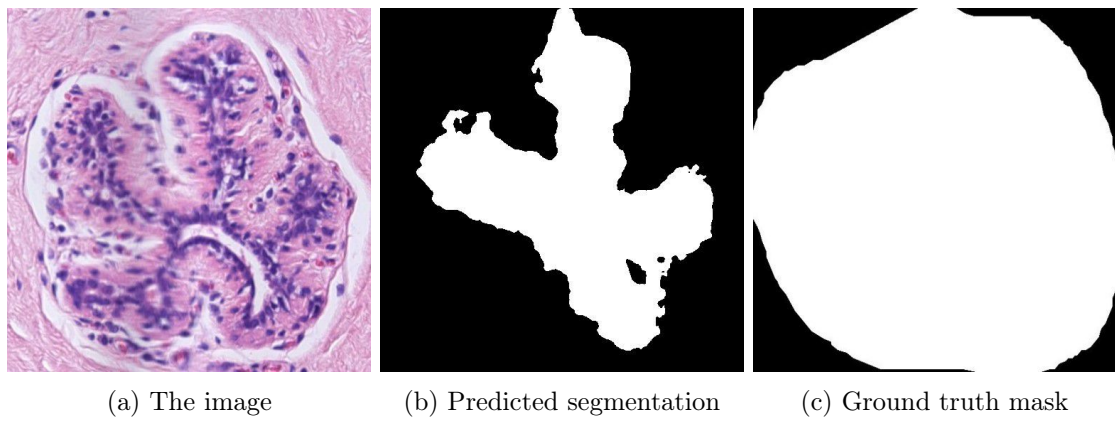


Figure 4.11: Poor gland segmentation from the CHU-ANAPATH-NST-DL dataset.

Inflammation

As previously mentioned in section 4.3, page 43, there is little available number of annotations for the inflammation. Therefore, the step size is adapted, regarding the CHU-ANAPATH-NST-DL dataset, the step size is about 20 annotations and for the ULG-LBTD-NEO04 dataset, the step size is 8 annotations, since the training set is only composed of 94 annotations.

Figure 4.12 presents the performance measured by the IoU and the Hausdorff distance. It can be observed for the ULG-LBTD-NEO04 dataset that the performance are very poor. Compared to the previous results about bronchus and gland, the models segmenting inflammation from the ULG-LBTD-NEO04 dataset achieve an IoU of at best 0.6. After the investigation of the images from the dataset, the presumable cause is the class imbalance. Since the network takes inputs of dimensions $512 \times 512 \times 5$, crops of 512×512 of height and width, respectively, are extracted from the whole slide images. However, the mask associated with the inflammatory regions are frequently very small resulting in a substantial class imbalance, i.e., the background covers most of the ground truth image as depicted in Figure 4.13. Nonetheless, the models seem to achieve higher IoUs on the CHU-ANAPATH-NST-DL dataset. It is mainly due to the fact that inflammatory regions are more consequent in the ground truth masks reducing slightly the class imbalance as shown in Figure 4.14. However, the ground truth masks present a tremendous amount of small regions labelled as inflammation, for which the NuClick architecture has some issues managing these small regions. Nevertheless, some illustrations of mediocre and poor segmentation are shown in Figure 4.15.

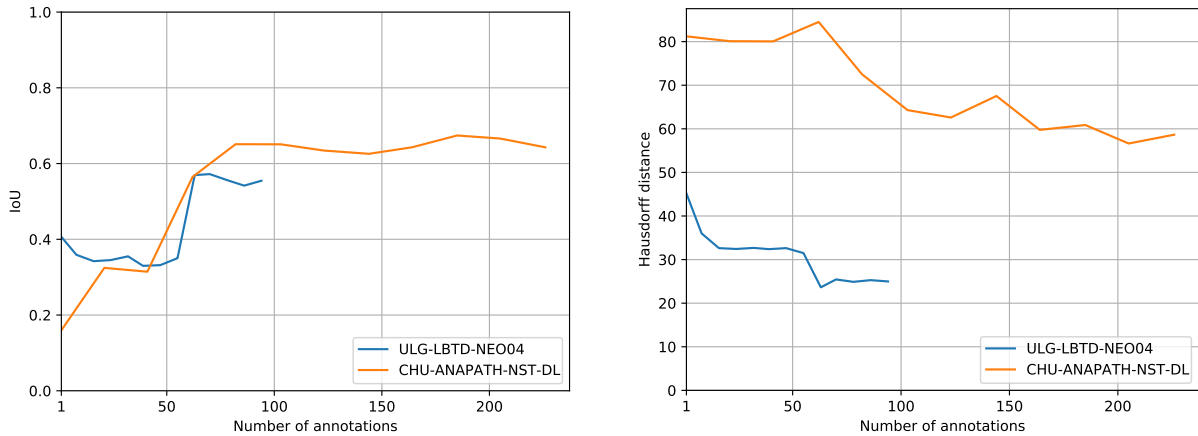


Figure 4.12: Performance for the inflammation. On the left is the performance using the intersection over the union metric and on the right is the Hausdorff distance.

Regarding the performance of the Hausdorff distance for both datasets from Figure 4.12, the Hausdorff distance for the CHU-ANAPATH-NST-DL is approximately double the distance of the ULG-LBTD-NEO04 one. After the comparison of annotated images in both datasets, the cause is related to the diversity of inflammatory regions in the CHU-ANAPATH-NST-DL dataset as shown in Figure 4.14. Thus, the network has more difficulties producing accurate segmentation at the boundaries primarily. The small conclusion that can be drawn from this specific experiment is that the NuClick architecture is not designed for this kind of annotations, i.e., numerous very small regions.

Lastly, with respect to the computation time reported in Table 4.9, the time taken to train the models on the inflammation is comparable to the time for the bronchus datasets.

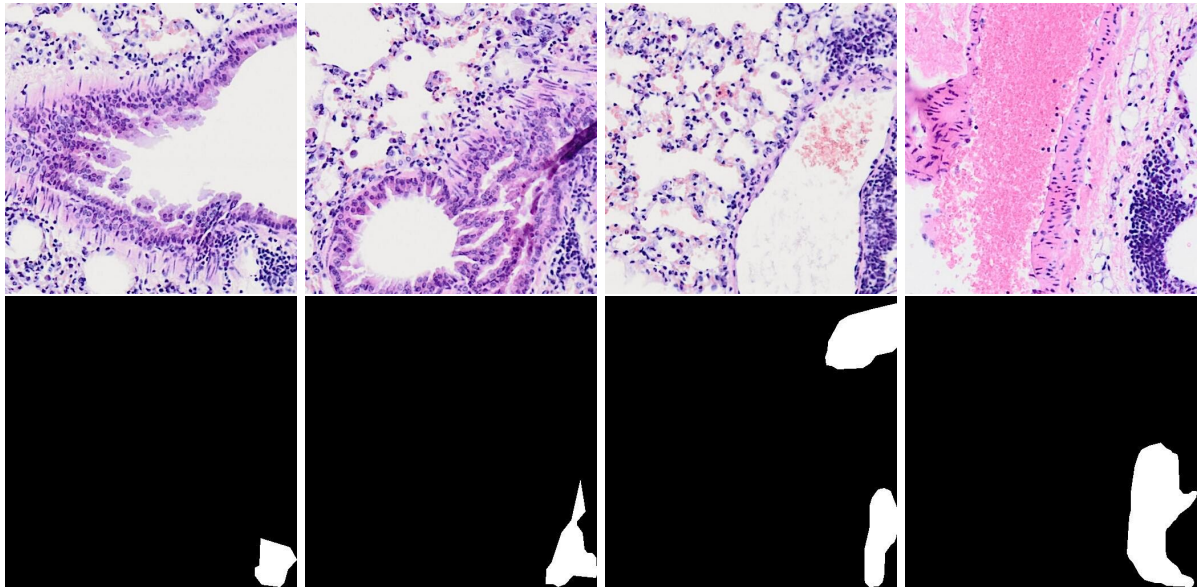


Figure 4.13: Images with their corresponding ground truth mask of inflammatory regions in the ULG-LBTD-NEO04 dataset.

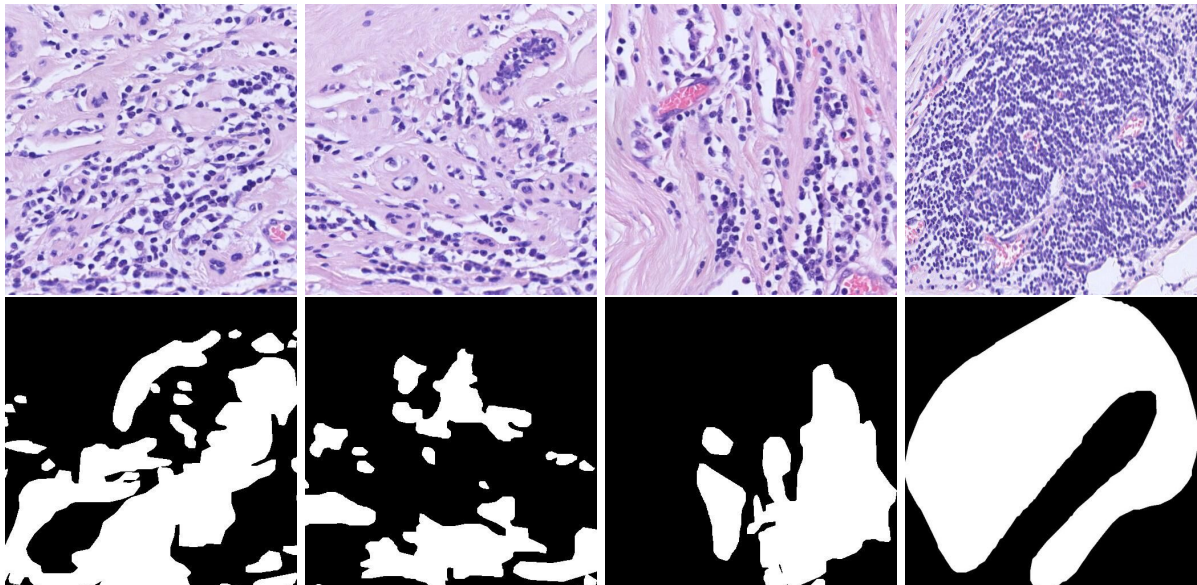


Figure 4.14: Images with their corresponding ground truth mask of inflammatory regions in the CHU-ANAPATH-NST-DL dataset.

Number of annotations	Average time for one epoch	Total time
1	0m30s	2h32m21s
21	0m38s	3h11m31s
41	0m46s	3h50m30s
62	0m58s	4h49m56s
82	1m12s	5h57m31s
103	1m18s	6h32m25s
123	1m26s	7h10m26s
144	1m38s	8h11m45s
164	1m54s	9h30m55s
185	1m59s	9h53m26s
205	2m13s	11h06m08s
226	2m24s	12h02m24s

(a) CHU-ANAPATH-NST-DL

Number of annotations	Average time for one epoch	Total time
1	0m8s	41m50s
8	0m12s	58m05s
16	0m14s	1h11m24s
24	0m20s	1h37m39s
32	0m21s	1h46m40s
39	0m26s	2h12m24s
47	0m30s	2h30m04s
55	0m31s	2h37m27s
63	0m35s	2h55m56s
70	0m41s	3h22m39s
78	0m43s	3h36m19s
86	0m45s	3h44m22s
94	0m50s	4h10m27s

(b) ULG-LBTD-NEO04

Table 4.9: Computation time regarding the trainings of the inflammation segmentation task.

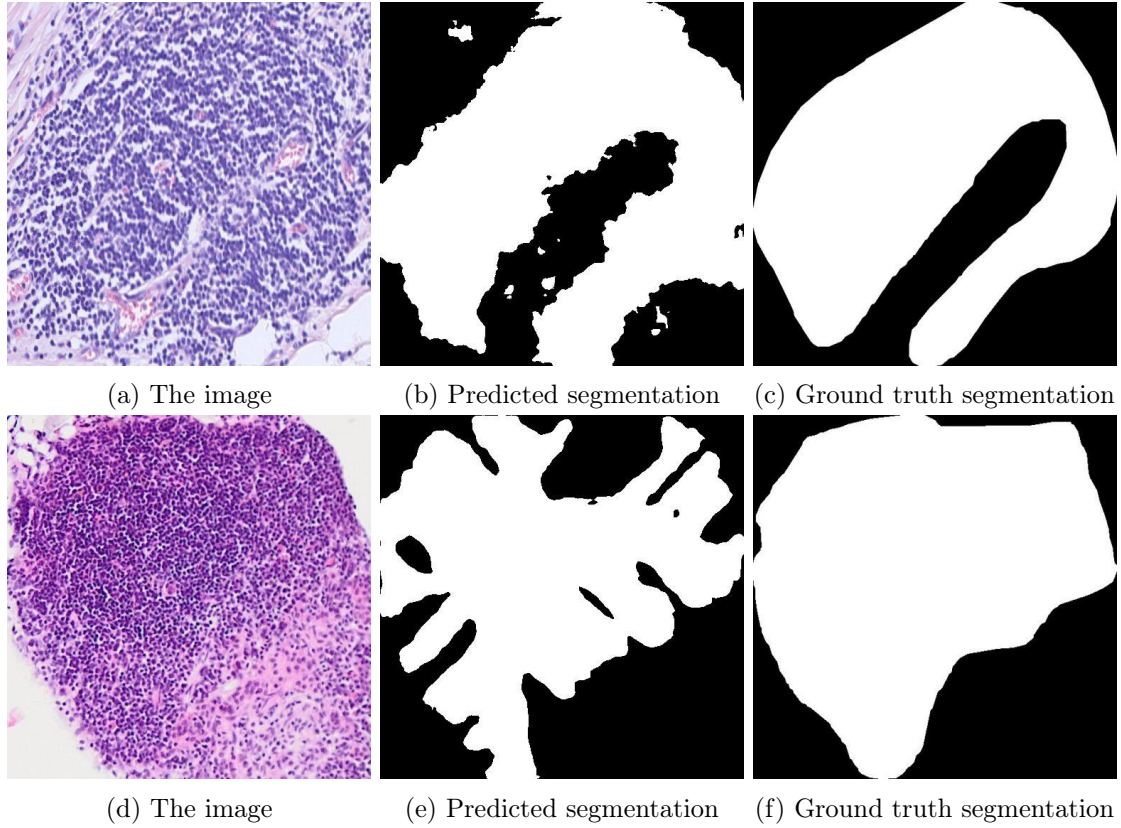


Figure 4.15: Illustration of a mediocre segmentation (upper) from the CHU-ANAPATH-NST-DL dataset and a poor segmentation (lower) of inflammation from the ULG-LBTD-NEO04.

Infiltration

Figure 4.16 shows the performance of the models on the infiltration test set. Ignoring the decrease at about 150 annotations in the IoU, the performance on the infiltration seems to be steady with an increasing number of annotations. Similar to the bronchus analysis, about 100 annotations are needed to achieve satisfactory results, i.e., an IoU of 0.7. After the examination of the 24 added ground truth masks to the training set for the decrease at about 150 annotations, the cause comes from the fact that these images are partially annotated and contains numerous small regions as shown in Figure 4.17. Regarding the performance of the Hausdorff distance, it is very low, indicating that the segmentation around the boundaries is close to the ground truth boundaries, which is very good. In this specific case, a relationship can be observed between the IoU and the Hausdorff distance, i.e., whenever there is a decrease in the IoU, there is an increase in the Hausdorff distance. It means that the neural network has more difficulties producing very accurate segmentation at the boundaries. Thus, leading to a small increase in the Hausdorff distance.

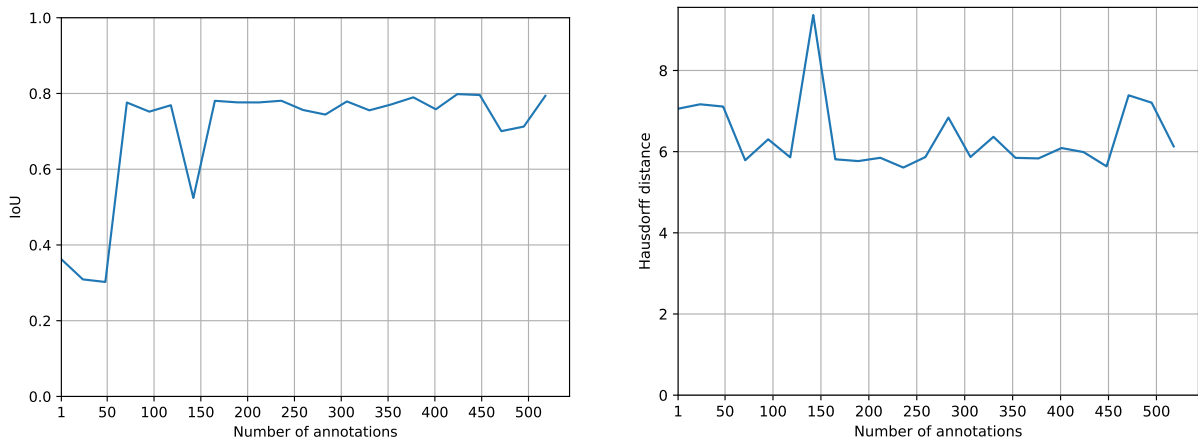


Figure 4.16: Performance for the infiltration.

Regarding the steadiness of the performance, a solution that could improve the performance is to perform specific fine-tuning in the hyperparameters. As formerly stated, the hyperparameters used for conducting this experiment is the same for all the other experiments and are shown in Table 3.2, page 34. First, performing a cross-validation search to determine the best optimiser, learning rate, and weight decay is to be made. Then, the number of epochs is also to be calculated to produce the best performance. These specific fine-tuning for the infiltration should hopefully lead to an increase in performance.

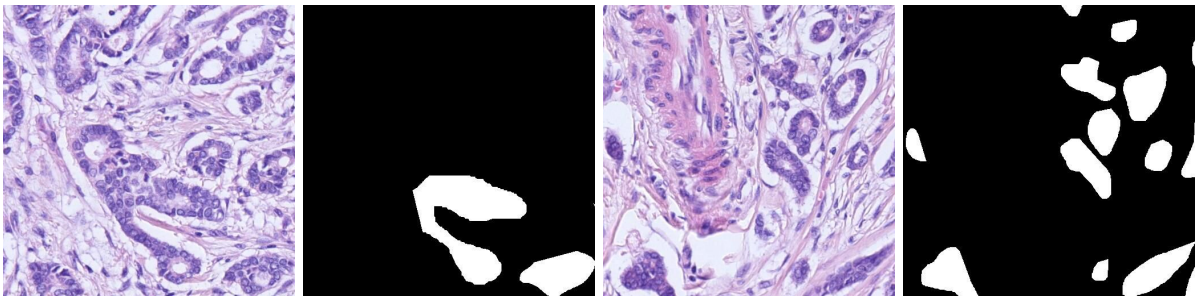


Figure 4.17: Illustrations of problematic images of infiltration from the CHU-ANAPATH-NST-DL dataset. An partially annotated image is shown on the left and several small regions of mask is shown on the right

For the computation time, Table 4.10 reports the time taken to for all the training in the experiment. Compared to the time for the bronchus in Table 4.7, page 46, it takes about four times more for training the model on one epoch. It is mainly due to the size of the validation set, similarly explained for the glands. To reach a performance achieving 0.7 IoU, about 100 annotations are needed as previously stated. Looking at the table, it requires about 13 hours to train the network to achieve the expected performance, which is objectively speaking very long compared to other models. Lastly, a good segmentation of infiltration is shown in Figure 4.18.

Number of annotations	Average time for one epoch	Total time	Number of annotations	Average time for one epoch	Total time
1	1m51s	9h13m10s	283	4m17s	21h27m08s
24	1m54s	9h28m25s	306	4m41s	23h26m08s
48	2m8s	10h37m41s	330	4m35s	22h56m33s
71	2m14s	11h09m56s	353	4m60s	1d00h57m44s
95	2m35s	12h56m07s	377	5m26s	1d03h07m55s
118	2m41s	13h24m29s	401	5m33s	1d03h46m26s
142	2m56s	14h40m24s	424	5m38s	1d04h10m55s
165	3m19s	16h34m29s	448	5m50s	1d05h08m57s
189	3m26s	17h11m56s	471	6m1s	1d06h03m49s
212	3m50s	19h10m10s	495	6m19s	1d07h37m24s
236	3m49s	19h03m25s	518	6m15s	1d07h17m14s
259	4m20s	21h37m46s	X	X	X

Table 4.10: Computation time regarding the trainings of the infiltration segmentation task.

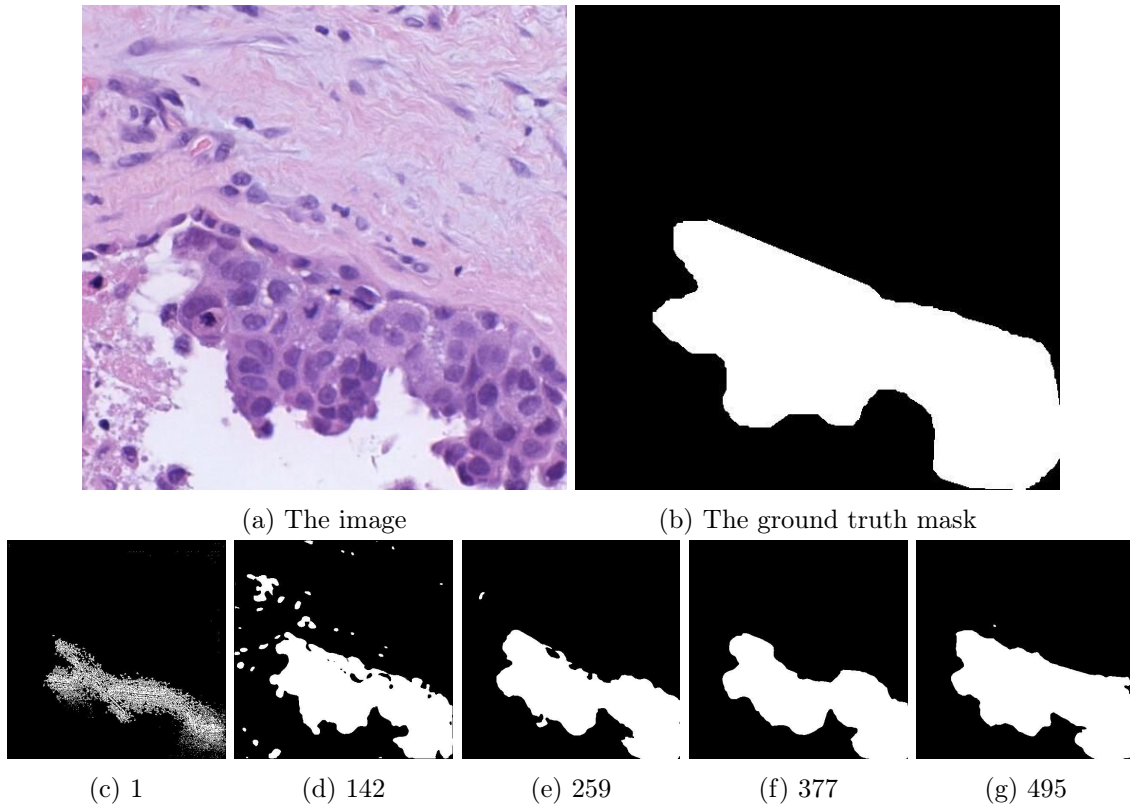


Figure 4.18: A good example of an infiltration segmentation from the CHU-ANAPATH-NST-DL dataset. The number from the (c) to (g) shows the number of annotations used for the training.

Tumour

As can be seen in Figure 4.19, the performance with the IoU metric is really bad. Similar results were observed for the inflammation, for which the cause was class imbalance. However, for the tumours, the cause is completely different. First, notice that the tumour dataset, i.e., CHALLENGE-CAMELYON16, is very different from all the other datasets in the scale of the annotations. The average size of the annotations for the other dataset is around 512×512 in height and width, respectively. Therefore, crops of this size can be extracted from the whole slide images. In contrast to the Camelyon16 dataset, the size of the annotations varies greatly, the annotations can be very small (smaller than 512×512), or very immense. Some examples of size are shown in Figure 4.20. As a consequence of the evaluation phase, all the annotations are resized to 512×512 , in which very large annotations can potentially lose information. About the Hausdorff distance, it seems to be steady as the number of annotations grows, meaning that the number of annotations does not influence the segmentation around the boundaries.

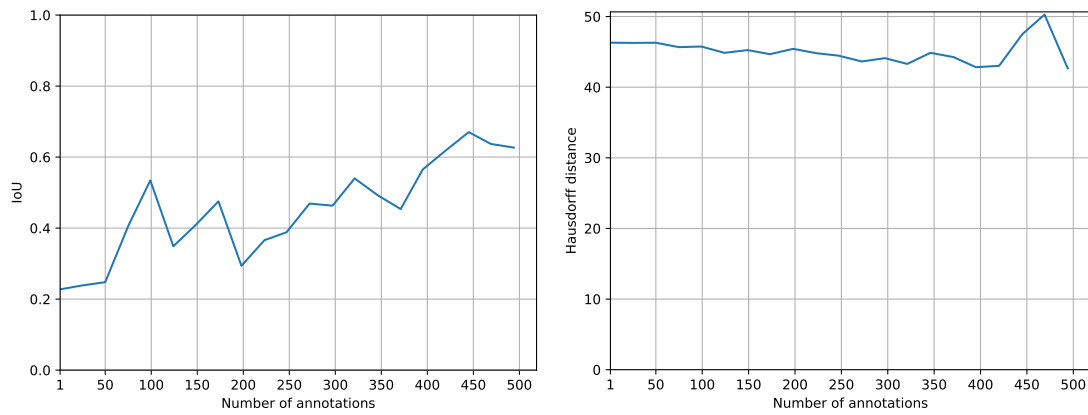


Figure 4.19: Performance for the tumour.

Number of annotations	Average time for one epoch	Total time	Number of annotations	Average time for one epoch	Total time
1	4m00s	20h01m50s	272	6m21s	1d07h42m40s
25	4m16s	21h21m50s	297	6m49s	1d10h5m5s
50	4m30s	22h31m45s	321	6m56s	1d10h42m6s
75	4m30s	22h28m50s	346	6m50s	1d10h9m24s
99	4m53s	1d0h25m16s	371	7m18s	1d12h31m30s
124	5m2s	1d01h7m57s	395	7m44s	1d14h42m22s
149	5m26s	1d03h10m36s	420	7m53s	1d15h25m12s
173	5m27s	1d03h14m43s	445	7m54s	1d15h28m20s
198	5m41s	1d04h27m13s	469	7m28s	1d13h20m49
223	6m5s	1d06h24m20s	494	8m36s	1d18h59m27s
247	5m59s	1d05h53m28s	✗	✗	✗

Table 4.11: Computation time regarding the trainings of the tumour segmentation task.

Looking at the computation time shown in Table 4.11, it is the training that took the longest time even though the number of annotations in the training and validation sets was very similar to the gland and infiltration experiments as seen in Table 4.6, page 43. It is explained by the fact that this dataset is very different from the rest of the datasets used in this thesis. During the training procedure, the target annotations must first be resized to a dimension of 512×512 which takes the major part of the epoch.

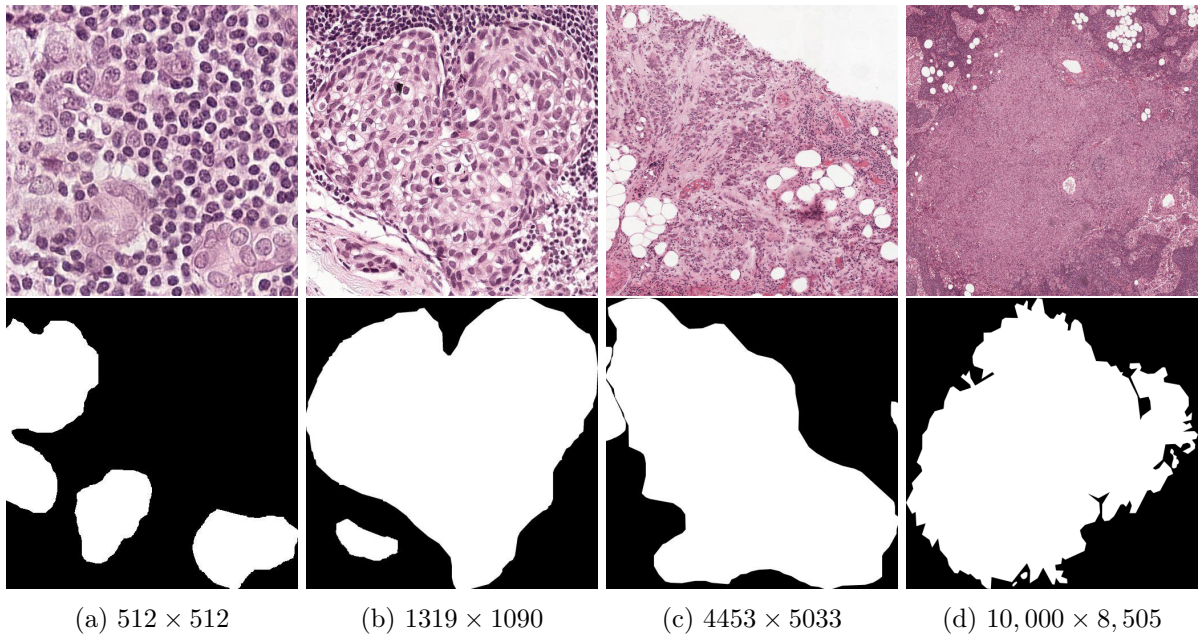


Figure 4.20: Various size of annotations present in the CHALLENGE-CAMELYON16 dataset.

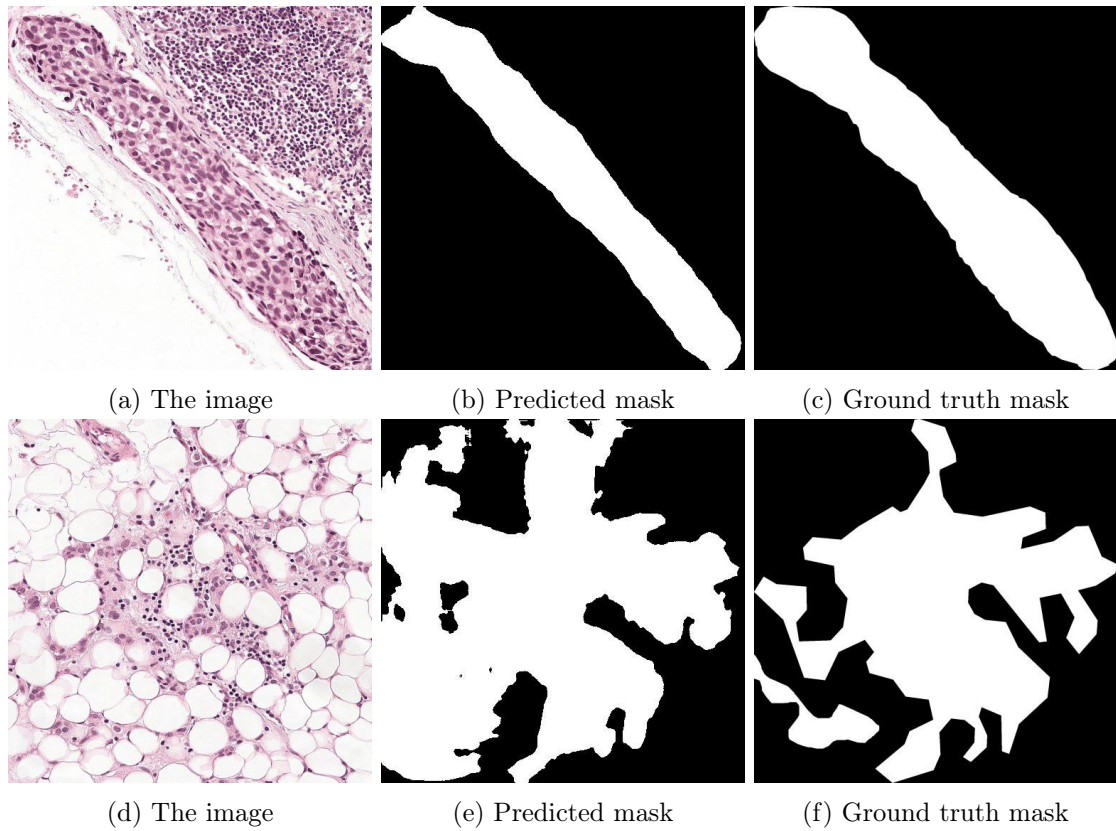


Figure 4.21: Illustration of a good (upper) and a poor (lower) segmentation of tumours from the CHALLENGE-CAMELYON16 dataset.

4.4.2 Quality analysis

This experiment studies the impact of the shape of the scribbles on the segmentation. The aim is to find which type of shapes produces the most accurate segmentation. As stated in section 3.3.1, page 27, performing a large scale experiment with real annotations by a user is unfeasible, mainly due to the time-consuming process of annotating the objects of interest by hand. Reusing the technique explained in section 3.3.1, page 27, for creating the line strokes is not worthwhile, since it will result in biased results, i.e., very high performance. Since the goal of this experiment is to analyse the scribbles, another technique for creating the scribbles is sought with the constraint of having more unexpected scribbles. The algorithm generating the scribble is first going to be described followed by the experiment results analyses. Lastly, simple geometric shapes, namely the circle and the square, are going to be used as scribbles to see the performance that can be obtained.

The developed algorithm to simulate scribbles is based on the algorithm presented in the paper *From A to B, randomly: a point-to-point random trajectory generator for animal movement* [Technitis et al., 2015]. Basically, given a source point and a destination point, the algorithm consists in generating several intermediate points at an equal distance that establishes the trajectory. Some trajectory examples of this algorithm are depicted in Figure 4.22.

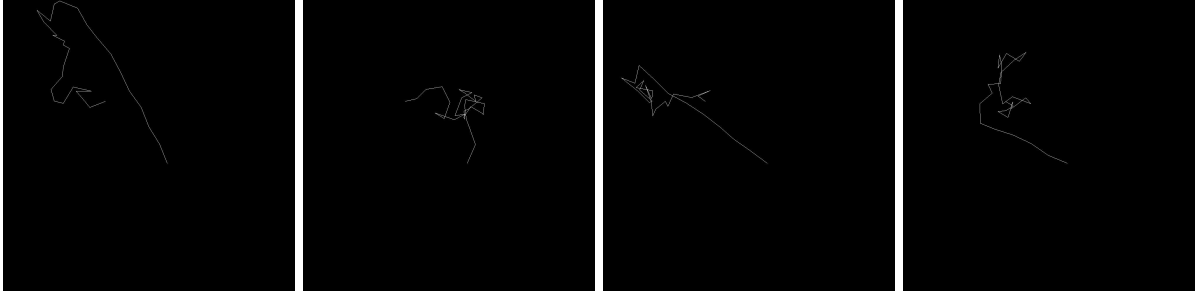


Figure 4.22: Various examples of random trajectory for the same source and destination points.

Starting from this algorithm, a supplementary constraint is put, which is that the random trajectory must lie inside the ground truth mask of the object. Otherwise, the network will predict inaccurate segmentations. The generation of random scribbles algorithm consists of two parts:

- A) Generating the source and destination points at random.
- B) Generating randomly the intermediate points connecting the source to the destination.

An example of scribbles with different numbers of intermediate points generated by the algorithm is shown in Figure 4.23.

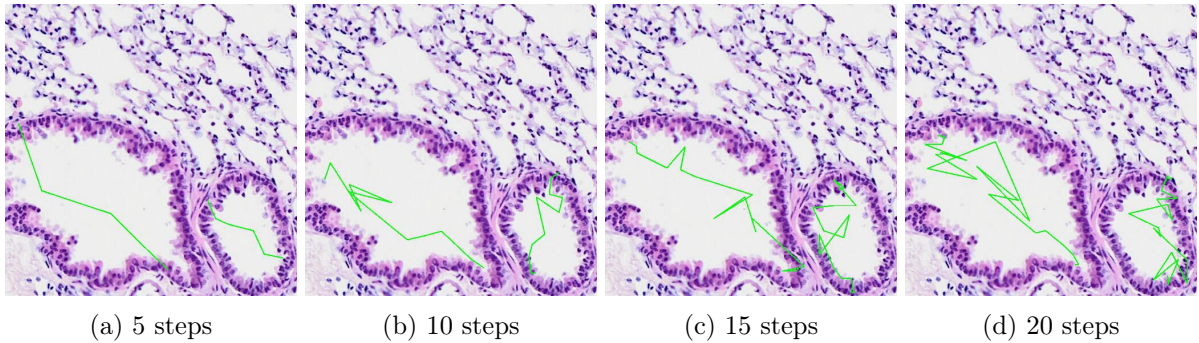


Figure 4.23: Random scribbles with different number of steps on bronchi.

Regarding the first part of the algorithm, the different steps are described with an example:

1. First, the input of the algorithm is the ground truth mask of the objects of interest. Figure 4.24a presents an example of input masks.
2. The contour coordinates of the masks are extracted. This step gives the number of objects present in the ground truth mask image as illustrated in Figure 4.24b. In the example at hand, there are two masks implying that there are two objects to segment.
3. To avoid having the source point close to the destination point, a minimum distance between the two points is calculated. First, the four extreme points from the extracted contour are calculated. The distance from one extreme point to another one is computed using the Euclidean distance for each combination of two points without repetition, i.e, 6 distances in total. The distances are represented by the green lines and the extreme points by the red dots in Figure 4.24c. The minimum distance is the average distance over the 6 computed distances. Using extreme points to compute the distance allows very far source and destination points.
4. Two random points are sampled inside the contour until the following constraints are met:
 - The Euclidean distance between the two points is greater than the minimum distance computed in the previous step.
 - Each of the two points should lie inside the contour of the mask. It is because the x-axis and the y-axis are sampled separately in the implementation of the algorithm. Thus, leading to points that are considered valid on both axes individually but invalid together.

Figure 4.24d depicts the sampled points after satisfying the aforementioned constraints.

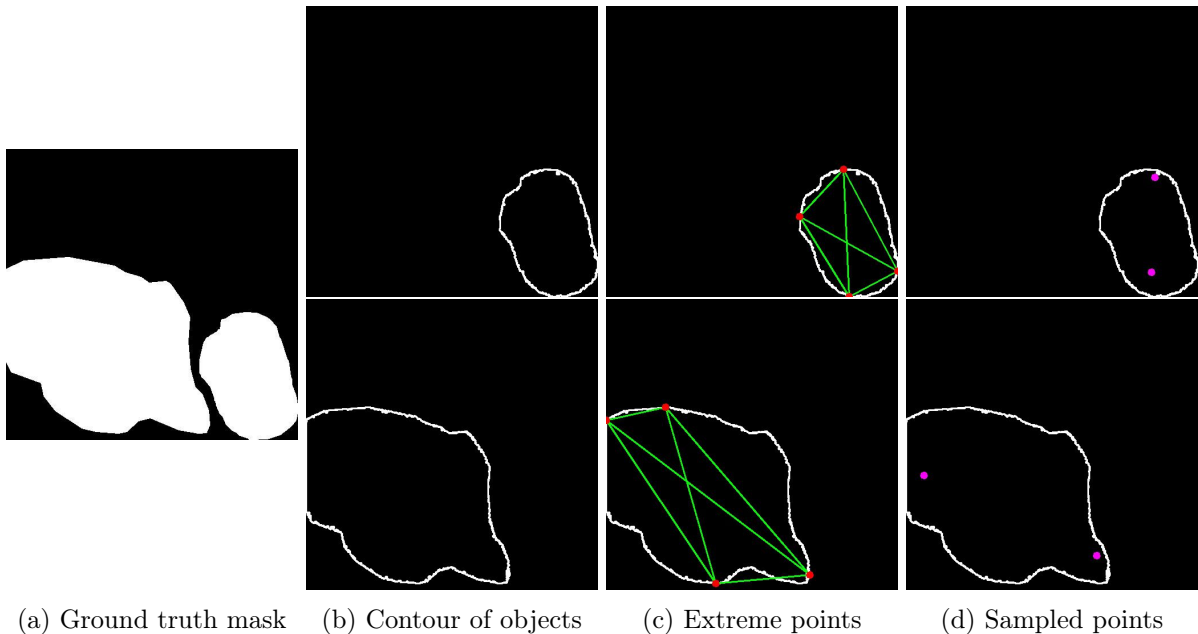


Figure 4.24: Illustration of the random generation of the source and destination points.

Concerning the second part of the algorithm, the different steps are going to be described continuing the example of the first part:

1. The required inputs are the mask of one object, the sampled points of the first part representing the source and destination points, respectively, the length of a line segment, and the number of intermediate points to generate. The length of a line segment m is computed with $\lceil d/s \rceil + 1$ where d is the Euclidean distance between the source and destination points and s the number of intermediate points.
2. For each intermediate point (step),
 - (a) Let the current point be the point of the last intermediate point. With the special case for the first point, where the current point is the source point.
 - (b) The radius is computed for the current point and the destination point by

$$\text{radiusA} = \text{step} * m, \quad \text{radiusB} = (s - \text{step}) * m \quad (4.1)$$

where step is the current step, m is the length of the line segment computed in the previous step, and s is the number of intermediate points. Using the radii, circles are created centred on the current point and the destination point as illustrated in Figure 4.25a.

- (c) The intersection of the two circles and the mask is computed. It is illustrated by the red zone in Figure 4.25b. The next intermediate point is sampled from this region uniformly at random.

Figure 4.25c depicted the repetition of step 2 for two consecutive steps.

3. The last step is to connect the source and destination points by linking the intermediate points as shown in Figure 4.25d. The generated trajectory forms the scribble that is going to be used for inclusion and exclusion maps.

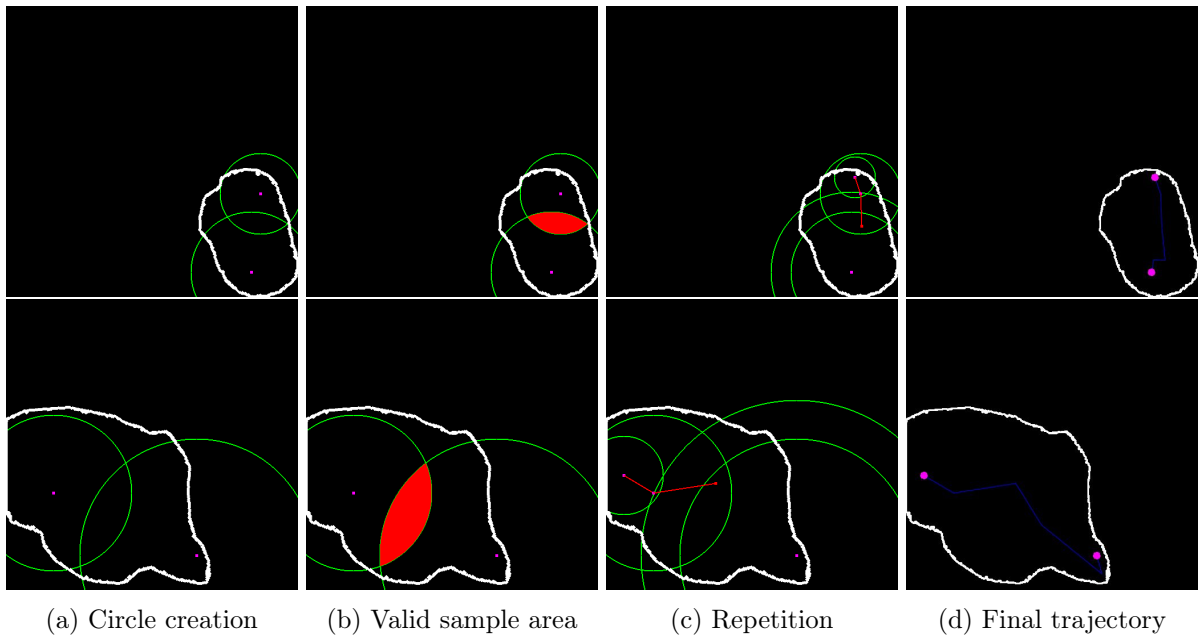


Figure 4.25: Random generation of the trajectory from the source point to the destination point.

Performing the generation of the intermediate points in the order produces a phenomenon named heavily drifted walk [Technitis et al., 2015]. Basically, for the first few intermediate points, the algorithm samples points at random in any possible direction. However, for the remaining ones, the algorithm begins to *hurry* towards the destination points because it has generated points far from the destination point. To respect the constraints, the algorithm thus generates nearly straight lines toward the destination. This phenomenon can be seen in Figure 4.22. To cope with this issue, the solution is to generate the intermediate points at random. The only modification to the algorithm is the step 2. (a). As the previous generated intermediate point might not exist, the current point is simply the source point. Otherwise, it is the previously generated intermediate point. Illustrations with an increasing number of intermediate points are depicted in Figure 4.23. Supplementary illustrations for each type of object are shown in the appendix section C.1, page 90.

For the following experiments, the number of intermediate points (steps) tested is 5, 10, 15, and 20. More steps sizes do not illustrate a random scribble that is realistic anymore. For each step, three different variations of the random squiggle are used. The average performance of the three is then taken.

Bronchus

As can be seen in Figure 4.26, more complex scribbles on bronchi do not improve significantly, but slightly the quality of the segmentation, i.e., the performance in term of IoU. The Hausdorff distance is very huge in comparison with Figure 4.6, page 45, of the quantity experiment. After the examination of the predicted segmentations, a few of them present an open hole more or less located at the centre as illustrated in Figure 4.27b, which generate a high Hausdorff distance value. Regarding good segmentation of bronchi, an illustration is shown in Figure 4.28.

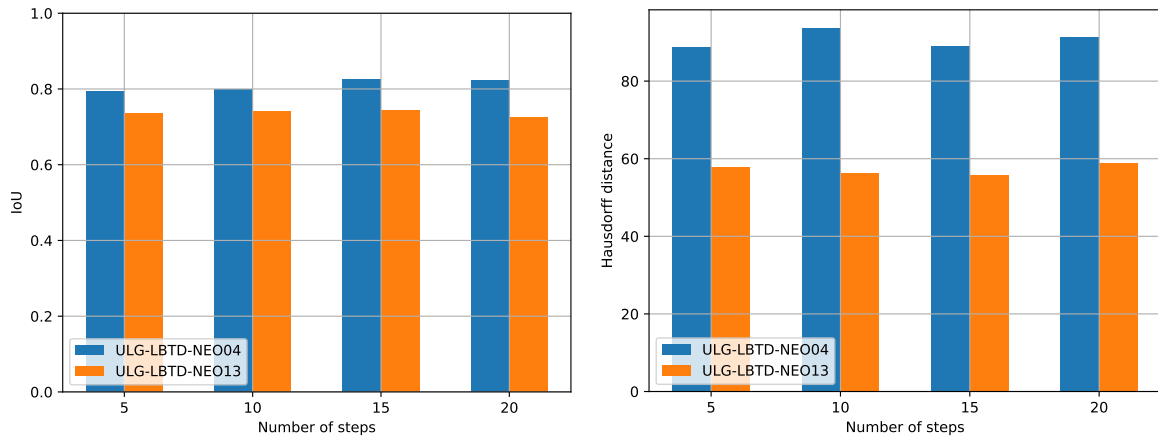


Figure 4.26: Performance of the model on bronchi. On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.

Gland

Comparing the IoU of Figure 4.9, page 47, at about 510 annotations with the one of Figure 4.29, a surprising increase of performance happen for images in the CHALLENGE-GLAS-2015 peeking their IoU at 0.8, which is very good. Regarding the Hausdorff distance, it is very high for the CHALLENGE-GLAS-2015 for the same reason as the bronchus. Nothing special about the performance on the CHU-ANAPATH-NST-DL, for which the performance is comparable to the one in the quantity analysis. An illustration of a good segmentation from the CHALLENGE-GLAS-2015 dataset is shown in Figure 4.30.

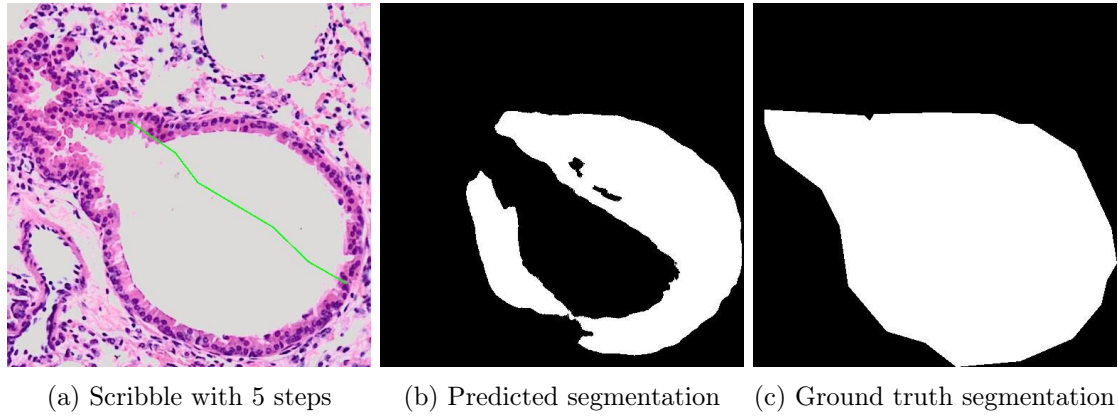


Figure 4.27: Example of a bronchus segmentation generating a high Hausdorff distance value, i.e., 169. The image comes from the ULG-LBTD-NEO04 dataset.

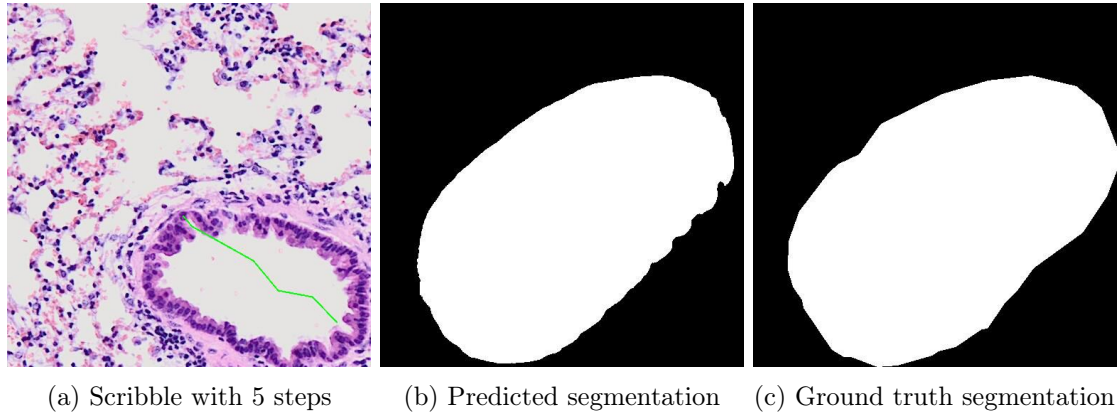


Figure 4.28: Example of a good bronchus segmentation from the ULG-LBTD-NEO04 dataset.

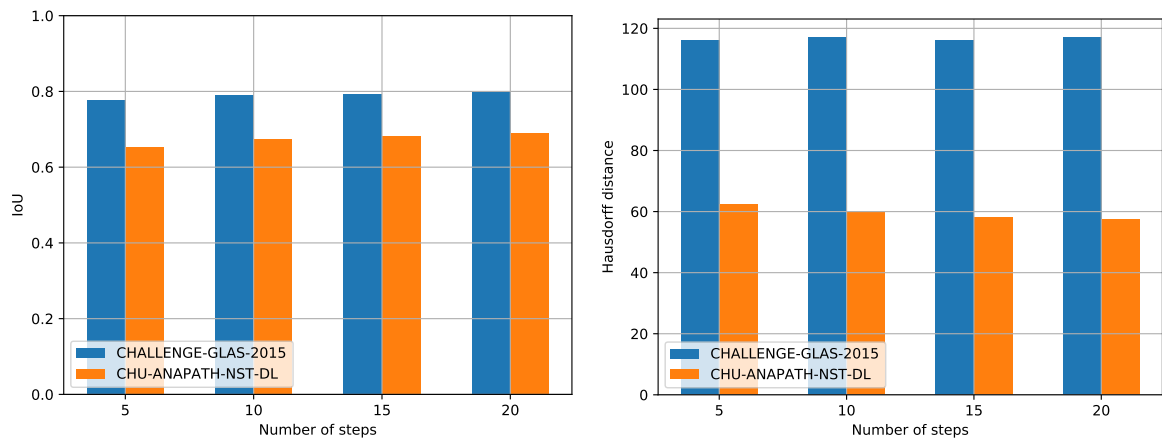


Figure 4.29: Performance of the model on glands. On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.

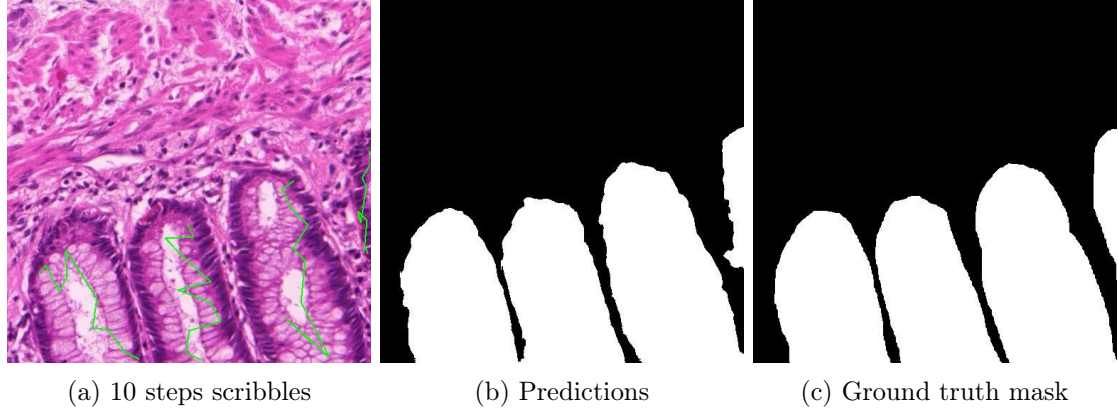


Figure 4.30: Example of a good gland segmentation from the CHALLENGE-GLAS-2015 dataset.

Inflammation

Since the performance on inflammation is bad as shown in the quantity analysis, scribbles covering more ground of the inflammation improve very slightly the performances as shown in Figure 4.31. A seldom good segmentation of inflammation is shown in Figure 4.32 and the frequently encountered poor segmentation is shown in Figure 4.33.

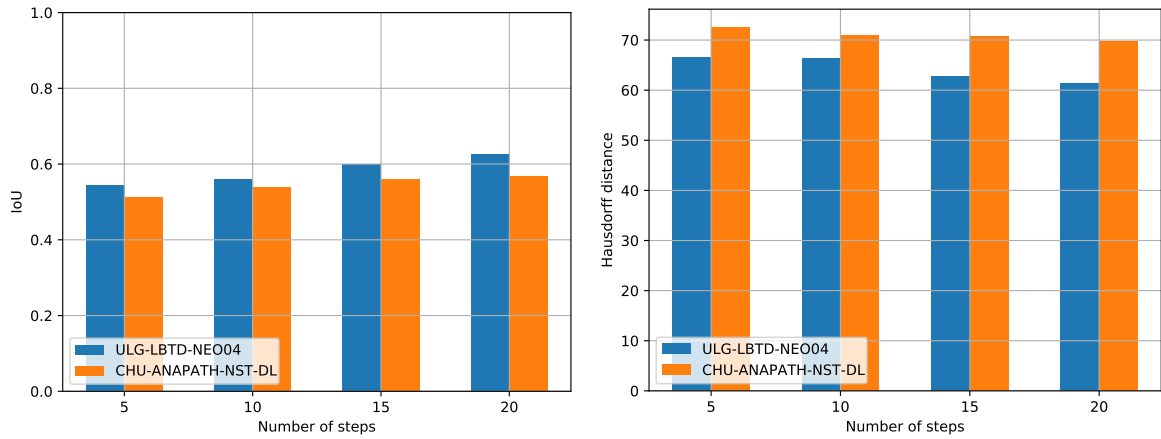


Figure 4.31: Performance of the model on inflammations. On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.

Infiltration and tumour

The same conclusion as inflammation can be made for the tumour experiment, where the performance is shown in Figure 4.34 in the orange bar. More precisely, scribbles that cover more ground of the tumour improve slightly the segmentation. An interesting observation can be made for the infiltration. Comparing the IoU of Figure 4.34, i.e., the blue bar plot, with the one of Figure 4.16, page 52, a substantial decrease of about 0.2 IoU can be seen. After the inspection of the predicted segmentations, shown in Figure 4.35, some area of the infiltration is not predicted as expected leaving a medium gap in the segmentation. It happens for a considerable number of images containing infiltration, thus leading to the decrease of performance. As a consequence of the gap from the segmentation, the Hausdorff distance is naturally larger. A simple solution to improve the segmentation is to scribble more sensible regions, i.e., regions which were not predicted as part of the segmentation on the first try. Naturally, this solution would waste some time because of the redundancy annotation.

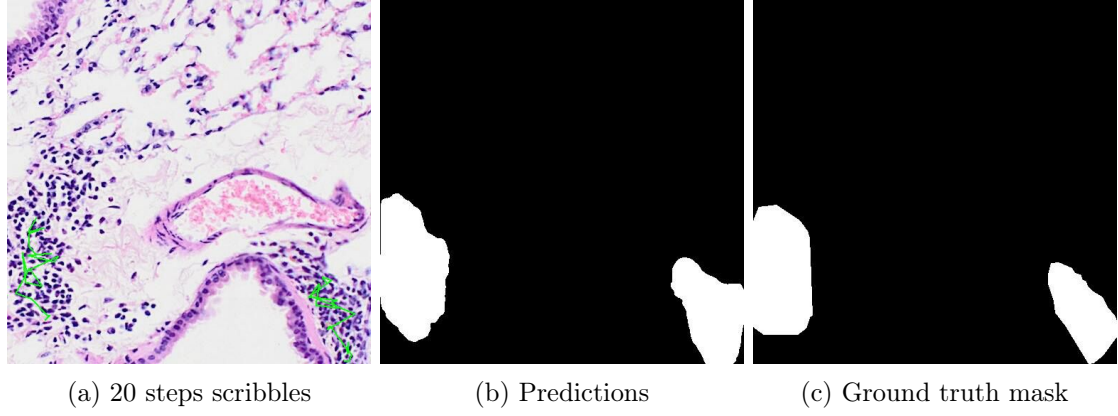


Figure 4.32: Example of a good inflammation segmentation from the ULG-LBTD-NEO04 dataset.

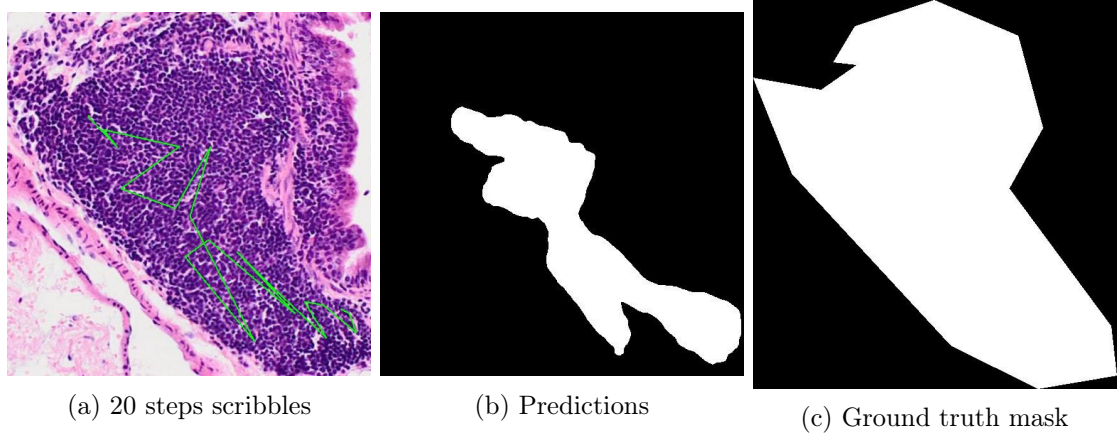


Figure 4.33: Example of a poor inflammation segmentation from the CHU-ANAPATH-NST-DL dataset.

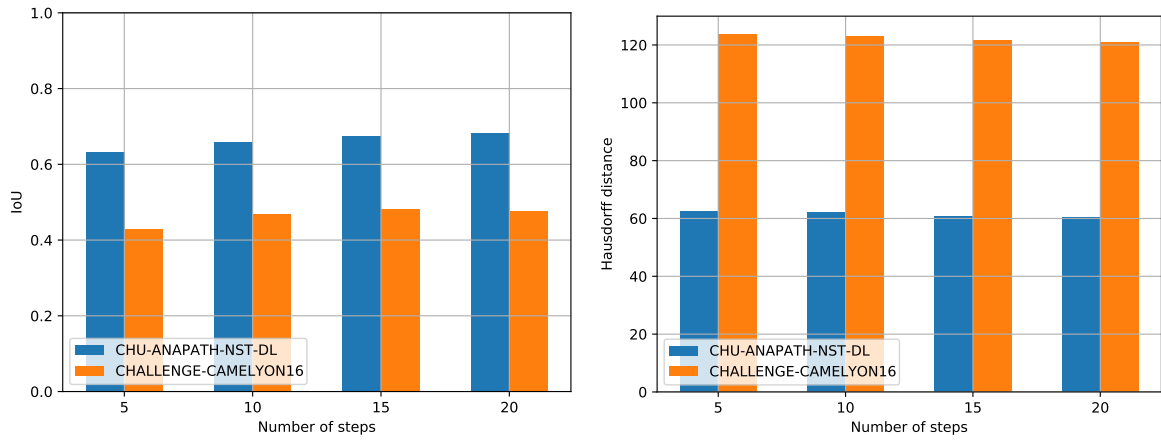


Figure 4.34: Performance of the model on infiltrations (left) and tumours (right). On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.

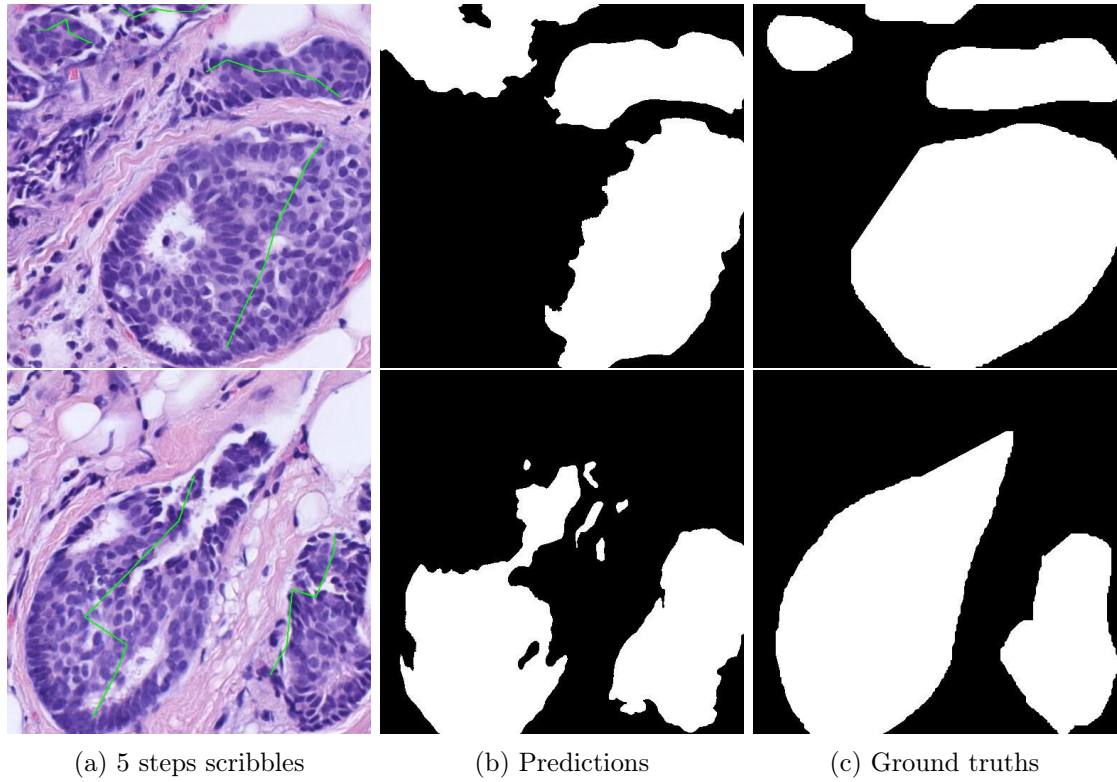


Figure 4.35: Example of mediocre segmentations of infiltration from the CHU-ANAPATH-NST-DL dataset.

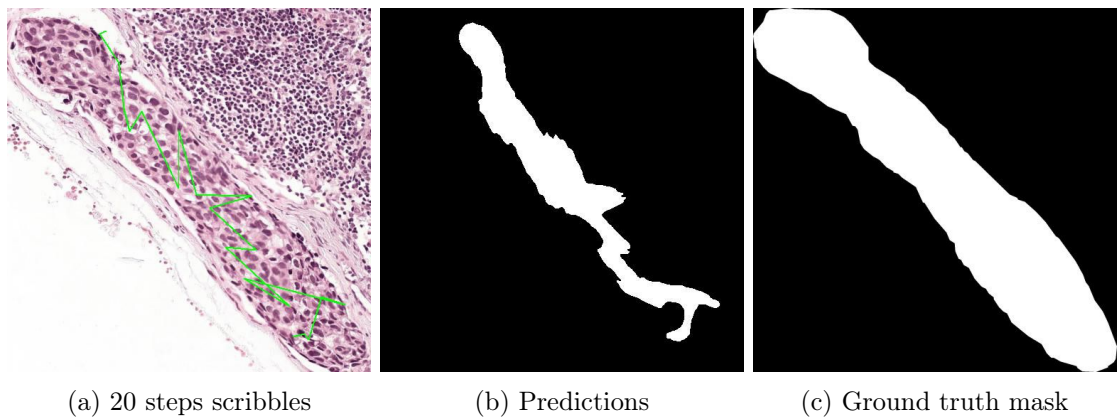


Figure 4.36: Example of a poor tumour segmentation from the CHALLENGE-CAMELYON16 dataset.

Simple geometric shape as scribble

In this small experiment, circles and squares are used as scribbles to see what performance can be achieved with these geometric shapes instead of simple line strokes. More precisely, the experiment consists in generating a circle (resp. square) inside the ground truth mask that is going to be used to build the inclusion and exclusion maps. The experiment is first done by generating circles and then by generating squares. The evaluation is done for both shapes together resulting in only one plot (Figure 4.38). Illustrations of generated circles and squares are shown in Figure 4.37.

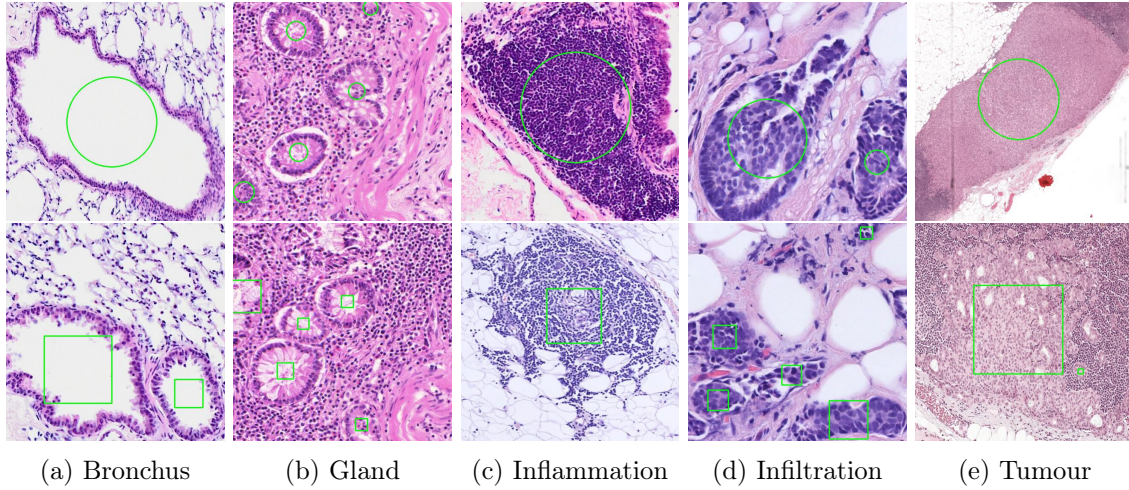


Figure 4.37: Illustrations of circles and squares inside the objects of interest.

Figure 4.38 reports the performance achieved by the models using geometric shapes as scribbles. As a general observation, the performance of the various models seems to achieve at most an IoU of 0.6, which are lower than the values observed with the random scribbles experiment. More precisely, using shapes as scribbles drastically decreases the performance on the bronchus of the dataset ULG-LBTD-NEO13. Regarding the Hausdorff distance, the highest values are achieved by the bronchus from the ULG-LBTD-NEO04 dataset, gland from the CHALLENGE-GLAS-2015 dataset, and tumour from the CHALLENGE-CAMELYON16 dataset, despite the fact that the IoU of these models varies greatly. The predicted segmentations of the illustrations in Figure 4.37 are shown in Figure 4.39.

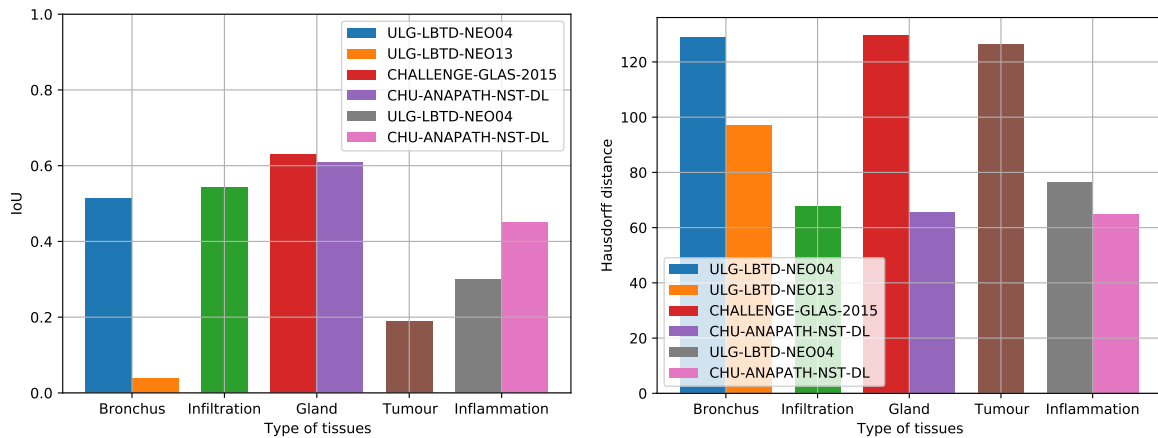


Figure 4.38: Performance of the model on the different tissues. The legend shows the dataset used for the training and evaluation of the model. The dataset used for infiltration and tumour is CHU-ANAPATH-NST-DL and CHALLENGE-CAMELYON16, respectively.

From the predicted segmentations, it can be observed that the models segmenting bronchus, inflammation, and tumour try to predict the shape of the scribbles instead of the object of interest. However, for gland and infiltration, the models seem to achieve mediocre segmentations, which confirms the performance displayed in Figure 4.38.

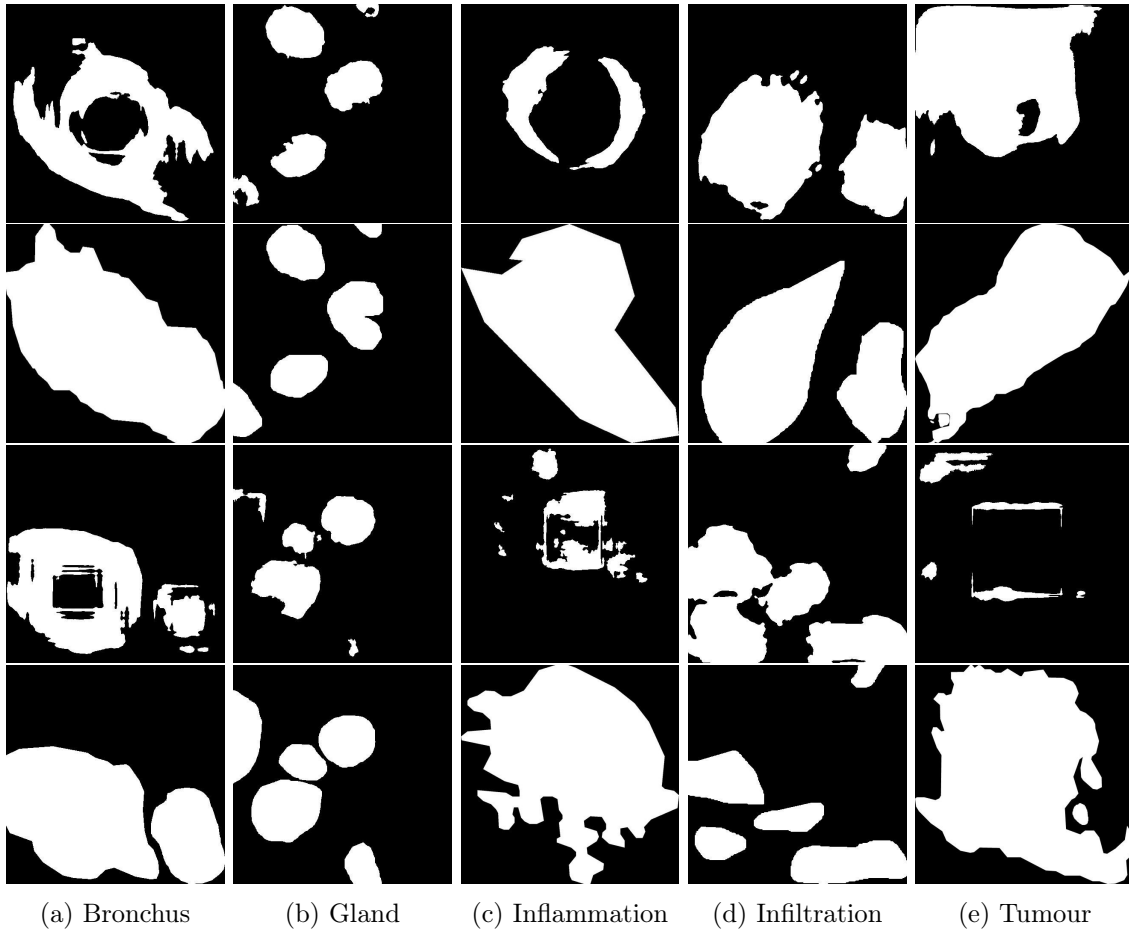


Figure 4.39: Illustrations of the segmentation from the images presented in Figure 4.37.

4.5 Robustness analysis

This section aims at determining the robustness of the NuClick model. In short, the experiment consists in training the model on one of the datasets presented in section 4.1, page 35, and evaluating the trained model on another dataset. From Table 4.1, page 35, the experiment can be done for bronchi, inflammations, and glands only, since there are two different datasets for each of the mentioned tissues. The legend presented in the subsequent figures shows the name of the dataset used for training the model. The abscissa shows the name of the testing set of the dataset.

4.5.1 Bronchus

As can be seen in Figure 4.40, the performance are more or less the same when testing the models on the datasets. It was the expected results since the two datasets were very similar. The only noticeable difference is that the model trained on the ULG-LBTD-NEO04 training set and evaluated on the ULG-LBTD-NEO04 testing set, produces a larger Hausdorff distance, as was the case for all the previous experiments on this dataset as well.

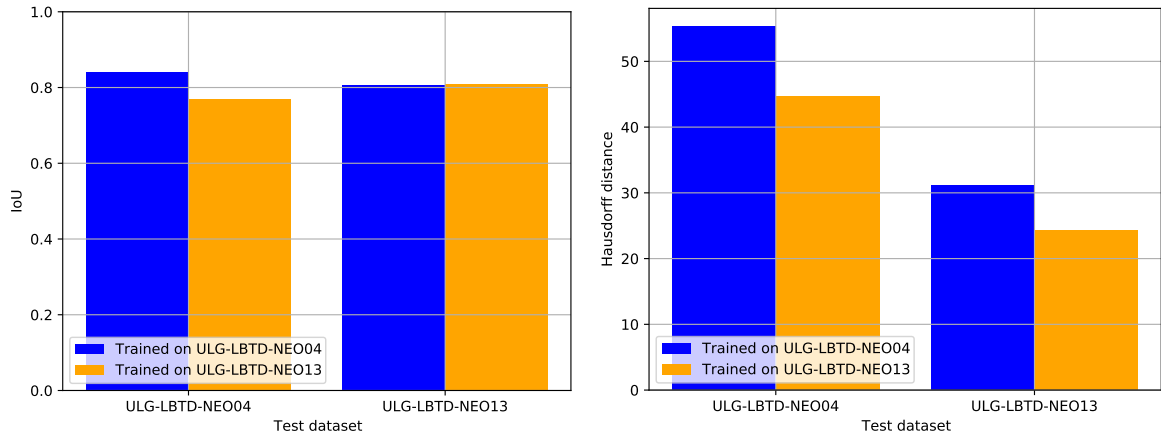


Figure 4.40: Performance of the models on bronchi. On the left is shown the performance of the IoU and on the right of Hausdorff distance.

4.5.2 Gland

Notice the poor performance from the evaluation of the model trained on CHALLENGE-GLAS-2015 training set on the CHU-ANAPATH-NST-DL testing set, shown in Figure 4.41. On the opposite, the model trained on the CHU-ANAPATH-NST-DL dataset seems to be more robust. After the investigation of both datasets, the images containing gland are very dissimilar as depicted in Figure 4.42. It explains why the model trained on CHALLENGE-GLAS-2015 has more difficulties segmenting the images from the CHU-ANAPATH-NST-DL dataset.

4.5.3 Inflammation

Regarding the inflammation, the models achieve about equal performance for both datasets as illustrated by Figure 4.43. Although the two datasets come from different laboratory and the provenance of the slides come from different tissues, i.e., lung tissues for the ULG-LBTD-NEO04 and breast tissues for CHU-ANAPATH-NST-DL, the models are quite robust in the variation of the images.

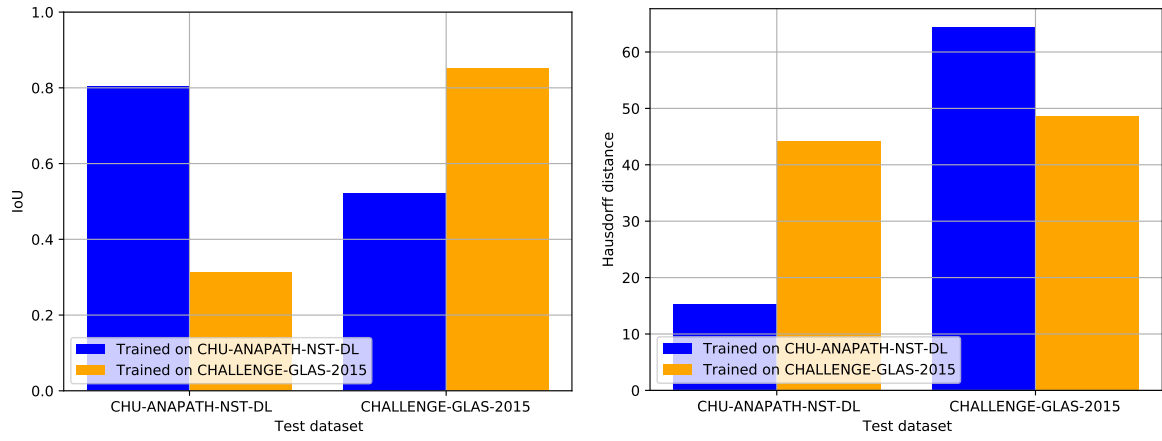


Figure 4.41: Performance of the models on glands.

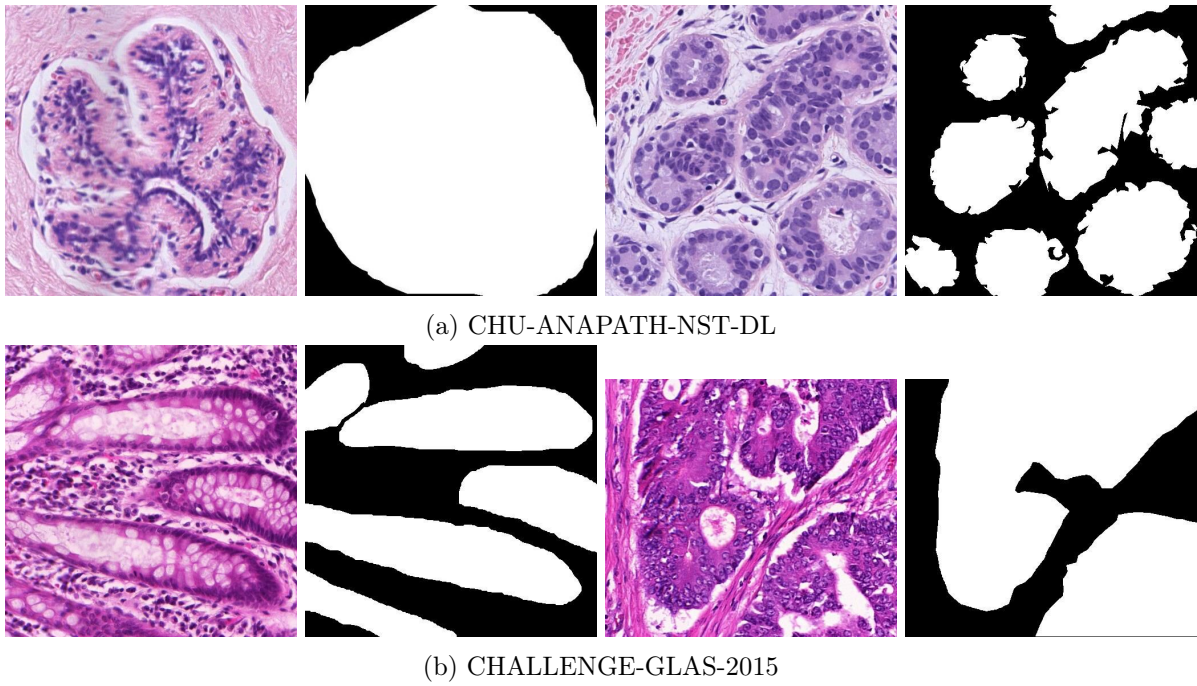


Figure 4.42: Illustrations of glands with their ground truth mask from the CHU-ANAPATH-NST-DL and CHALLENGE-GLAS-2015 datasets.

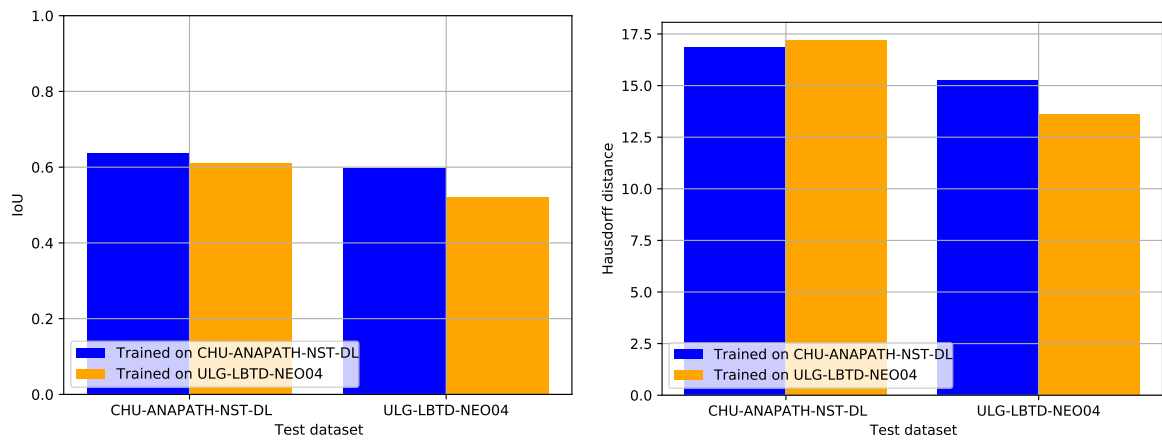


Figure 4.43: Performance of the models on inflammations.

4.6 Model analysis

This section focuses on various aspects of the neural network, NuClick. The first experiment aims at comparing the performance of the NuClick architecture with the well known U-Net architecture presented in subsection 2.1.6, page 8. The second and third experiments try to determine the implication of the inclusion and exclusion map. The results of these experiments are going to be analysed jointly.

4.6.1 U-Net comparison

This experiment aims at comparing the U-Net architecture with the NuClick architecture. To train U-Net, the classical approach of training a deep learning model is used, which is presented in subsection 2.1.3, page 5. For a comparison focused on the model architecture only, the parameters used are the one stated in section 4.3, page 43.

4.6.2 Absence of signal maps

This second experiment seeks at determining the implication of the signal provided in the inclusion and exclusion maps. The goal is to compare the NuClick model with and without these supplementary signal. The experiment consists in training the NuClick architecture with empty inclusion and exclusion maps, i.e., the fourth and fifth channels are always black. In the figures presented in the result section, this model is referred to as *Black NuClick*.

4.6.3 Automatic NuClick architecture

NuClick was introduced with the notion of inclusion and exclusion map explained in subsection 3.3.1. This experiment tries to answer the question of whether these signal maps are indeed useful for the segmentation or not. To train this architecture, it uses the classical approach as explained earlier for U-Net, which is presented in subsection 2.1.3, page 5. This architecture do not take the inclusion and exclusion maps as supplementary channels, thus the input to this network is only $512 \times 512 \times 3$. This model is referred to as *Simple NuClick* in the results section.

4.6.4 Results

For the following figures, the four architectures are presented with their performance on the test sets. The first one is the NuClick presented section 3.3, page 26, followed by the simple NuClick, black NuClick, and U-Net. The legend of the figures reports the dataset used for the testing set.

Bronchus

As can be seen in Figure 4.44, the different models achieve very similar performance in IoU on the ULG-LBTD-NEO04 dataset. However, on the ULG-LBTD-NEO13, the performance seems to decrease as the complexity of the model decreases. Furthermore, as the IoU decreases, the Hausdorff distance increases, meaning that the predicted segmentations are similar to Figure 4.35, page 63.

Gland

Regarding the gland, Figure 4.45 reports the performance of the two datasets containing glands. It can be seen from this figure that the performance of the various models is quite similar. The NuClick seems to perform the best followed by the model in the order ending with U-Net having the smallest performance. Notice that the Hausdorff distance is high for all the models except for NuClick, which shows the importance of the inclusion and exclusion maps.

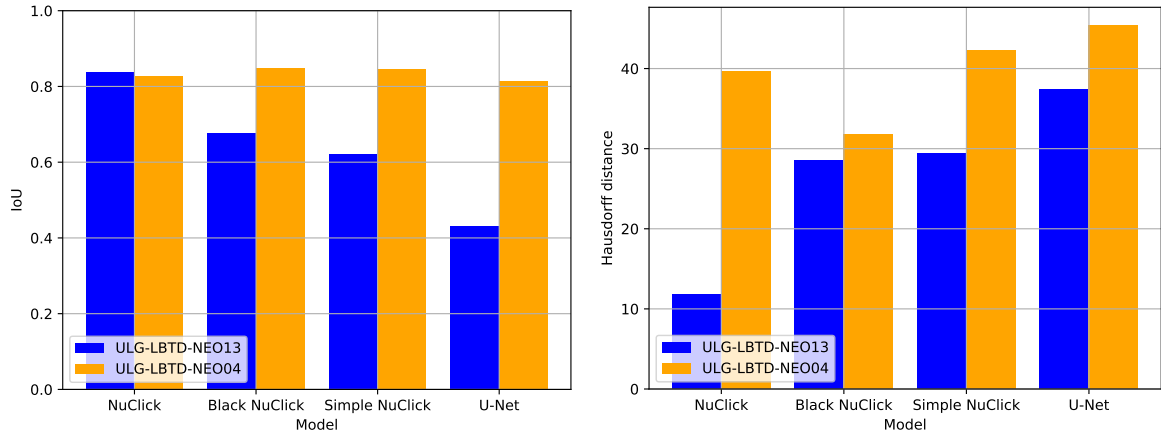


Figure 4.44: Performance for the bronchus. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.

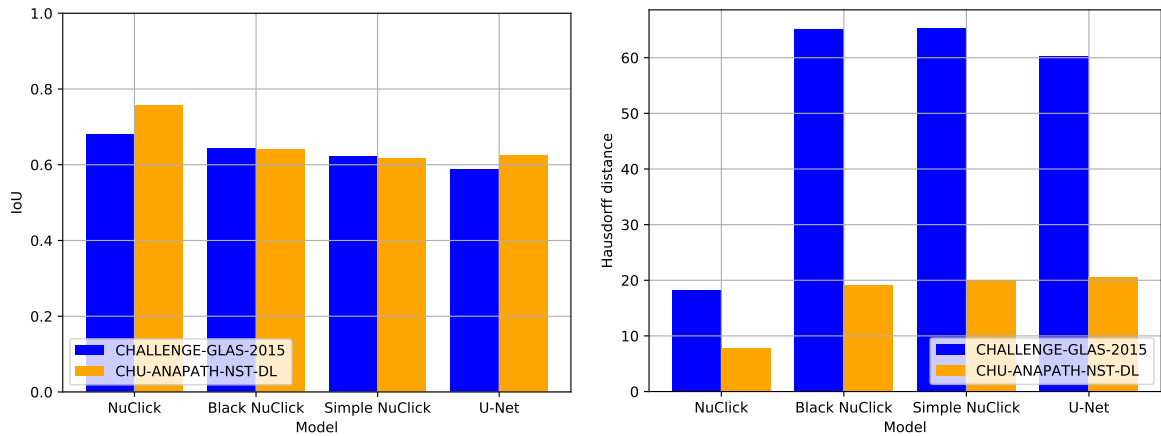


Figure 4.45: Performance for the gland. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.

Infiltration

For bronchus and gland, the IoU decreases in the order of the model. However, for infiltration, the *Simple NuClick* model seems to achieve a better performance than *Black NuClick* and U-Net. Again, U-Net seems to achieve the worst performance among the four models.

Inflammation

A first observation that can be drawn from Figure 4.47 jointly with Figure 4.44, is that the ULG-LBTD-NEO04 dataset seems to contain very robust annotations so that the performances obtained are comparable for similar neural network architecture. In contrast to the infiltration, the *Simple NuClick* achieves the worse performance on the CHU-ANAPATH-NST-DL dataset.

Tumour

Analogous to the inflammation, *Simple NuClick* seems to perform the worse among the four architectures as shown by the performance with the IoU in Figure 4.48. However, it can be noticed that the U-Net architecture manages to perform better than *Black NuClick* and *Simple NuClick*. Regarding the Hausdorff distance, the values are close to each other even though the performance with the IoU varies considerably.

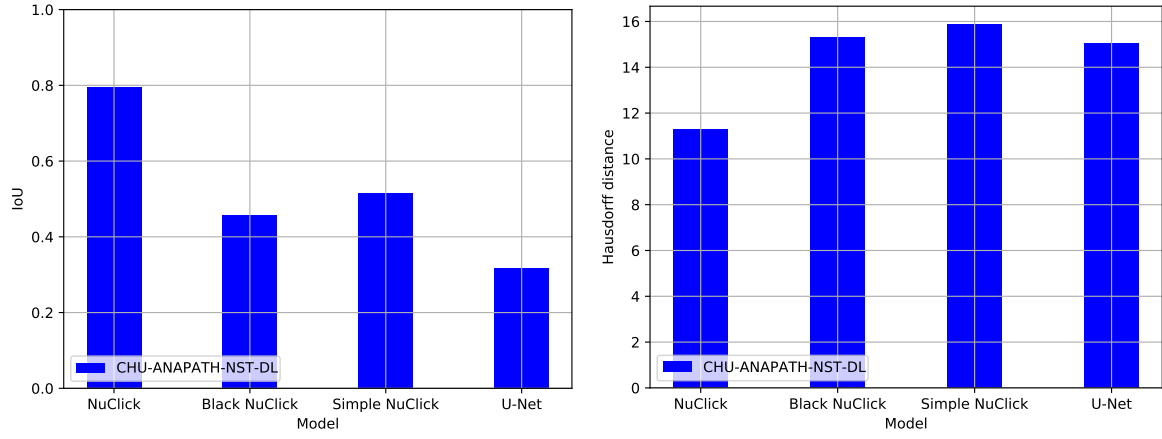


Figure 4.46: Performance for the infiltration. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.

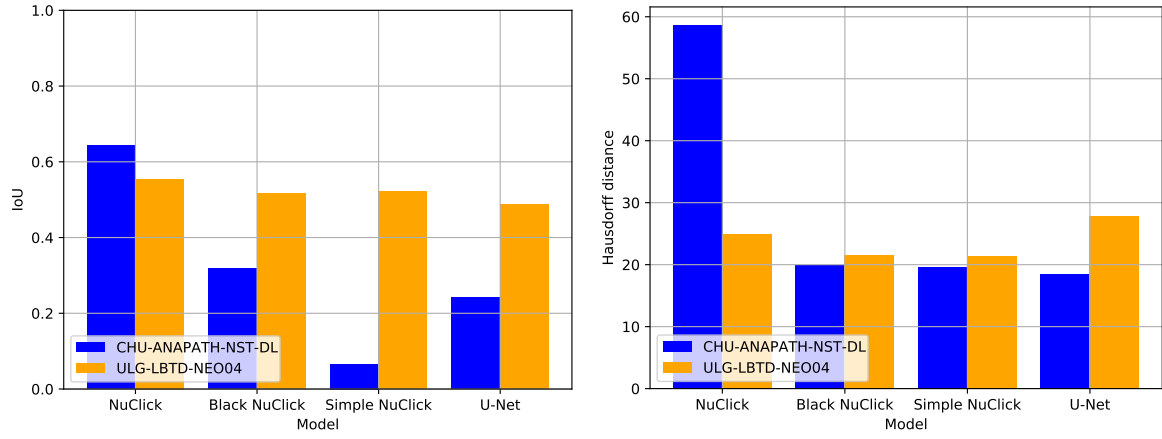


Figure 4.47: Performance for the inflammation. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.

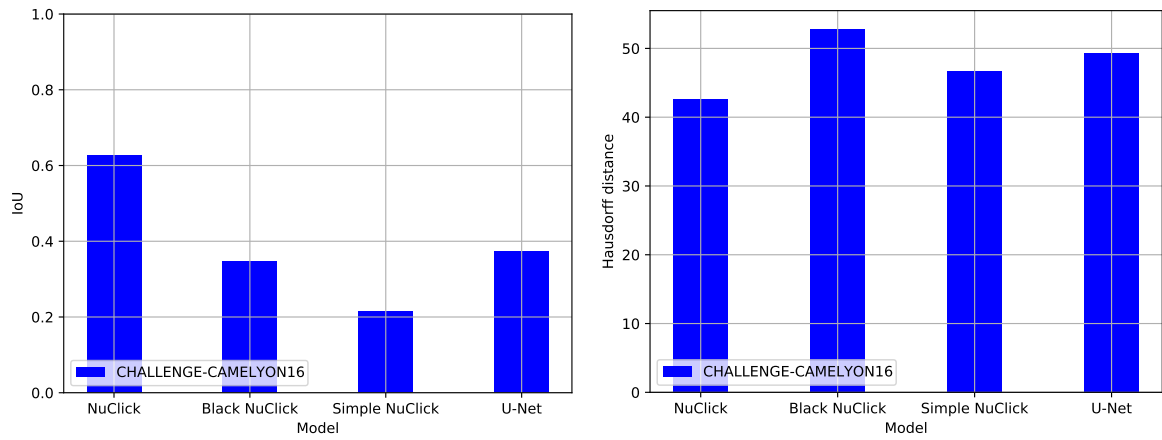


Figure 4.48: Performance for the tumour. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.

4.7 Discussion

Before discussing the results of the experiments, a small discussion about the impact of the values set in the post-processing presented in subsection 3.3.3, page 30, is done. First, for the object removal, specifying a too large threshold could lead to the removal of small mask objects as shown in Figure 4.14, page 50. Therefore, with the considerations of these small masks, the final threshold was set at 100 pixels. Intuitively, for the holes filling, one might set the area threshold to be filled at the highest possible value to fill holes. An example of a hole that could have been filled with a higher area threshold is shown in Figure 4.39a, page 65. Unfortunately, some ground truth masks contain holes as shown in Figure 4.49. Therefore, the final value was set to 300 to avoid filling correct the holes.

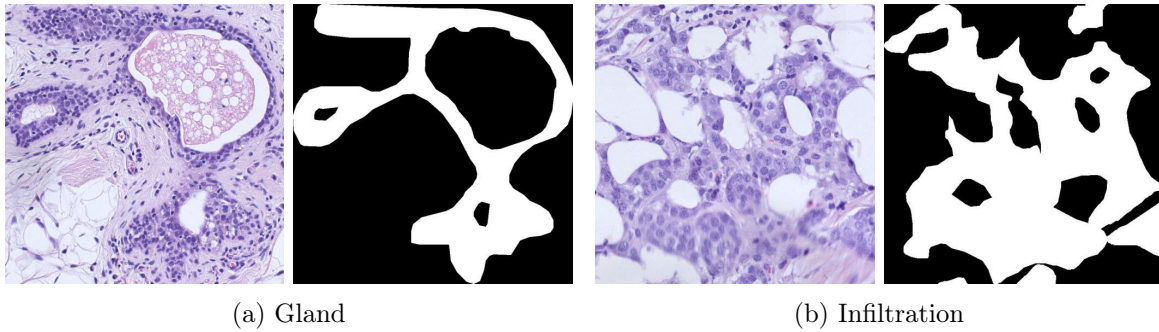


Figure 4.49: Holes in ground truth masks of a gland and an infiltration, respectively.

The very first experiment was the replication of the original experiment on glands by NuClick [Alemi Koohbanani et al., 2020]. Our reimplementaton of the NuClick model was trained on the training set of the GlaS challenge 2015 [Sirinukunwattana et al., 2015, 2016] using the same hyperparameters as the original paper. Comparing our results with the original results, they seem to be lower than the original performance in term of the Dice similarity coefficient. However, our version of NuClick achieves better performance in Hausdorff distance in both testing sets of GlaS. Naturally, the next step was to use their implementation to replicate their experiment with their own code. However, the creation of the dataset in their implementation uses a specific protocol that is not explained in their paper nor in the code. A minor issue was that their code uses MATLAB, for which a license is needed. Therefore, it was impossible to use their code for training the model.

Then, the second experiment was about the determination of the minimum number of annotations needed to have satisfactory results. For bronchi, glands, and infiltrations, about 120 annotations were needed to produce an intersection over the union between 0.7 and 0.8, which is considered to be good performance. However, regardless of the number of annotations, the model seems to struggle to segment inflammation and tumour annotations. Regarding the inflammations, it is mainly because of class imbalance, i.e., the annotations cover less than a quarter of the ground truth mask image. A proposed solution is to train the network taking a much smaller input size, i.e., $128 \times 128 \times 5$. With images of a height and width of 128×128 , the class imbalance is greatly reduced. As for the tumours, the problem seems to come from the fact that the annotations are too diversified in shape and size, e.g., some annotations could have a dimension of $10,000 \times 8,505$. This result suggests that the approach proposed in this thesis is not suitable for this kind of annotations and that other approaches or techniques that can deal with very large annotations might be a better solution.

After the quantity analysis, a quality analysis was performed on the quality of the scribbles done by annotators. Employing real annotators to scribble each object of interest in the test set is not feasible. Therefore, an algorithm generating random scribbles was designed to cope with this issue. This study demonstrates that indeed the shape of the scribbles impacts the predicted

segmentation. Scribbles that cover more ground of the object of interest tend to produce more accurate segmentation. However, scribbles that look like geometric shapes, in particular a circle or a square, are not recommended. This analysis supports the fact that the models tend to produce segmentation that look like the geometric shapes. The general guidelines for a scribble are that it should cover more ground of the object of interest with simple line strokes.

The last experiment aims at determining the impact of the additional information given by the interactions of the annotators in form of scribbles. The results indicate that these interactions indeed bring an added value to the network, producing better and more accurate segmentation than without these interactions.

Chapter 5

Conclusion and perspectives

This thesis explores various semi-automatic approaches to speed up the annotation process in biomedical tissue images, which is known to be time-consuming.

First, a review of the literature was made, which was focused on semi-automatic or interactive approaches about segmentation tasks. These approaches use the interactions of the annotators in various forms, giving rise to methods such as click-based methods, where clicks are incorporated as supplementary signals in the input of the models, etc.

One promising architecture, named NuClick, was chosen from this review, because of its high performance, mainly. A reimplementation was made and evaluated on several datasets. These datasets were acquired from Cytomine and contained various type of tissues. In particular, bronchi, glands, inflammations, infiltrations, and tumours were used in the evaluation from the various datasets.

Based on the quantitative analysis, it has been shown that about 100 of complete annotations are needed for training a model that can achieve satisfactory segmentation, more precisely for bronchi, glands, and infiltrations. However, this analysis also showed that not all type of tissues trained on the aforementioned quantity naturally produce decent segmentation. Indeed, more complex tissues, such as inflammations or tumours, failed to achieve tolerable segmentation.

To analyse the impact of the shape of scribbles, an algorithm was designed to mimic the scribbles of a human annotator. By analysing results based on this algorithm, this thesis has shown that indeed the shape of the scribbles has a substantial impact on the predicted segmentation. Based on this conclusion, annotators should consider scribbles that cover more ground of the object of interest rather than a simple line stroke.

Lastly, this thesis has shown that using the interactions of the annotator as supplementary information for the models tend to outperform automatic models in the task of segmentation of biomedical images.

5.1 Perspectives

Further investigations in the literature could have been done to find other methods that could potentially achieve better performance globally than the NuClick architecture.

5.1.1 Improvements

Various improvements can be done in this thesis. The first and most important one is the specific fine-tuning to each of the explored type of tissues. More specifically, to perform a complete cross-validation search of the hyperparameters to achieve the best performance and a specific fine-tuning of the value chosen for the post-processing.

Another improvement is to further investigate the cause for the performance issues caused by the inflammations and tumours in the quantity and quality analysis and also to implement

the solution to these issues.

Next, other approaches could have been explored in this thesis rather than using only one to conduct the experiments. It can be useful, in the case that other approaches can, for instance, produce better performance on the inflammation and tumour.

5.1.2 Integration to Cytomine

A future work concerns the Cytomine web interface. Currently, a user can initiate algorithms to perform segmentation over the desired object. However, this process is very static, in the sense that once the user launches the experiment, no further interaction can be done. Thus, repeating the process each time a parameter has changed is not ideal. To this end, integrating an algorithm with the interactions of the user could be done in Cytomine. The inference time was measured to see if this work can be used for a real-time use case. It takes about 1-2 minutes to have the network segment one gland image of the GlaS 2015 dataset on CPU and about 1 second on a GPU (tested on a personal computer with a GeForce MX150 GPU).

To integrate the approach developed in this thesis, Cytomine provides a protocol that uses various kinds of technologies, such as JavaScript, Docker container, and many more. The protocol to integrate an application to Cytomine is available at [Cytomine apps documentation](#).

List of Figures

1.1	Illustration of a whole slide image of a sentinel lymph node shown in the Cytomine web interface.	1
1.2	Example of the segmentation of a bronchus, where the foreground corresponds to the bronchus and the background to all the other pixels in the image.	2
1.3	The type of tissues used in this thesis with their corresponding segmentation mask.	4
2.1	A multilayer perceptron with 3 layers of neurons.	6
2.2	Residual block (Source: [He et al., 2015]).	8
2.3	U-Net neural network architecture (Source: [Ronneberger et al., 2015]).	8
2.4	Complete pipeline of their software (Source: [Lee et al., 2021]).	10
2.5	Complete pipeline of the framework (Source: [Ho et al., 2020]).	11
2.6	Deep Multi-Magnification Network architecture (Source: [Ho et al., 2021]).	12
2.7	P-Net neural network architecture (Source: [Wang et al., 2019]).	12
2.8	NuClick neural network architecture (Source: [Alemi Koohbanani et al., 2020]).	13
2.9	The FCNN architecture (Source: [Sakinis et al., 2019]).	14
2.10	The architecture of the network (Source: [Kitrungrotsakul et al., 2020]).	15
2.11	Complete pipeline of the framework (Source: [Wang et al., 2017]).	16
2.12	Examples using the Click Carving approach (Source: [Jain and Grauman, 2019]).	17
2.13	Illustration of the initial segmentation produced by Curve-GCN (Source: [Ling et al., 2019]).	18
2.14	Illustration of an segmentation produced by the Deep Snake approach (Source [Peng et al., 2020]).	19
3.1	Simplified overview of the methodology.	22
3.2	Illustration of the creation of the dataset from the raw downloaded Cytomine data. This process is done for each whole slide image in the dataset.	23
3.3	Illustration of the training procedure.	24
3.4	Illustration of the testing procedure.	24
3.5	NuClick neural network architecture.	26
3.6	Creation process of the signal map.	27
3.7	Signal map with different threshold value τ	28
3.8	Inclusion and exclusion map.	28
3.9	Scribble on the objects of interest, i.e., glands, done by a user.	28
3.10	Example of the post-processing for an noisy prediction and a prediction with a hole.	30
3.11	The intersection over the union (IoU).	31
3.12	Dice coefficient.	32
3.13	Hausdorff distance, where the dashed line is the predicted mask and the full line is the ground truth mask (Source: [Karimi and Salcudean, 2019]).	33
4.1	Example of whole slide images in the Camelyon16 dataset. The tumour regions can be seen in red.	37
4.2	Example of images in the GlaS2015 dataset.	38

4.3	Example of a whole slide image at different magnifications in the CHU-ANAPATH-NST-DL dataset (Source: NEW_201708241740.tif). Infiltration can be seen in light green, In situ in dark green, and artefact in blue.	39
4.4	Example of a whole slide image at different magnifications in the ULG-LBTD-NEO04 dataset (Source: NEO4_CURCU_INH_8.20_01.tif). Bronchus annotations can be seen in purple, inflammation in yellow, and tumour in burgundy. . .	40
4.5	Example of a whole slide image at different magnifications in the ULG-LBTD-NEO13 (3) dataset (Source: NEO13_CNS_1.30_5_3_01.tif). In burgundy is shown tumour regions, bronchus in purple, and inflammation in yellow.	41
4.6	Performance for the bronchus. On the left is the performance using the intersection over the union metric and on the right is the Hausdorff distance.	45
4.7	Example of a good segmentation of bronchus. The number from the subfigure (c) to (g) shows the number of annotations used for the training.	45
4.8	Poor bronchus segmentation generating a high Hausdorff distance value, i.e., 150.	46
4.9	Performance for the gland. On the left is the performance using the intersection over the union metric and on the right is the Hausdorff distance.	47
4.10	Example of a good segmentation of gland. The number from the subfigure (c) to (g) shows the number of annotations used for the training.	48
4.11	Poor gland segmentation from the CHU-ANAPATH-NST-DL dataset.	48
4.12	Performance for the inflammation. On the left is the performance using the intersection over the union metric and on the right is the Hausdorff distance.	49
4.13	Images with their corresponding ground truth mask of inflammatory regions in the ULG-LBTD-NEO04 dataset.	50
4.14	Images with their corresponding ground truth mask of inflammatory regions in the CHU-ANAPATH-NST-DL dataset.	50
4.15	Illustration of a mediocre segmentation (upper) from the CHU-ANAPATH-NST-DL dataset and a poor segmentation (lower) of inflammation from the ULG-LBTD-NEO04.	51
4.16	Performance for the infiltration.	52
4.17	Illustrations of problematic images of infiltration from the CHU-ANAPATH-NST-DL dataset. An partially annotated image is shown on the left and several small regions of mask is shown on the right	52
4.18	A good example of an infiltration segmentation from the CHU-ANAPATH-NST-DL dataset. The number from the (c) to (g) shows the number of annotations used for the training.	53
4.19	Performance for the tumour.	54
4.20	Various size of annotations present in the CHALLENGE-CAMELYON16 dataset.	55
4.21	Illustration of a good (upper) and a poor (lower) segmentation of tumours from the CHALLENGE-CAMELYON16 dataset.	55
4.22	Various examples of random trajectory for the same source and destination points.	56
4.23	Random scribbles with different number of steps on bronchi.	56
4.24	Illustration of the random generation of the source and destination points.	57
4.25	Random generation of the trajectory from the source point to the destination point.	58
4.26	Performance of the model on bronchi. On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.	59
4.27	Example of a bronchus segmentation generating a high Hausdorff distance value, i.e., 169. The image comes from the ULG-LBTD-NEO04 dataset.	60
4.28	Example of a good bronchus segmentation from the ULG-LBTD-NEO04 dataset.	60
4.29	Performance of the model on glands. On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.	60
4.30	Example of a good gland segmentation from the CHALLENGE-GLAS-2015 dataset.	61

4.31	Performance of the model on inflammations. On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.	61
4.32	Example of a good inflammation segmentation from the ULG-LBTD-NEO04 dataset.	62
4.33	Example of a poor inflammation segmentation from the CHU-ANAPATH-NST-DL dataset.	62
4.34	Performance of the model on infiltrations (left) and tumours (right). On the left, the performance is expressed with IoU and on the right with the Hausdorff distance.	62
4.35	Example of mediocre segmentations of infiltration from the CHU-ANAPATH-NST-DL dataset.	63
4.36	Example of a poor tumour segmentation from the CHALLENGE-CAMELYON16 dataset.	63
4.37	Illustrations of circles and squares inside the objects of interest.	64
4.38	Performance of the model on the different tissues. The legend shows the dataset used for the training and evaluation of the model. The dataset used for infiltration and tumour is CHU-ANAPATH-NST-DL and CHALLENGE-CAMELYON16, respectively.	64
4.39	Illustrations of the segmentation from the images presented in Figure 4.37.	65
4.40	Performance of the models on bronchi. On the left is shown the performance of the IoU and on the right of Hausdorff distance.	66
4.41	Performance of the models on glands.	67
4.42	Illustrations of glands with their ground truth mask from the CHU-ANAPATH-NST-DL and CHALLENGE-GLAS-2015 datasets.	67
4.43	Performance of the models on inflammations.	67
4.44	Performance for the bronchus. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.	69
4.45	Performance for the gland. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.	69
4.46	Performance for the infiltration. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.	70
4.47	Performance for the inflammation. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.	70
4.48	Performance for the tumour. On the left, can be seen the performance measured in IoU and on the right in Hausdorff distance.	70
4.49	Holes in ground truth masks of a gland and an infiltration, respectively.	71
C.1	Scribbles with 5 intermediate steps on bronchi from the ULG-LBTD-NEO04 dataset.	90
C.2	Scribbles with 10 intermediate steps on glands from the CHALLENGE-GLAS2015 dataset.	91
C.3	Scribbles with 5 intermediate steps on infiltrations from the CHU-ANAPATH-NST-DL dataset.	92
C.4	Scribbles with 15 intermediate steps on tumours from the CHALLENGE-CAMELYON16 dataset.	92
C.5	Scribbles with 20 intermediate steps on tumours from the CHALLENGE-CAMELYON16 dataset.	93

List of Tables

2.1	The performance stated by each of the reviewed approaches.	20
2.2	Selection of the architecture based on the criteria.	21
3.1	The list of GPUs from the Alan cluster used during the thesis.	34
3.2	Summary of hyperparameters and other parameters used in this thesis.	34
3.3	Hyperparameters tested in a cross-validation search.	34
4.1	Summary table for the types and datasets used. It shows the number of annotations for each type of object.	35
4.2	The smallest to the largest dimension of the whole slide images in each dataset.	35
4.3	Parameters used by the original NuClick.	42
4.4	The split used by the original NuClick.	42
4.5	Performance comparison between the stated result from the NuClick paper and the reimplemented version.	42
4.6	The split of annotations for each type of object in the various datasets.	43
4.7	Computation time regarding the trainings of the bronchus segmentation task.	46
4.8	Computation time regarding the trainings of the gland segmentation task.	47
4.9	Computation time regarding the trainings of the inflammation segmentation task.	51
4.10	Computation time regarding the trainings of the infiltration segmentation task.	53
4.11	Computation time regarding the trainings of the tumour segmentation task.	54
B.1	The split of annotations for each type of object in the various datasets.	82
B.2	Complete results of the experiment for the bronchus of the ULG-LBTD-NEO04 dataset.	83
B.3	Complete results of the experiment for the bronchus of the ULG-LBTD-NEO13 (3) dataset.	83
B.4	Time taken for the training using the CHALLENGE-GLAS-2015 dataset.	84
B.5	Table containing the full results of the experiment for the gland in the CHALLENGE-GLAS-2015 dataset.	85
B.6	Table containing the full results of the experiment for the gland in the CHU dataset.	86
B.7	Table containing the full results of the experiment for the inflammation in the NEO04 dataset.	87
B.8	Table containing the full results of the experiment for the inflammation in the CHU dataset.	87
B.9	Table containing the full results of the experiment for the infiltration in the CHU dataset.	88
B.10	Table containing the full results of the experiment for the infiltration in the CHALLENGE-CAMELYON16 dataset.	89
C.1	Complete results of the quality analysis with the random scribble generation algorithm.	94

Bibliography

- Alemi Koohbanani, N., Jahanifar, M., Zamani Tajadin, N., and Rajpoot, N. (2020). Nuclick: A deep learning framework for interactive segmentation of microscopy images. *arXiv e-prints*, page arXiv:2005.14511.
- Cytomine Corporation SA (2021). Cytomine official website. <https://cytomine.be/>.
- Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J. A. W. M., and the CAMELYON16 Consortium (2017). Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22):2199–2210.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.
- Ho, D. J., Agaram, N. P., Schüffler, P. J., Vanderbilt, C. M., Jean, M.-H., Hameed, M. R., and Fuchs, T. J. (2020). Deep interactive learning: An efficient labeling approach for deep learning-based osteosarcoma treatment response assessment. *Lecture Notes in Computer Science*, page 540–549.
- Ho, D. J., Yarlagadda, D. V., D’Alfonso, T. M., Hanna, M. G., Grabenstetter, A., Ntiamoah, P., Brogi, E., Tan, L. K., and Fuchs, T. J. (2021). Deep multi-magnification networks for multi-class breast cancer image segmentation. *Computerized Medical Imaging and Graphics*, 88:101866.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.
- Jahanifar, M., Alemi Koohbanani, N., and Rajpoot, N. (2019). Nuclick: From clicks in the nuclei to nuclear boundaries. *arXiv e-prints*, page arXiv:1909.03253.
- Jain, S. D. and Grauman, K. (2019). Click carving: Interactive object segmentation in images and videos with point clicks. *International Journal of Computer Vision*, 127(9):1321–1344.
- Karimi, D. and Salcudean, S. E. (2019). Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. *arXiv e-prints*, page arXiv:1904.10030.
- Kitrungrotsakul, T., Yutaro, I., Lin, L., Tong, R., Li, J., and Chen, Y.-W. (2020). Interactive deep refinement network for medical image segmentation. *arXiv e-prints*, page arXiv:2006.15320.
- Lee, S., Amgad, M., Mobadersany, P., McCormick, M., Pollack, B. P., Elfandy, H., Hussein, H., Gutman, D. A., and Cooper, L. A. (2021). Interactive classification of whole-slide imaging data for cancer researchers. *Cancer Research*, 81(4):1171–1177.
- Ling, H., Gao, J., Kar, A., Chen, W., and Fidler, S. (2019). Fast interactive object annotation with curve-gcn. *CoRR*, abs/1903.06874.

- Marée, R., Rollus, L., Stévens, B., Hoyoux, R., Louppe, G., Vandaele, R., Begon, J.-M., Kainz, P., Geurts, P., and Wehenkel, L. (2016). Collaborative analysis of multi-gigapixel imaging data using cytomine. *Bioinformatics*, 32(9):1395–1401.
- Marée, R., Rollus, L., Stévens, B., Louppe, G., Caubo, O., Rocks, N., Bekaert, S., Cataldo, D., and Wehenkel, L. (2014). A hybrid human-computer approach for large-scale image-based measurements using web services and machine learning. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 902–906.
- Peng, S., Jiang, W., Pi, H., Bao, H., and Zhou, X. (2020). Deep snake for real-time instance segmentation. *CoRR*, abs/2001.01629.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597.
- Sakinis, T., Milletari, F., Roth, H., Korfiatis, P., Kostandy, P. M., Philbrick, K., Akkus, Z., Xu, Z., Xu, D., and Erickson, B. J. (2019). Interactive segmentation of medical images through fully convolutional neural networks. *CoRR*, abs/1903.08205.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv e-prints*, page arXiv:1409.1556.
- Sirinukunwattana, K., Pluim, J. P. W., Chen, H., Qi, X., Heng, P., Guo, Y. B., Wang, L. Y., Matuszewski, B. J., Bruni, E., Sanchez, U., Böhm, A., Ronneberger, O., Cheikh, B. B., Racocceanu, D., Kainz, P., Pfeiffer, M., Urschler, M., Snead, D. R. J., and Rajpoot, N. M. (2016). Gland segmentation in colon histology images: The glas challenge contest. *CoRR*, abs/1603.00275.
- Sirinukunwattana, K., Snead, D., and Rajpoot, N. (2015). A stochastic polygons model for glandular structures in colon histology images. *IEEE transactions on medical imaging*, 34.
- Technitis, G., Othman, W., Safi, K., and Weibel, R. (2015). From a to b, randomly: a point-to-point random trajectory generator for animal movement. *International Journal of Geographical Information Science*, 29(6):912–934.
- Wang, G., Li, W., Zuluaga, M. A., Pratt, R., Patel, P. A., Aertsen, M., Doel, T., David, A. L., Deprest, J., Ourselin, S., and Vercauteren, T. (2017). Interactive medical image segmentation using deep learning with image-specific fine-tuning. *CoRR*, abs/1710.04043.
- Wang, G., Zuluaga, M. A., Li, W., Pratt, R., Patel, P. A., Aertsen, M., Doel, T., David, A. L., Deprest, J., Ourselin, S., and et al. (2019). Deepigeos: A deep interactive geodesic framework for medical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1559–1572.

Appendix A

Neural network architecture

This appendix presents the number of parameters for the NuClick architecture and the U-Net architecture using the *torchsummary* package. The complete architecture of the two networks can be seen in the Jupyter Notebook at <https://github.com/bathienle/master-thesis-code/blob/master/summary.ipynb>.

A.1 NuClick architecture

```
-----  
Total params: 68,342,785  
Trainable params: 68,342,785  
Non-trainable params: 0  
-----  
Input size (MB): 5.00  
Forward/backward pass size (MB): 3821.00  
Params size (MB): 260.71  
Estimated Total Size (MB): 4086.71  
-----
```

A.2 U-Net architecture

```
-----  
Total params: 21,603,265  
Trainable params: 21,603,265  
Non-trainable params: 0  
-----  
Input size (MB): 3.00  
Forward/backward pass size (MB): 3208.00  
Params size (MB): 82.41  
Estimated Total Size (MB): 3293.41  
-----
```

Appendix B

Quantity analysis

Table B.1 reports the full split of the various dataset according to subsection 3.2.4 page 25 using the three-way data splits.

Dataset	Type	Train set	Val set	Test set
ULG-LBTD-NEO04	Bronchus	242	65	72
ULG-LBTD-NEO13	Bronchus	262	73	74
CHALLENGE-GLAS-2015	Gland	984	250	304
CHU-ANAPATH-NST-DL	Gland	4,011	1,018	1,239
ULG-LBTD-NEO04	Inflammation	94	24	30
CHU-ANAPATH-NST-DL	Inflammation	226	59	69
CHU-ANAPATH-NST-DL	Infiltration	1,813	471	549
CHALLENGE-CAMELYON16	Tumour	1,629	418	498

Table B.1: The split of annotations for each type of object in the various datasets.

B.1 Bronchus experiments

Number of annotations	IoU	Dice	Hausdorff
1	0.450195419788361	0.620631468296051	40.4021713256836
25	0.783478963375092	0.880696892738342	40.1981197357178
49	0.684497678279877	0.807057416439056	42.4123653411865
73	0.611747014522553	0.761756527423859	42.6862163543701
97	0.830332601070404	0.913344609737396	24.0297668457031
121	0.813278567790985	0.901434767246246	35.4426918029785
145	0.863190460205078	0.9245640873909	27.9164554595947
169	0.869396018981933	0.932712769508362	23.6371686935425
193	0.847817671298981	0.925952005386353	24.1715850830078
217	0.767217004299164	0.872367668151856	37.2206298828125
241	0.827474808692932	0.915436720848084	39.6291393280029

Table B.2: Complete results of the experiment for the bronchus of the ULG-LBTD-NEO04 dataset.

Number of annotations	IoU	Dice	Hausdorff
1	0.365927964448929	0.548777383565903	15.7947153091431
27	0.706108379364014	0.819836068153381	18.9540369033813
53	0.732633340358734	0.836392688751221	14.6795478820801
79	0.777828633785248	0.873605859279632	14.274057674408
105	0.742463326454163	0.85052455663681	13.9565808296204
131	0.703098905086517	0.821806871891022	14.6795478820801
157	0.812654042243957	0.897537195682526	12.78844871521
183	0.763413882255554	0.864408791065216	14.8795478820801
209	0.791922736167908	0.879161930084229	14.4983114242554
235	0.793628573417664	0.885142374038696	13.5280077934265
261	0.837925553321838	0.913293182849884	11.8304124832153

Table B.3: Complete results of the experiment for the bronchus of the ULG-LBTD-NEO13 (3) dataset.

B.2 Gland experiments

Number of annotations	Average time for one epoch	Total time	Number of annotations	Average time for one epoch	Total time
1	1m53s	9h24m43s	517	5m36s	1d03h57m34s
25	2m7s	10h34m17s	541	5m59s	1d05h55m41s
50	2m17s	11h27m20s	566	6m10s	1d06h49m28s
74	2m30s	12h28m47s	590	6m23s	1d07h52m40s
99	2m47s	13h54m07s	615	6m22s	1d07h47m48s
123	2m41s	13h25m28s	639	6m44s	1d09h41m57s
148	2m59s	14h53m40s	664	6m54s	1d10h28m56s
173	3m8s	15h40m42s	689	7m6s	1d11h29m28s
197	3m19s	16h33m46s	713	7m3s	1d11h16m10s
222	3m28s	17h21m58s	738	7m33s	1d13h44m21s
246	3m39s	18h15m00s	762	7m43s	1d14h34m28s
271	3m49s	19h5m36s	787	7m57s	1d15h43m01s
295	3m48s	19h1m56s	811	8m14s	1d17h8m11s
320	4m12s	20h57m59s	836	8m22s	1d17h48m32s
345	4m23s	21h55m10s	861	8m32s	1d18h39m20s
369	4m32s	22h41m28s	885	8m50s	1d20h11m56s
394	4m31s	22h34m51s	910	9m50s	2d01h11m34s
418	4m56s	1d00h38m08s	934	9m18s	1d22h29m42s
443	5m10s	1d01h51m39s	959	9m33s	1d23h44m29s
467	5m20s	1d02h40m30s	984	9m53s	2d01h24m59s
492	5m33s	1d03h45m46s	X	X	X

Table B.4: Time taken for the training using the CHALLENGE-GLAS-2015 dataset.

Number of annotations	IoU	Dice	Hausdorff
1	0.134963451639602	0.233061553616273	19.6667278691342
25	0.406097694447166	0.573415564863305	16.6370426479139
50	0.370065143233851	0.538010918780377	17.014522652877
74	0.565027440849103	0.717834971453014	16.8666074652421
99	0.554853953813252	0.710372786772879	16.6899562132986
123	0.488039319452486	0.652133919690785	17.1214390804893
148	0.586054778412769	0.732101007511741	20.1228167885228
173	0.558100692535702	0.710337924329858	18.6521890037938
197	0.58984060193363	0.736622383719996	17.910711790386
222	0.590322390982979	0.736908784038142	17.2870426177979
246	0.654088790479459	0.785220927313755	16.8128088398984
271	0.547820229279368	0.700693786144257	17.0467812387567
295	0.64595259961329	0.779058544259322	17.2716783724333
320	0.631325339016161	0.767760345810338	17.038527689482
345	0.667492105772621	0.795507735327671	16.1330974478471
369	0.641284362265938	0.77762842805762	16.714274908367
394	0.640963714373739	0.775890802082263	16.5789187581916
418	0.554752158491235	0.706994568046771	16.6800789080168
443	0.683422160776038	0.807007617072055	15.9847479368511
467	0.704057023713463	0.821703879456771	16.8653569472464
492	0.672421820853886	0.799557908585197	16.32184199283
517	0.679585403517673	0.803458922787717	18.1761774765818
541	0.691584474162052	0.814091340491646	16.2369672373722
566	0.712329403350228	0.827562200395684	17.0866102921335
590	0.722233916583814	0.835673272609711	16.0646611263877
615	0.723334958678798	0.836590738672959	16.0919168371903
639	0.720322359549372	0.83279150724411	16.7286749388042
664	0.698294758796692	0.820119766812575	16.3499927771719
689	0.772873225964998	0.87121414824536	15.3045013829281
713	0.726248349014081	0.83964613236879	15.2640788429662
738	0.76788518303319	0.869275378553491	15.3494365089818
762	0.780982111629687	0.875190869758004	15.914461562508
787	0.793277128746635	0.8850881174991	14.9911182805112
811	0.795336914689917	0.885686952816812	14.5177038594296
836	0.833561087909498	0.908752353567826	13.727401281658
861	0.808896045935781	0.893568224028537	15.3461092145819
885	0.829450817484605	0.906479939034111	14.3594557109632
910	0.817796340114192	0.899386766709779	14.7458968162537
934	0.7548671741234628	0.8571663185169822	16.07640437075966
959	0.8599421350579513	0.9248113443976954	13.91171932220459

Table B.5: Table containing the full results of the experiment for the gland in the CHALLENGE-GLAS-2015 dataset.

Number of annotations	IoU	Dice	Hausdorff
1	0.440822282280677	0.612641685666182	8.13557607088334
25	0.374878545029041	0.546147212768212	7.67179421889476
49	0.350311283308726	0.519696556986907	7.6366591942616
73	0.629942946709119	0.77041711180638	10.4907422249134
97	0.554593648665991	0.709998256885088	12.980841703904
121	0.663443371271476	0.792890244569534	8.97251705022959
145	0.676344820704216	0.804475092735046	9.69570806087592
170	0.67923902395444	0.80346140723962	9.43254778935359
194	0.685655509432157	0.806892549380278	9.41375037340018
218	0.687880571071918	0.81131196480531	9.24235386726184
242	0.643694247954931	0.777159046668273	11.6693780789009
266	0.728477478791506	0.83776872127484	8.4410623159164
290	0.600390459100405	0.743488134099887	11.4351963813488
315	0.661804567926969	0.792421012352675	10.4845440723957
339	0.684452715592507	0.807273336710074	8.53163249676044
363	0.737225973453277	0.843991226874865	7.97313126845238
387	0.718349345219441	0.830262609017201	9.92606143156687
411	0.737453180627945	0.84052644096888	8.94300832809546
435	0.712816102000383	0.822591763658401	8.29882540763953
459	0.760604207332318	0.858449993225244	8.04218769073486
484	0.739174623137865	0.842088230909445	8.47891779434987
508	0.755243786634543	0.851233331820904	7.86951974110726

Table B.6: Table containing the full results of the experiment for the gland in the CHU dataset.

B.3 Inflammation experiments

Number of annotations	IoU	Dice	Hausdorff
1	0.406481656432152	0.581738195816676	45.17033598423
8	0.359065507849057	0.538078526655833	36.0087419748306
16	0.342213466763496	0.518751005331675	32.6253609339396
24	0.345124622186025	0.523719137907028	32.4382936477661
32	0.354984952012698	0.534026807546616	32.682200050354
39	0.329852708180745	0.504154853026072	32.3957365274429
47	0.331807555754979	0.507238660256068	32.6273934284846
55	0.350121786197027	0.528172461191813	31.4913201014201
63	0.569519169131915	0.729021904865901	23.6433340708415
70	0.57212119003137	0.731265749533971	25.4491042455037
78	0.556380029519399	0.72206524014473	24.8933689753215
86	0.541706463694572	0.69729749361674	25.2855469385783
94	0.554597232739131	0.703943834702174	24.9839519500732

Table B.7: Table containing the full results of the experiment for the inflammation in the NEO04 dataset.

Number of annotations	IoU	Dice	Hausdorff
1	0.159739800526396	0.275838824177998	81.2161566969277
21	0.324579921348587	0.476007913333782	80.1031936700793
41	0.314432064395236	0.46535787925772	80.057087884433
62	0.564761470949304	0.694634568108165	84.5184162526891
82	0.651104487910651	0.789183939064758	72.5076418061187
103	0.650830406209697	0.792921129776084	64.2910900392394
123	0.634355722778085	0.770441805754883	62.5941625401594
144	0.625734321449114	0.764316058677176	67.5329426198766
164	0.642891797466554	0.776350764476735	59.7526270410289
185	0.674104315431222	0.803024431933527	60.8727937228438
205	0.666050258753956	0.798581275171128	56.6290492182193
226	0.642875248539275	0.792744343263515	58.6542471249898

Table B.8: Table containing the full results of the experiment for the inflammation in the CHU dataset.

B.4 Infiltration experiments

Number of annotations	IoU	Dice	Hausdorff
1	0.362322944402695	0.526112189463207	7.05807527814593
24	0.30905404984951	0.46689201933997	7.16667282921927
48	0.302257842251233	0.45880777665547	7.11129572732108
71	0.776145819255284	0.872241253512246	5.78780201503209
95	0.752011362143925	0.85703661101205	6.30412729808262
118	0.768954638072423	0.866910658563886	5.85971341133118
142	0.524071366872106	0.678203595536096	9.36685434068952
165	0.780642770017896	0.874610338892255	5.8113518851144
189	0.776497612680708	0.872109622614724	5.7675998210907
212	0.776357611588069	0.871057690892901	5.84856879370553
236	0.780796182155609	0.87378260578428	5.6082049369812
259	0.756516907044819	0.856987416744232	5.8671888760158
283	0.7443261572292873	0.8518579483032227	6.839428983415876
306	0.778990832396916	0.8722250495638166	5.868098395211356
330	0.7554889406476702	0.8581765311104911	6.3658715384347095
353	0.770774291242872	0.866341279234205	5.846986906869071
377	0.7898010117667061	0.8810719421931675	5.833331680297851
401	0.7582096525600979	0.8607830813952855	6.089215346745083
424	0.798336689812796	0.886754519598825	5.98935544150216
448	0.795871753352029	0.884401263509478	5.63733976909093
471	0.700509382145745	0.820951776845114	7.38862213407244
495	0.712574339764459	0.828495693206787	7.20661446707589
518	0.79376368692943	0.884222342286791	6.13005519594465

Table B.9: Table containing the full results of the experiment for the infiltration in the CHU dataset.

B.5 Tumour experiments

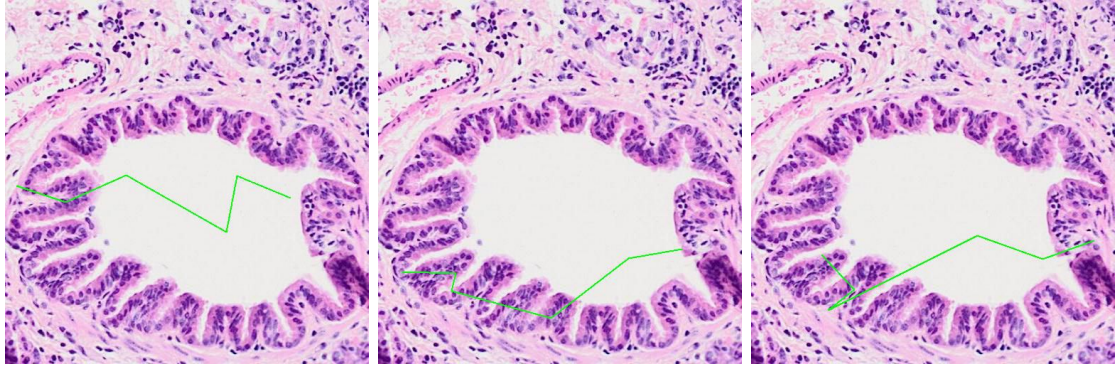
Number of annotations	IoU	Dice	Hausdorff
1	0.227500966900299	0.365966380123169	46.3029984774128
25	0.238460536144914	0.378472058042403	46.2689224135491
50	0.247675652585683	0.389997242198836	46.2906359088036
75	0.405951484677292	0.552531872425349	45.6663653004554
99	0.534479062643743	0.682189364587107	45.7554179314644
124	0.348777617147613	0.465069098277919	44.8635968046804
149	0.410638933340388	0.546202463248084	45.2516380683068
173	0.475391396111058	0.616505647979436	44.6728329793099
198	0.293784201866196	0.393837534912652	45.4294792202211
223	0.3659738764287	0.47527743894006	44.8131164139317
247	0.388221800687813	0.518014233078687	44.4626470669623
272	0.469075928171796	0.609325970372846	43.6415013151784
297	0.463154654469221	0.606585264926957	44.1091715885747
321	0.539990028666873	0.680131687752662	43.293980796491
346	0.4926329244288706	0.6363293456454431	44.86554942207952
371	0.4535755629260694	0.5931625584921529	44.25749895072752
395	0.5651130121081106	0.7052940050921133	42.82950025220071
420	0.6187174632664649	0.7568738626376275	43.02151649036715
445	0.670323216626721	0.796966002352776	47.491218993740695
469	0.6370217523747875	0.7679017753370346	50.28076563919744
494	0.62679844950476	0.7646672574262465	42.6734324328361

Table B.10: Table containing the full results of the experiment for the infiltration in the CHALLENGE-CAMELYON16 dataset.

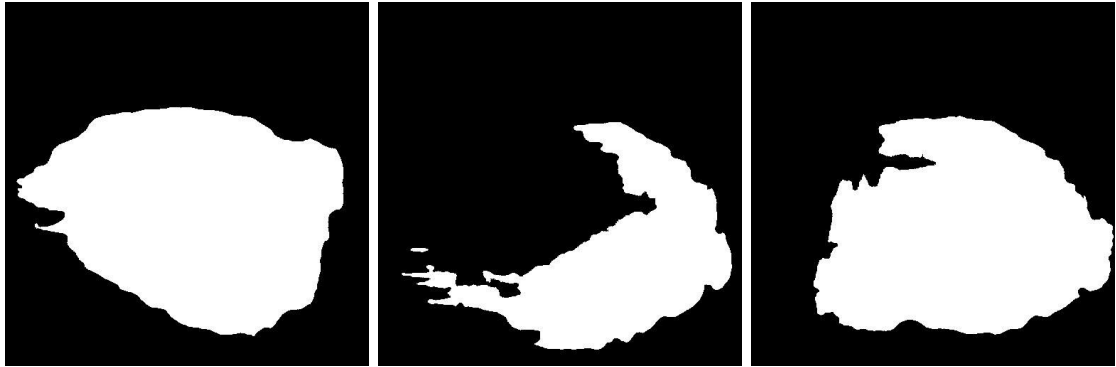
Appendix C

Quality analysis

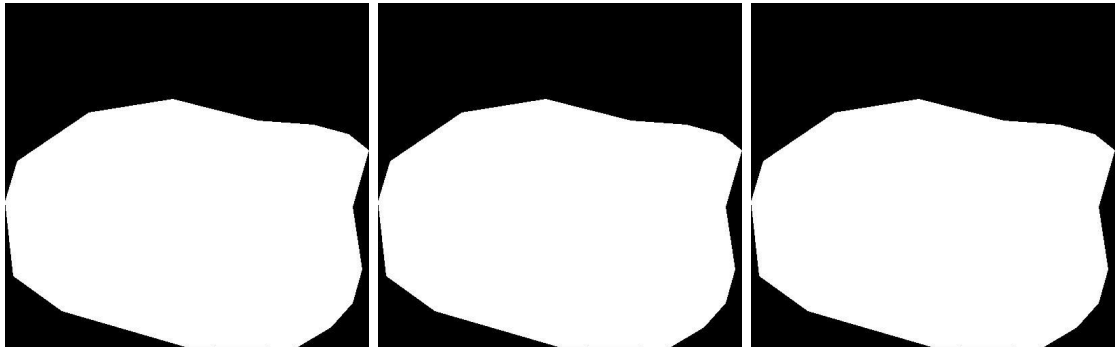
C.1 Illustrations of the scribbles



(a) Random scribbles

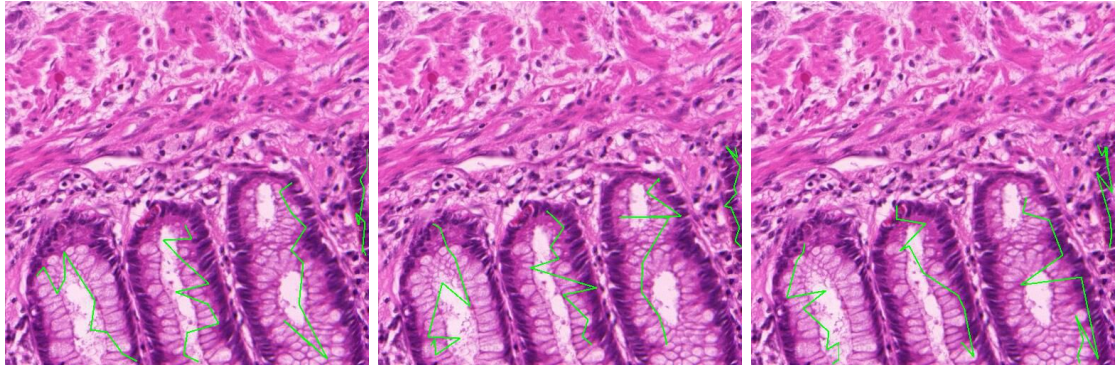


(b) Predicted mask



(c) Ground truth mask

Figure C.1: Scribbles with 5 intermediate steps on bronchi from the ULG-LBTD-NEO04 dataset.



(a) Random scribbles



(b) Predicted mask



(c) Ground truth mask

Figure C.2: Scribbles with 10 intermediate steps on glands from the CHALLENGE-GLAS2015 dataset.

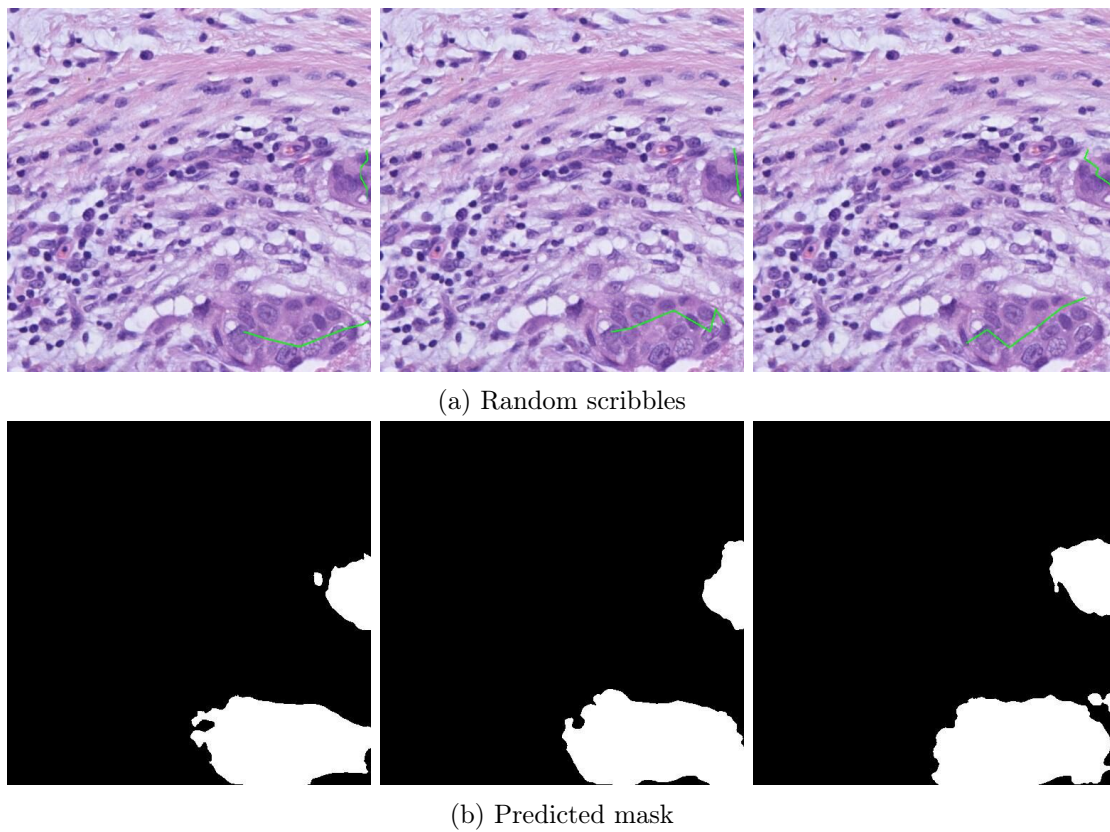


Figure C.3: Scribbles with 5 intermediate steps on infiltrations from the CHU-ANAPATH-NST-DL dataset.

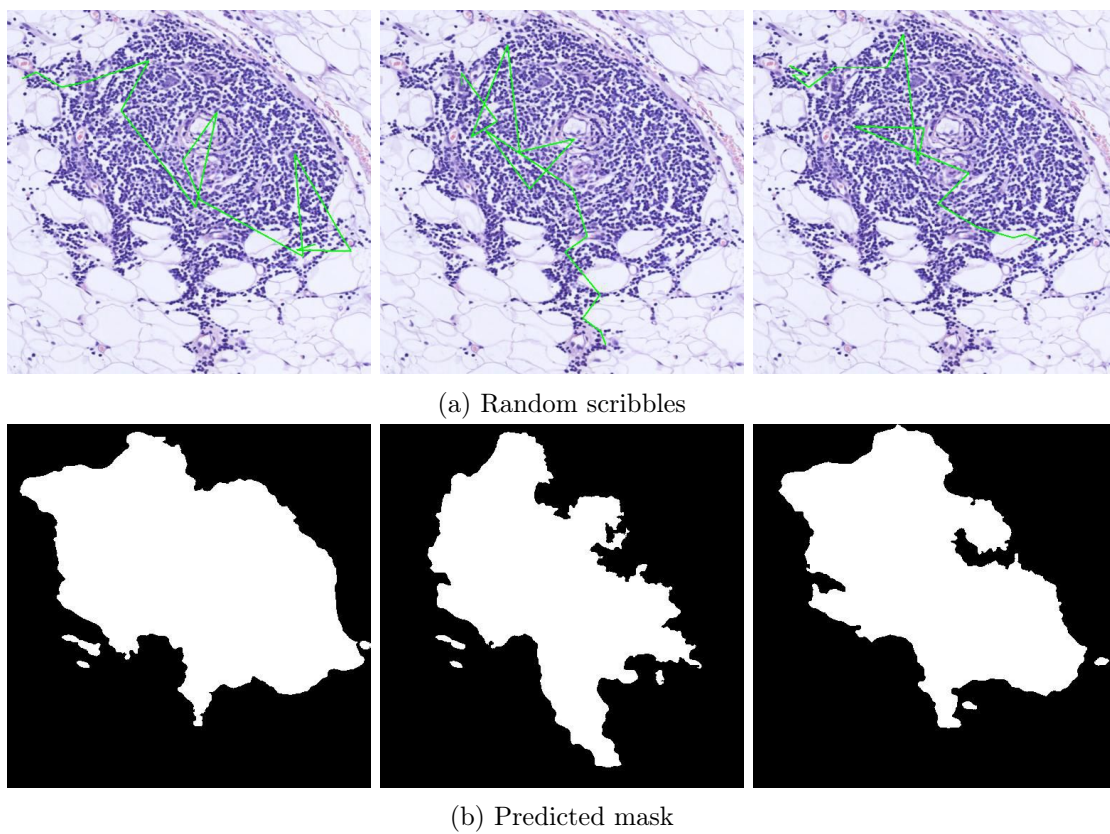
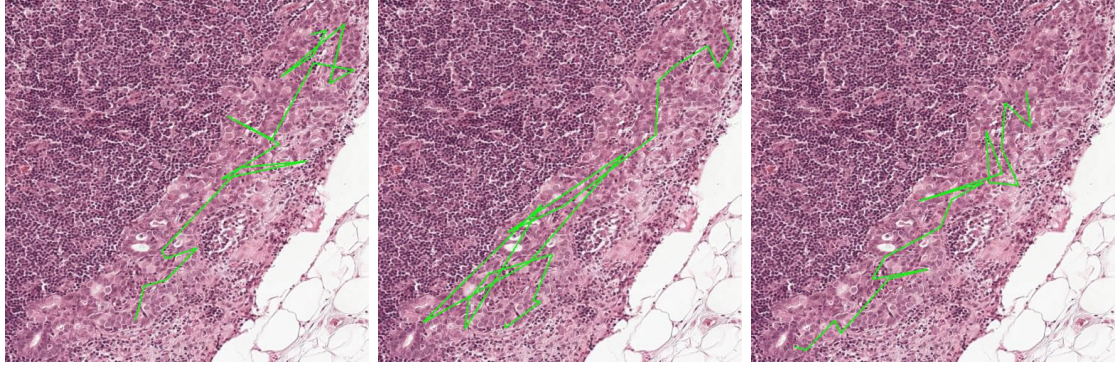
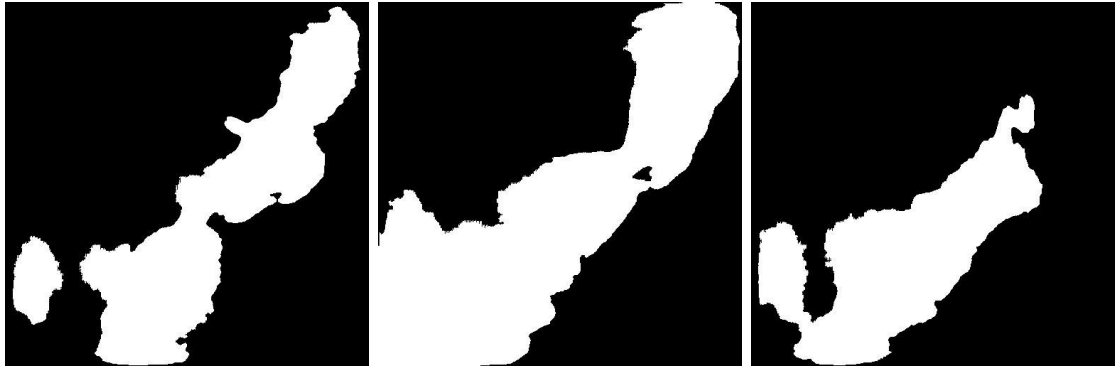


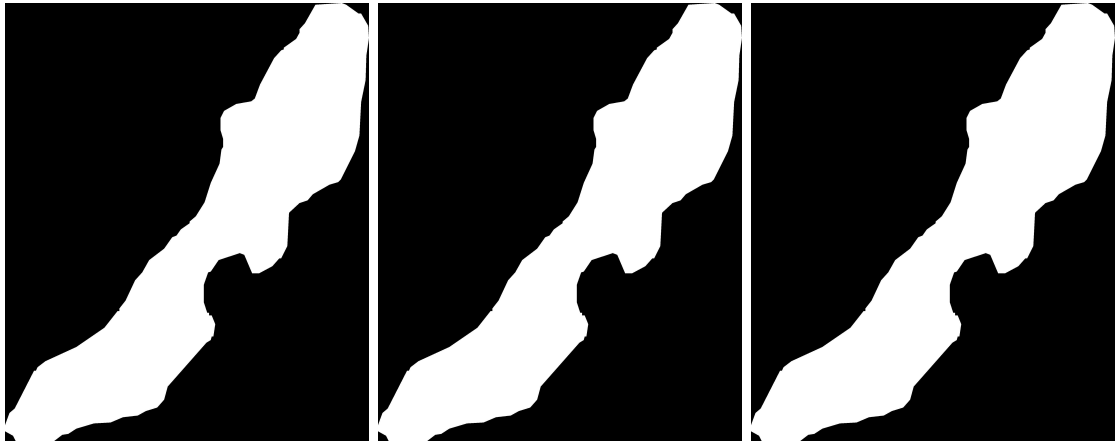
Figure C.4: Scribbles with 15 intermediate steps on tumours from the CHALLENGE-CAMELYON16 dataset.



(a) Random scribbles



(b) Predicted mask



(c) Ground truth mask

Figure C.5: Scribbles with 20 intermediate steps on tumours from the CHALLENGE-CAMELYON16 dataset.

C.2 Performance

Dataset	Type	Step	IoU	Hausdorff
ULG-LBTD-NEO04	Bronchus	5	0.7924444227011551	88.79120825377989
ULG-LBTD-NEO04	Bronchus	10	0.7997899039390501	93.68647591496857
ULG-LBTD-NEO04	Bronchus	15	0.8258550722554256	89.00003325323544
ULG-LBTD-NEO04	Bronchus	20	0.8229476512877594	91.39493540195232
ULG-LBTD-NEO13	Bronchus	5	0.7343734914923573	57.80455151549331
ULG-LBTD-NEO13	Bronchus	10	0.7411434906157287	56.41544609241657
ULG-LBTD-NEO13	Bronchus	15	0.7429414529789675	55.86773766698064
ULG-LBTD-NEO13	Bronchus	20	0.7250272998133221	58.96873589249345
CHALLENGE-GLAS-2015	Gland	5	0.7773686563432739	115.93626215165122
CHALLENGE-GLAS-2015	Gland	10	0.790741185935443	116.93976867721791
CHALLENGE-GLAS-2015	Gland	15	0.793711974236526	115.94567150818675
CHALLENGE-GLAS-2015	Gland	20	0.7977959711043561	117.20166609196063
CHU-ANAPATH-NST-DL	Gland	5	0.6537191954416534	62.32481808948517
CHU-ANAPATH-NST-DL	Gland	10	0.67480719775334	59.66361352570851
CHU-ANAPATH-NST-DL	Gland	15	0.6830274444321791	58.06248302300771
CHU-ANAPATH-NST-DL	Gland	20	0.6906074182093144	57.30859044265747
CHU-ANAPATH-NST-DL	Infiltration	5	0.632712764793747	62.36056819069611
CHU-ANAPATH-NST-DL	Infiltration	10	0.6594355287387403	62.07432293862796
CHU-ANAPATH-NST-DL	Infiltration	15	0.6733321669774178	60.85761963491475
CHU-ANAPATH-NST-DL	Infiltration	20	0.682606091903984	60.50106979918407
CHU-ANAPATH-NST-DL	Inflammation	5	0.5118517303380413	72.53272956811287
CHU-ANAPATH-NST-DL	Inflammation	10	0.5384854531086586	70.90056349805012
CHU-ANAPATH-NST-DL	Inflammation	15	0.560386522092681	70.77812474937255
CHU-ANAPATH-NST-DL	Inflammation	20	0.5679312490178767	69.76636138860731
ULG-LBTD-NEO04	Inflammation	5	0.5453791595224676	66.45165642924692
ULG-LBTD-NEO04	Inflammation	10	0.5599174225467375	66.36781884335923
ULG-LBTD-NEO04	Inflammation	15	0.6000265573290573	62.80394690064178
ULG-LBTD-NEO04	Inflammation	20	0.6258719739900238	61.45073866570133
CHALLENGE-CAMELYON16	Tumour	5	0.42993677595404856	123.77677389122974
CHALLENGE-CAMELYON16	Tumour	10	0.467661785863666	123.1132267075355
CHALLENGE-CAMELYON16	Tumour	15	0.48188927785277447	121.48883154393855
CHALLENGE-CAMELYON16	Tumour	20	0.47644606034666237	121.00725452185806

Table C.1: Complete results of the quality analysis with the random scribble generation algorithm.