ECOLE POLYTECHNIQUE DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES, UNIVERSITÉ D'EUROPE

Detection of tumor area in histological sections of breast cancer using deep learning

Mémoire présenté en vue de l'obtention du diplôme
D'Ingénieur Civil en Informatique à finalité spécialisée

Nicolas Wallemacq

Promoteur
Professeur Christine Decaestecker

Superviseurs
Adrien Foucart
Egor Zindy

Service
LISA – Laboratory of Image Synthesis and Analysis

Année académique
2021-2022

ULB UNIVERSITÉ LIBRE DE BRUXELLES

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Exemplaire à apposer sur le mémoire ou travail de fin d'études,
au verso de la première page de couverture.

Fait en deux exemplaires, **Bruxelles, le** 21/08/2022
Signature

Réservé au secrétariat :	**Mémoire réussi***	**OUI**
		NON

CONSULTATION DU MEMOIRE/TRAVAIL DE FIN D'ETUDES

Je soussigné

NOM :
Wallemacq

PRENOM :
Nicolas

TITRE du travail :
EN: Detection of tumor area in histological sections of breast cancer using deep learning

FR: Detection de régions tumorales dans des coupes histologiques de cancer du sein en utilisant un réseau d'apprentissage profond

AUTORISE*

~~**REFUSE***~~

la consultation du présent mémoire/travail de fin d'études par les utilisateurs des bibliothèques de l'Université libre de Bruxelles.

Si la consultation est autorisée, le soussigné concède par la présente à l'Université libre de Bruxelles, pour toute la durée légale de protection de l'œuvre, une licence gratuite et non exclusive de reproduction et de communication au public de son œuvre précisée ci-dessus, sur supports graphiques ou électroniques, afin d'en permettre la consultation par les utilisateurs des bibliothèques de l'ULB et d'autres institutions dans les limites du prêt inter-bibliothèques.

Formulaire version 04/2015

** Biffer la mention inutile*

** Biffer la mention inutile*

Abstract

Detection of tumor area in histological sections of breast cancer using deep learning by Nicolas Wallemacq, Computer Science Engineering, Université Libre de Bruxelles, 2021-2022.

Histopathology tissue analysis became a gold standard in cancer diagnosis and prognosis. Its recent digitization offer the possibility to use computational tools to segment automatically the tumorous regions and help accelerate the workflow of pathologists. In the case of breast cancer, previous works focused on deep learning approaches trained on whole slide images of haematoxylin & eosin stained tissues to segment tumor regions. This work proposes to use a U-Net architecture to learn to segment tumors but on Ki-67-labelled digital whole slide images instead. In short, the 40 annotated whole slide images provided by a pathologist for this thesis were split into a training, validation and test sets according to a statistic rule based on the proportion of tumor on the tissues of the images. Pre-processing was used on the images to extract the tissue regions and generate tissue tiles using a sliding-window approach, with and without overlap. Traditional data augmentation techniques were also used to compensate the scarcity of training data. Different tile contexts and resolutions were investigated to assess the best setting in which to train the U-Net model. The model yielding the best results on the validation set was trained on individual tiles of 128x128 pixels at a x5 magnification. The test metrics were not assessed at the tile level but at the whole tissue region level. The segmentation mask from individually predicted tiles was obtained by stitching the tile predictions together. Using overlapping tiles in this post-processing phase and then averaging the probabilities for each pixels showed no significant difference on the test metrics. The test metrics of the model are on average: $precision = 78.97\%$, $accuracy = 65.30\%$, $specificity = 68.94\%$, $recall = 62.93\%$, $F1 - score = 67.14\%$, $IoU = 53.75\%$ and $MCC = 0.31$. Those results are not satisfying compared to previous works, mostly because of their high variance, with high performances of the model observed on some images of the test set, and bad performances on others. It was discovered at the end of this thesis that the data set used contained unannotated tissue parts. This most probably explains the under-performance of the model and the high standard deviation observed for each metric.

Key words : deep learning, U-Net, histopathology, tumor segmentation, breast cancer, Ki67.

Abstract en français

Detection de régions tumorales dans des coupes histologiques de cancer du sein en utilisant un réseau d'apprentissage profond par Nicolas Wallemacq, Ingénierie Informatique, Université Libre de Bruxelles, 2021-2022.

L'analyse histopathologique des tissus est devenue un standard dans le diagnostic et le pronostic du cancer. Sa récente numérisation offre la possibilité d'utiliser des outils informatiques pour segmenter automatiquement les régions tumorales et aider à accélérer le travail des pathologistes. Dans le cas du cancer du sein, de précédents travaux se sont concentrés sur des approches d'apprentissage profond (*deep learning*) entraînées sur des images de lames entières de tissus colorés à l'hématoxyline et à l'éosine pour segmenter les régions tumorales. Ce travail propose d'utiliser une architecture U-Net pour apprendre à segmenter les tumeurs, mais sur des images numériques de lames entières marquées au Ki-67. En résumé, les 40 images de lames entières annotées fournies par un pathologiste pour cette thèse ont été réparties en ensembles d'entraînement, de validation et de test selon une règle statistique basée sur la proportion de tumeur sur les tissus des images. Un prétraitement a été utilisé sur les images pour extraire les régions de tissu et générer des tuiles de tissu en utilisant une approche de fenêtre coulissante, avec et sans chevauchement. Des techniques traditionnelles d'augmentation des données ont également été utilisées pour compenser la faible quantité de données d'entraînement. Différents contextes et résolutions de tuiles ont été étudiés afin d'évaluer la meilleure configuration pour l'entraînement du modèle U-Net. Le modèle qui a donné les meilleurs résultats sur l'ensemble de validation a été entraîné sur des tuiles individuelles de 128x128 pixels à un grossissement x5. Les métriques de test n'ont pas été évaluées au niveau des tuiles mais au niveau de toute la région de tissu prédite. Le masque de segmentation des tuiles prédites individuellement a été obtenu en assemblant les prédictions de chaque tuile. L'utilisation de tuiles qui se chevauchent dans cette phase de post-traitement, puis le calcul de la moyenne des probabilités pour chaque pixel, n'ont montré aucune différence significative sur les métriques de test. Les métriques de test du modèle sont en moyenne: $précision = 78,97\%$, $exactitude = 65,30\%$, $spécificité = 68,94\%$, $rappel = 62,93\%$, $F1 - score = 67,14\%$, $IoU = 53,75\%$ et $MCC = 0,31$. Ces résultats ne sont pas satisfaisants par rapport aux travaux précédents, surtout à cause de leur forte variance, avec de hautes performances du modèle observées sur certaines images de l'ensemble de test, et de mauvaises performances sur d'autres. Il a été découvert à la fin de cette thèse que les données utilisées contenaient des parties de tissus non annotées. Ceci explique très probablement la sous-performance du modèle et l'écart-type élevé observé pour chaque métrique.

Mots-clefs : deep learning, U-Net, histopathologie, segmentation de tumeur, cancer du sein, Ki67

Acknowledgements

I would like to give a special thank you to Prof. Decaestecker, Mr. Foucart, and Mr. Zindy, my supervisors, who were always available and very helpful when I needed help.
Thank you to my friends Baptiste van Tuyn and Cécile Castiaux for their moral support in difficult times.
Finally, I would like to thank my family for their unconditional support throughout the years and without whom I would not be where I am today.

Contents

Acronyms

AI Artificial Intelligence. 7

CAD Computer Aided Diagnostic. 2

CNN Convolutional Neural Network. 7, 8, 11, 13, 17, 19, 22, 24, 25, 28, 61

CRF Conditional Random Field. 26, 28

DCNN Deep Convolutional Neural Network. 24, 25

DL Deep Learning. 2–5, 7, 8, 10, 11, 19, 23, 24, 29–31, 35, 36, 43, 61

DNN Deep Neural Network. 8–11

FCN Fully Convolutional Network. 7, 16, 19

H&E Haematoxylin & Eosin. 1, 4, 28

IHC immunohistochemistry. 1, 3, 4, 26, 29, 35, 60, 61, 63

ML Machine Learning. 7, 29, 46

NN Neural Network. 8, 9, 11

ReLU Rectified Linear Unit. 8

ROI Region Of Interest. 22, 24, 25

WSI Whole Slide Image. 1, 3–5, 7, 21–25, 28–37, 39, 40, 42, 46–48, 52–59, 61–69, 75–84

Chapter 1

Introduction

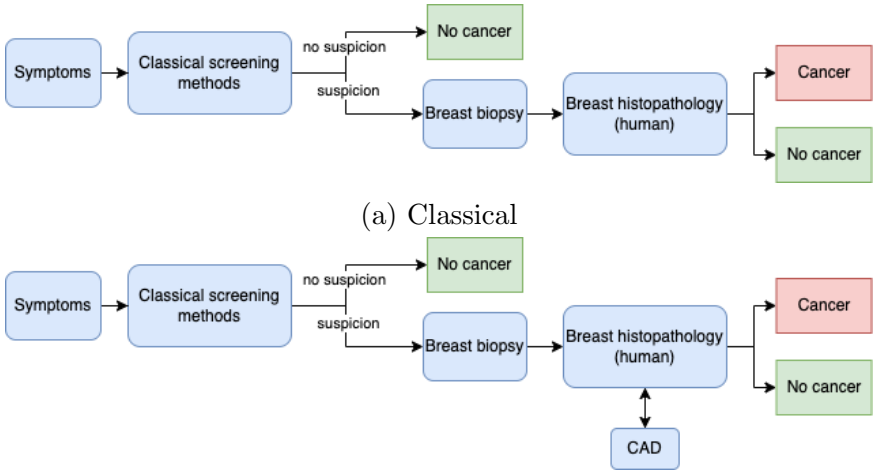
Breast cancer manifests itself by an uncontrolled growth of cells in the breast and is the second most invasive cancer in women [1]. Amongst all the cancers that were detected in 2020 worldwide, 11.7% of them were breast cancer, making it the most widely diagnosed type of cancer. It impacts primarily women, with 12% of women having a lifetime risk of getting breast cancer. Only in 2020, nearly 700 000 people died from it, representing 6.9% of all the cancer deaths [2].

As stressed in [3], early diagnosis of cancer significantly increases the probability of survival. The need to detect it early motivated the awareness campaigns among the communities to do a regular check-up [4].
The first step in the diagnosis pipeline is the detection of a breast abnormality thanks to medical imaging. Nowadays, biomedical imaging has become a pillar of modern healthcare. So much that, today "a world without imaging is clearly not imaginable" [5]. Thus, one of the following screening methods (or a combination of them) can be used to detect breast cancer : screen film mammography, digital mammography, ultrasound, magnetic resonance imaging, digital breast tomosynthesis [4].

Once an abnormality has been detected through one of the latter imaging modalities, a breast tissue biopsy is conducted. Breast histopathology on that tissue sample helps to confirm the presence of cancerous cells. Histopathology tissue analysis has become the gold standard in cancer diagnosis and prognosis [6] but there is a shift towards its digitization. According to [7], the ongoing adoption of digital pathology in the clinic will be "one of the most disruptive technologies introduced into the route working environment of pathologists". The pathologists can detect the tumorous regions on a digital Whole Slide Image (WSI) and assess the tumour proliferation rate by either identificating and enumerating mitoses in Haematoxylin & Eosin (H&E) stained tissues or by using immunohistochemistry (IHC) to label a proliferation marker in cells [8]. In this thesis, the IHC labelled proliferation marker is Ki67, a protein associated with cell proliferation. The pathologists first needs to detect a tumorous region and then detect its hotspot. It is defined as the region of highest concentration of Ki67-positive tumour cells and it is used for the prognosis and the treatment of breast cancer [8].

However, this whole pathological analysis is tiresome, difficult, highly specialized and time-consuming for pathologists. Due to the human's subjectivity and fatigue, it is error-prone and there is a consequent variability in the results of their analysis [9], so much that there is a difficulty of reaching a consensus between pathologists. Indeed, a 2015 study [10] examining breast biopsy concordance found out that the pathologists disagreed on average 24.7% of the time between each other on a diagnosis.

Those downsides, the lack of consensus and the growing adoption of digital pathology (meaning more data to be processed) clearly emphasize the need to use Computer Aided Diagnostic (CAD) to automate the image analysis and segment automatically the tumorous regions, as well as computing the different indicators. This synergistic tool would alleviate the workload of pathologists, increase the throughput and accuracy, as well as allow a standardization of the clinical practices in the diagnosis of breast cancer [9] [6] [7]. A comparison of the diagnostic pipeline with and without using CAD is presented in Figure 1.1.



(a) Classical

(b) Using computer aided diagnostic (CAD)

Figure 1.1: 2 diagnostic pipelines for breast cancer

The first attempts of automating image analysis began with the introduction of digital scanning technologies in biomedical imaging in the 60's. From this moment until recently, classical methods for automated bio-image analysis consisted in feature engineered algorithms [6]. However, as stated in [9], microscopy images could contain a plethora of information which may go unnoticed even to the expert human eye, so designing classical methods to extract this information is a great challenge. Their efficiency is limited because their algorithms rely on a predetermined set of features. They can thus only classify something as good as this set of features allows to [6].

At the end of the 2000's and the beginning of the 2010's, a major paradigm shift occurred as Deep Learning (DL) techniques began to outperfom other machine learning algorithms for various tasks in nearly all fields of science, including bio-image analysis. This big resurgence of Deep Learning is principally explained by 2 factors: larger dataset availability due to the fast digitization (DL needs a lot of data to work properly) and

increasing computing power (DL is computationally expensive) . Contrary to classical methods where features are extracted using handcrafted techniques, Deep learning takes its strength in the fact that it automatically derives features from the input data [9] [6] [4]. According to [11], DL has the potential to reach the level of experienced pathologists.

In this context, this thesis aims at using the power of deep learning to segment automatically the tumor area in IHC Ki67 stained histological sections of breasts. This delineation must be robust to changes in immunohistochemical labeling so that it can be applied to other markers of interest. This work is structured in the following way. Chapter 2 presents the main challenges of working with WSIs ; Chapter 3 presents the key DL concepts to understand the neuronal architecture used in this work and how performances of a model can be assessed ; Chapter 4 lies out the previous works in the literature on similar tasks ; Chapter 5 presents the data that was used to carry out the objective of this thesis and how it was separated into a training, validation and test sets ; Chapter 6 details all the methods of this work that were used to train and evaluate the different models ; Chapter 7 sums up the performances of the different models, selects one model based on the validation metrics, and tests it on previously unseen data to assess its true performance ; and Chapter 8 recapitulates the work done, the performance achieved and proposes ideas of improvements.

Chapter 2

Whole Slide Imaging

This chapter aims at briefly introducing what is a WSI and the challenges of using such images to train a DL model to automatically segment tumor regions.

The WSIs are the result of the scanning and digitization of entire histology slides. Those contain for instance tissue sections from formalin-fixed paraffin-embedded (FFPE) blocks that were then stained with the relevant marker [8]. The context in which such images are used was already explained in Chapter 1. All WSIs are captured using brightfield illumination [7]. These images are in the order of gigapixels and are usually stored in multi-resolution pyramidal format [6]. The Figure 2.1 shows 2 examples of WSIs of different tissues: one using H&E staining and the other one using the IHC proliferation marker Ki67.

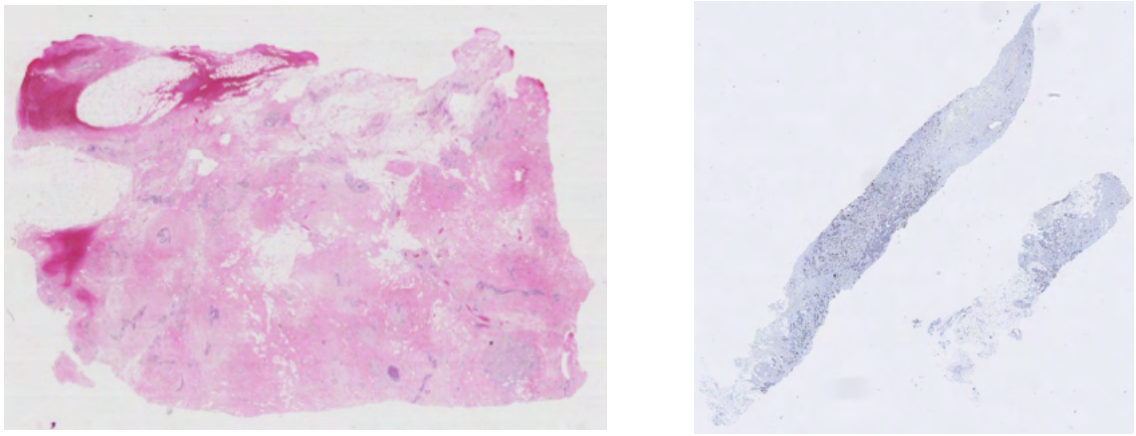


Figure 2.1: Example of WSI at low magnification of a H&E stained WSI on the left (image taken from [12]), and a Ki67 WSI on the right

The WSIs present several issues/challenges that are listed below:

- The high dimensionality of WSIs: their size - typically 100 000 x 100 000 pixels -

is too big for classical deep learning algorithms or at least the current hardware is unable to deal with such big images [7].

- Insufficient training samples: pathologist annotated WSIs are costly and take a lot of time to make. These data are scarce, the few available are of small sample size and have a limited geographic variety [6].
  Moreover, there exists multiple image formats for WSIs, with different resolutions. This slows down the curation of large public datasets and consequently slows down the research. This is problematic since there is a need of thousand of annotated training samples to train successfully a deep learning network [13]. There are, however, challenges that are being organized and that make the data set used for the challenge available to the public. It should be noted though that there are currently discussions to adopt a single open source file format [7].

- The scanner variability across laboratories: multiple WSI scanners exist in the market. In addition to participating to the scarcity of data due to the different compression types, sizes and output formats, they use different illumination conditions and objectives [7]. These varying illumination conditions constitute an additional challenge when trying to train a DL model able to generalize across data coming from different scanners.

- The stain variability across laboratories: there is a lack of uniformity in the staining protocols, meaning that data acquired from different sources have a high variability in staining, stain chemicals and staining time. This poses a problem because it can lead to unsatisfactory performances, with one model trained on one center's data set with satisfactory results performing non optimally on another center's data set [14] [6].

- The presence of artifacts: they can appear either during the sample preparation workflow or/and during the imaging process. Examples of artifacts can be: ischemia times, fixation times, cutting artifacts, focusing, adenosis, presence of red blood cells in the tumor, tear in tissue, foreign object during coverslip, pen demarcations [7] [15] [12]. While humans can easily train to ignore those artifacts, ways to overcome this problem must be found for the DL model to generalize well to other data sets.

- The Ki-67 expression itself: it is not limited to the tumor and the tumor can contain Ki-67-negative cells as well as Ki-67-positive nuclei. In addition, the expression is not uniform across the tissue, complicating furthermore the training of a segmentation algorithm [8].

- Lack of annotations across an image: this shortfall in annotation can lead to erroneous local predictions [15].

- Data curation bias [15]

- Extreme class imbalance: the tumorous region can be underrepresented in the WSI [15].

Such challenges will need to be addressed - when possible - during the training pipeline of this work.

Chapter 3

Key Deep Learning Concepts

Deep Learning (DL) is a form of Machine Learning (ML), itself a major branch in the field of Artificial Intelligence (AI), a scientific discipline concerned with the science and engineering of developing machines exhibiting characteristics associated with human intelligence [9]. This set and subset of AI families are shown in Figure 3.1.

Recently, DL became the most used computational approach in ML thanks to outstanding results on complex tasks, beating other ML techniques in many domains - including medical information processing - and even human performance [16].
The aim of this thesis is to use deep supervised learning to train an agent to learn to segment tumor regions on breast tissue WSIs. Before diving into the related works (Chapter 4), this chapter aims at introducing the subject of DL to the reader. Rather than focusing on the mathematics, this Chapter aims at presenting the key ideas, concepts and components of such computational approach. Since the task is to segment images, a particular emphasis will be put on Convolutional Neural Network (CNN)s and Fully Convolutional Network (FCN), which are particular DL architectures performing well on image data since it automatically detects significant features without any human supervision.

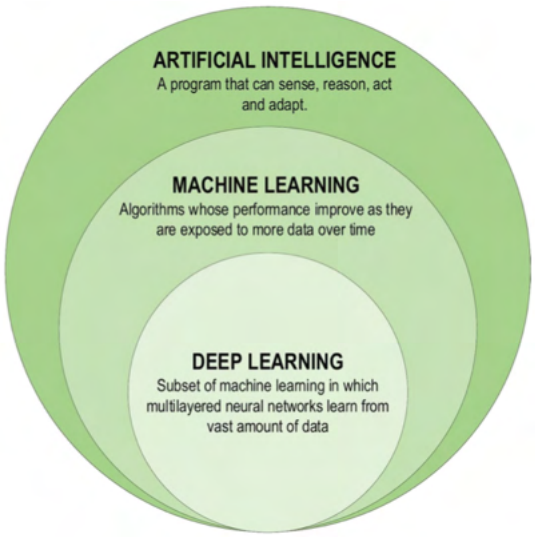


Figure 3.1: DL in AI. Illustration taken from [16]

3.1 Deep Neural Network

The idea behind DL is to algorithmically emulate the biological neural networks present in the human brain. Thus, DL uses Artificial Neural Networks (ANN) with *multiple* layers of elementary computational cells (the "neurons") that act as function approximators to extract higher-level representations of given input data in order to perform data analysis tasks through multiple nonlinear transformations. Those neurons are organized into 3 interconnected layers: input, hidden, and output layers. When a NN has numerous layers of hidden units, it is referred to as a *deep* neural network (DNN) [17] [9].
Before explaining how multiple interconnected layers are organized and interact within DNN, the structure and training method of a *single* neuron is first introduced in the next paragraph.

A standard neuron is a computational unit taking n input values $\mathbf{x} = [x_1, ..., x_n]$ and their associated weights $\mathbf{w} = [w_1, ..., w_n]$. The neuron first applies a linear transformation of its inputs $z = \mathbf{w}^T\mathbf{x} = \sum_{i=1}^{n} w_i x_i$, followed by a nonlinear activation function $g(z)$. Different activation function g exist, such as sigmoid, tanh, and Rectified Linear Unit (ReLU). The latter is the most commonly used with CNNs and is defined as:

$$g(x)_{ReLU} = max(0, x) \tag{3.1}$$

The nonlinearity induced by the activation function gives the ability to learn complex things, so the output $g(z)$ can be used to perform, for example, a classification. For the neuron to learn something, a loss function L is needed to encode the optimization objective of the NN and quantify the gap between the predicted output of the neural network $\hat{\mathbf{y}} = f(\mathbf{x}; \mathbf{W})$ and the ground truth $\mathbf{y}$, where $\mathbf{W}$ is the weight matrix. Different loss functions exist and its choice depends on the data types and tasks. However, it should be noted that this choice will have a big effect on the NN since different loss functions lead to different loss value for a same prediction. An example of loss function L is the negative log-likelihood which is commonly used for binary classification: $L = -\sum_{i=1}^{N} y_i log\hat{y}_i + (1 - y_i) log(1 - \hat{y}_i)$; where y_i is the ground truth binary label for the observation $i = 1, ..., N$, and $\hat{y}_i = P(y_i|\mathbf{x}_i)$ is the probability estimate of y_i based on the data $\mathbf{x}_i$. [17]
To train this single neuron, a stochastic gradient descent is used to adjust its weights. The interested reader should refer to [17] for more information on this gradient descent. The training of a single neuron neural network is represented in Figure 3.2.

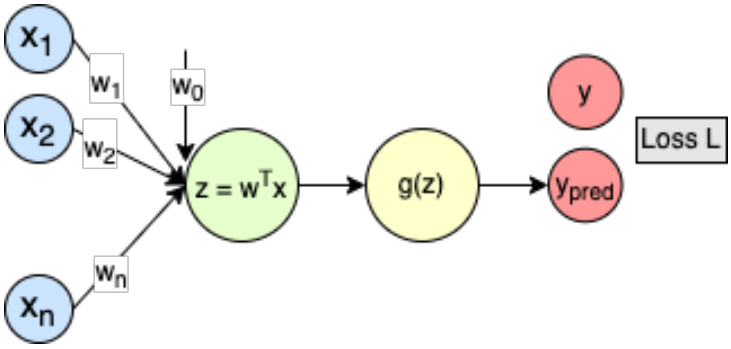


Figure 3.2: Training of neural network consisting of a single neuron. Illustration inspired by [17]

Let's use the aforementioned neuron and incorporate it in a more complex network such as a DNN which, as previously mentioned, uses one or several hidden layers of neurons, with connections between the neurons of consecutive layers. This network defines a mapping function $\mathbf{y} = F(\mathbf{x}; \mathbf{W})$, where $\mathbf{W}$ is now the parameter set of the NN and $\mathbf{W} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)} | 1 \leq l \leq L\}$, where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector of the l-th layer, respectively. There is thus a flow of computation starting from the input vector $\mathbf{x}$ to the ouput vector $\mathbf{y}$ via hidden layers $h^{(l)}$. [17]
In the same fashion as with the training of a single neuron, training a DNN also requires a gradient descent algorithm to update the weights and the bias of the network. The intuition behind gradient descent is given in Figure 3.3 with a 1D example. To reach the optimum of the objective function (minimum in the case of a minimization problem), x is modified by small steps using the opposite of the derivative of the function[18].

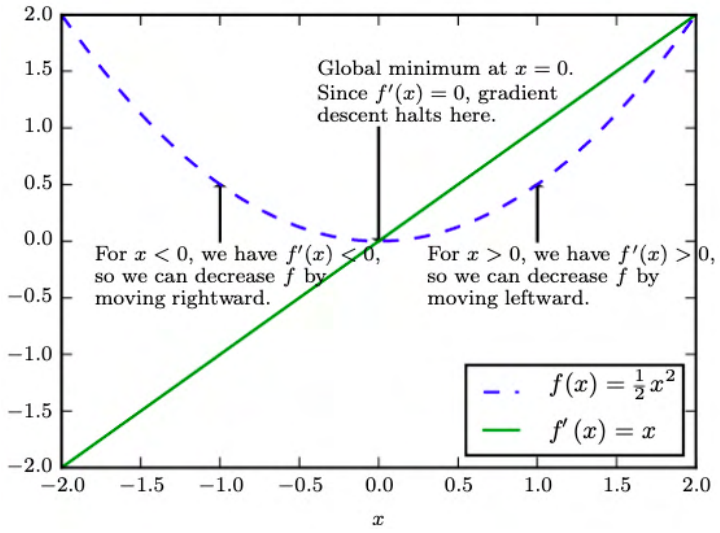


Figure 3.3: Illustration of the gradient descent with a simple math function. Illustration taken from [18]

In a minimization problem, different critical points can be found: local minimum, saddle point (Figure 3.4 and 3.5) or global minimum. Sometimes, the search space

is too broad to be looked in its entirety but a good stochastic gradient descent algorithm should find a local minimum that performs nearly as well as the global one in a reasonable amount of time[18].

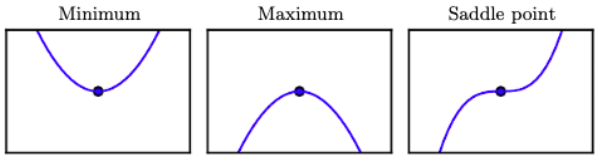


Figure 3.4: Different types of critical points. Illustration taken from [18]

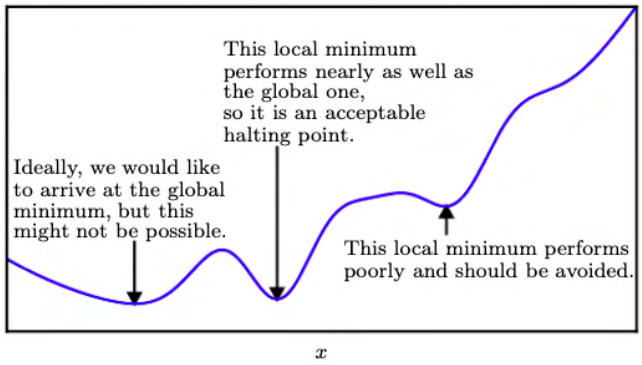


Figure 3.5: Illustration of an approximate minimization. Illustration taken from [18]

In practice, in DL, x is a vector thus the gradient - the generalization of the derivative to a vector - is used to find the minimum of the loss function that is defined for the network. In DL, optimizers such as Adam (Adaptative moment estimation) are used to improve and speed up the convergence while avoiding the local minima and overfitting. This proves useful, especially in the context of high-dimensional parameters space [19]. The interested reader should refer to [17] and [18] for more information on gradient descent. The training of a DNN with one hidden layer is represented in Figure 3.6.

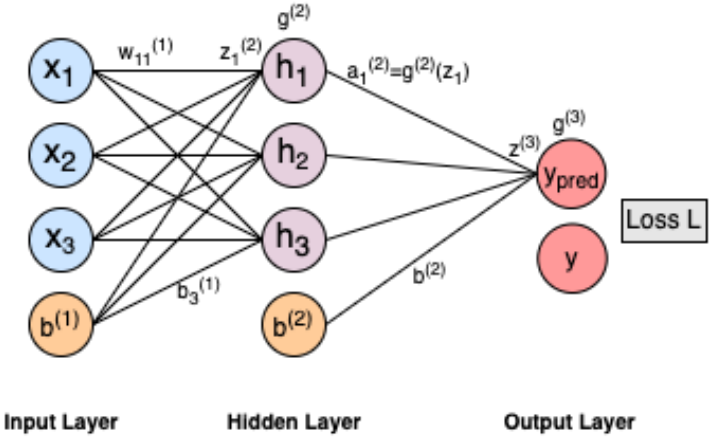


Figure 3.6: Training of a deep neural network with one hidden layer. Illustration inspired by [17]

3.2 Convolutional Neural Network

CNNs are a specific type of DNN that are ideally used when the raw input data are 2D, 3D, such as images and videos and the task of the network is classification (explanation of what is depicted by the data), object detection (bounding box to classify distinct objects on an image, for instance), or segmentation (pixel-wise classification to provide the exact mask of where the object of interest is on, for instance, an image). CNNs are NNs with convolution layers that often come with additional nonlinear activation functions (after all layers with weights) and pooling layers, followed by fully connected layers. The convolutions allow the network to focus on local properties by creating local connections, such that each region in the input layer is connected to a unit in the next layer. The presence of those convolutions is what distinguishes CNNs from other DL models [17][16]. An example of a CNN architecture is given in Figure 3.7 for the task of a binary classification consisting in assigning a label "cat" or "dog" to an input RGB image.

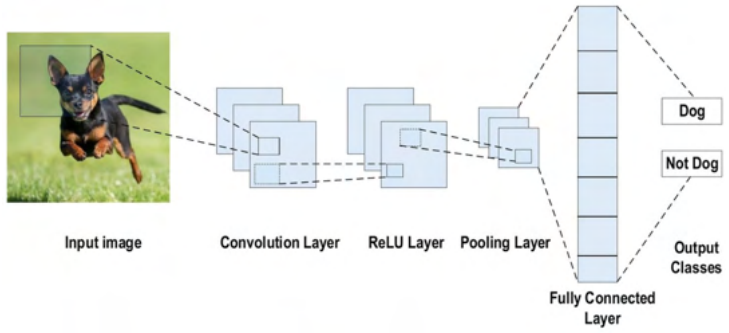


Figure 3.7: Example of CNN architecture for image classification. Illustration taken from [16]

The next subsections give more insights to the key components of CNNs, namely the convolutional layers, the pooling layers and the fully connected layers.

3.2.1 Convolutional Layer

A convolutional layer is a collection of convolutional filters. Those filters, also called kernels, are a grid of discrete numbers that can be seen as local pattern extractors that will be applied everywhere on the input image to generate meaningful features[17]. Those numbers - the kernel weights - are either initialized randomly or recovered from a previously trained model (transfer learning), and will be adjusted through the learning process in such a way that it reduces the loss function [16]. To obtain the feature maps (the output of the convolution), the dot product, i.e. the element-wise multiplication followed by summation to get a scalar value, between the image and the kernel is computed. The kernel slides horizontally and vertically according to a stride (step-size over all vertical or horizontal locations) on the image and each time the dot product is computed until

no further sliding is possible[17][16]. An example of convolution of a 5x5 single input channel with a 3x3 kernel and a stride of 1 is given on Figure 3.8. In practice, there are more channel inputs, e.g. 3 in the case of RGB images (one for each color channel) but the same principle applies. Note that in this example, the feature map is of lower dimensions. In fact, increasing the stride value lowers even more the dimensions of the output map. [16]. To counter that, padding can be used to deal with elements on the boundary that otherwise disappear. It consists in adding extra values (usually 0s) outside the input data boundary to ensure that convolution can be applied to elements near the boundary [17]. As a result, the size of the input image increases and so does the output feature map [16], meaning it is possible to get an output of the same size than the input in a convolution.

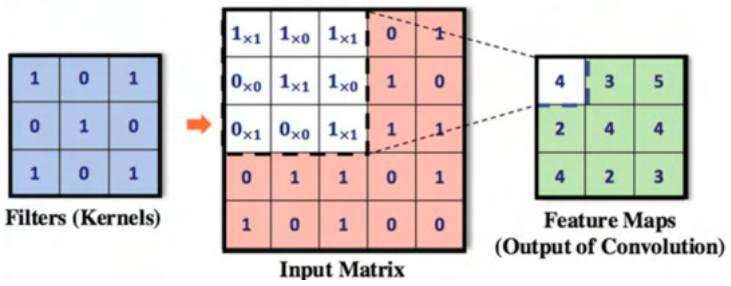


Figure 3.8: Example of a 2D convolution. Illustration taken from [17]

3.2.2 Pooling Layer

Pooling layers usually follow convolutions in order to down-sample the feature maps by aggregation. It has the advantage of reducing the number of network parameters (and thus accelerating the training process) by shrinking large-size feature maps to create smaller feature maps that maintain the the majority of the dominant local information/features and are translation invariant. There exist several pooling methods, namely: *mean pooling* that computes the average values from the pooling region, *max pooling* that picks the maximum value from the pooling region, and *sum pooling* that sums up all values from the pooling region [16]. The Figure 3.9 illustrates a max pooling operation where, as expected, the dimensionality of the feature space shrinks from 5x5 to 3x3.

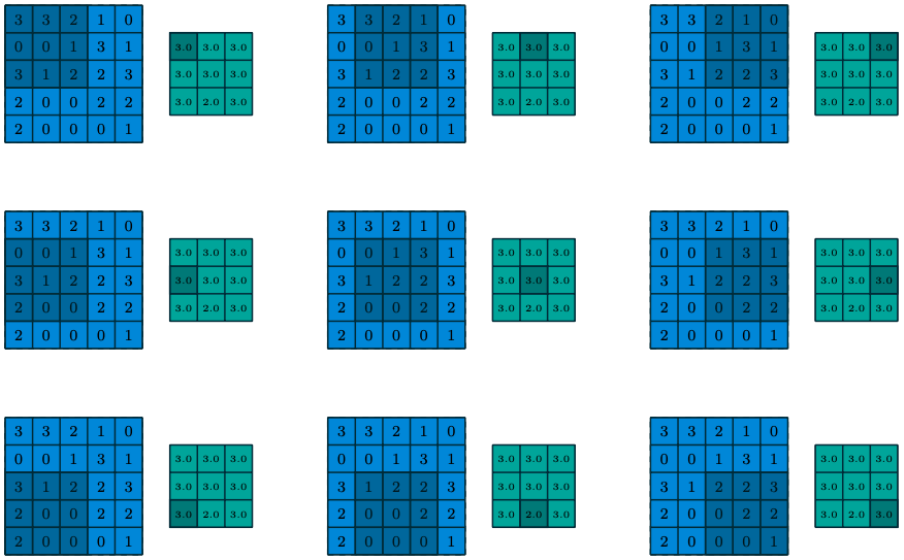


Figure 3.9: Example of a a max pooling operation. Illustration taken from [20]

3.2.3 Fully Connected Layers

Fully connected layers were already discussed in Section 3.1, where all neurons from one layer are connected to the neurons of the next layers. In CNNs, they are often added after all convolutions and pooling layers. The reasons are twofolds: they can work as classifiers to use the high-level features obtained after a series of convolutional and pooling layers to map the input image into different classes based on the training dataset, and they can yield to better classification results thanks to the integration of features thanks to nonlinear combinations [17].

3.2.4 Classification metrics

Performance metrics are essential when training and evaluating a model. [16]. One should understand well the metrics in order to be fully able to compare the obtained results. In the literature, many performance metrics are used. While some metrics are more popular than others, one ought to be cautious to interpret correctly the results. Indeed, some statistical measures can show overoptimistic results when working for instance with imbalanced data sets [21].

A confusion matrix is used to evaluate a classification model [22]. However, it is also relevant for binary segmentation, as the latter can be viewed as a per-pixel binary classification. Such matrix takes the form presented in Table 3.1.

Confusion Matrix		
	Actual Postive Class	Actual Negative Class
Predicted Positive Class	**TP**	**FN**
Predicted Negative Class	**FP**	**TN**

Table 3.1: Confusion Matrix [22]

Where:

- TP (True Positive) represents the number of positive instances that were correctly classified
- TN (True Negative) represents the number of negative instances that were correctly classified
- FP (False Positive) represents the number of misclassified positive instances
- FN (False Negative) represents the number of misclassified negative instances

Below are presented the main metrics that are generally used [22].

The *accuracy* gives the percentage of correct predictions over the total number of instances evaluated. Mathematically, this gives:

$$Accuracy = Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.2}$$

The *recall*, also known as *sensitivity*, gives a model's proclivity to detect the **positive** class. It is the ratio of correctly identified positive patterns. Mathematically, this gives:

$$Recall = R = \frac{TP}{TP + FN} = Sensitivity = Sn \tag{3.3}$$

The *specificity* gives a model's proclivity to detect the **negative** class. It is the ratio of correctly identified negative patterns. Mathematically, this gives:

$$Specificity = Sp = \frac{TN}{TN + FP} \tag{3.4}$$

The *precision* P, also known as *Positive Predictive Value* PPV, gives the the percentage of positive patterns that are correctly predicted from the total predicted patterns in a positive class. Mathematically, this gives:

$$Precision = PPV = P = \frac{TP}{TP + FP} \tag{3.5}$$

The *Negative Predictive Value* NPV, gives the the percentage of negative patterns that are correctly predicted from the total predicted patterns in a negative class. Mathematically, this gives:

$$NPV = P = \frac{TN}{TN + FN} \tag{3.6}$$

The *F1-score* is the harmonic mean between precision and sensitivity. Its value ranges from 0 (all the positive samples are misclassified) to 1 (perfect classification) [21]. However, as can be clearly seen with its mathematical expression, the F1-score has a major drawback: it does not take into account the TN. Mathematically, this gives:

$$F1 - score = \frac{2 * P * Sn}{P + Sn} = \frac{2 * TP}{2 * TP + FN + FP} \tag{3.7}$$

The *Mathhews Correlation Coefficient* (MCC) on the other hand is an alternative measure less affected by imbalanced datasets. Indeed, the previous metrics, although popular, do not consider the ratio between positive and negative elements. MCC is invariant for class swapping, meaning it is as good to evaluate a data set imbalanced towards the positive class than it is to evaluate a data set unbalanced towards the negative class. However, a drawback of MCC (that Sn and Sp do not exhibit) is that it is sensible to variations of the percentage of positive or negative instances in the data set.
To get a good MCC score, the model needs to predict correctly the majority of the positive class and at the same time predict correctly the majority of the negative class, which makes it the preferred metric for the segmentation task of this work. Its value ranges from -1 (perfect misclassfication) to 1 (perfect classification), with 0 being the expected value for a random coin tossing classifier [21]. Mathematically, this gives:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{3.8}$$

To assess the performance of a model in distinguishing between the positive and the negative classes, one can draw the *Receiver Operating Curve* (ROC), which is a 2-dimensional curve representing the sensitivity as a function of $1 - specificity$, with

a range between $(0, 0)$ and $(1, 1)$, as shown on Figure 3.10. The *Area Under the Curve* (AUC) gives a measure of that performance and summarizes the ROC. The values of AUC range from 0 to 1 (perfect classification) [12].

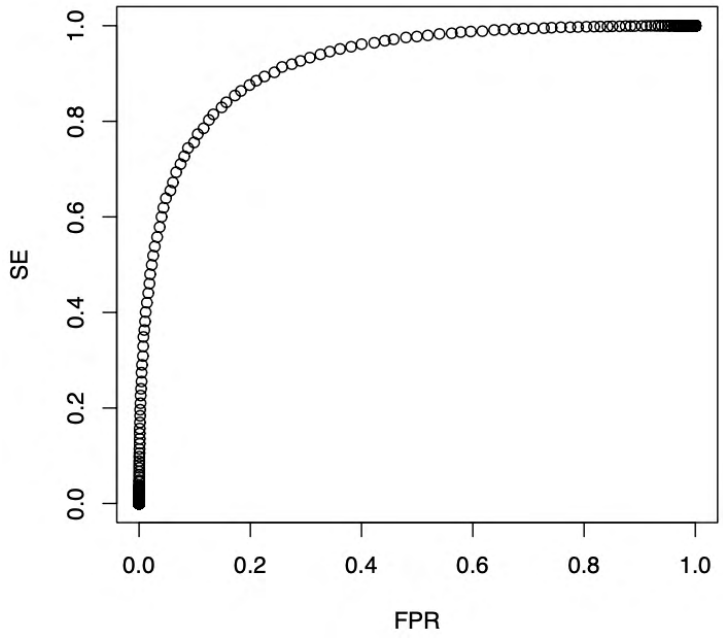


Figure 3.10: Example of ROC curve. Illustration taken from [23]

3.3 Fully Convolutional Network

As seen previously, convolutional networks can be used for classification tasks where the output of an image is a single class label. However, and especially in biomedical imaging, the desired output can be a precise localization (segmentation), where a class label is assigned for each pixel of the input image. FCN do precisely that by supplementing the usual contracting networks with successive layers containing upsampling operators that increase the resolution of the output by combining high resolution features from the contracting path with the upsampled output[13]. One extension of this FCN architecture has gained a lot of popularity due to high performances in the biomedical imagery: the U-Net. The latter is detailed in Section 3.3.1.

3.3.1 U-Net architecture

As [13] states it, the U-Net is a segmentation network which is a modification and extension of the FCN. The architecture of the U-Net is presented in Figure 3.11. It consists of a contracting path (encoder) that captures the context and a symmetric expanding path (decoder) that enables precise localization. The chaining of those two paths, with

a “symmetric” number of filters, gives the “U” shape architecture to the network [13].

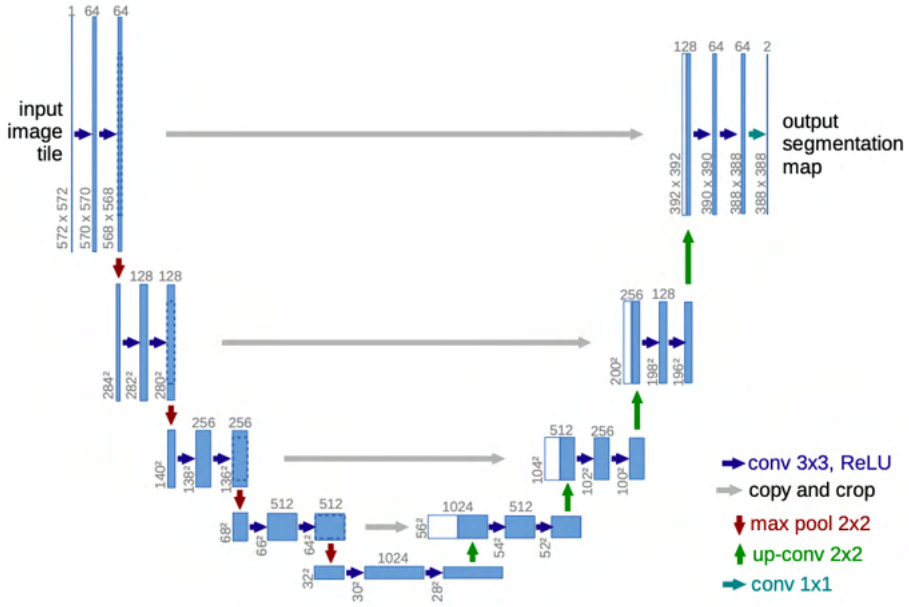


Figure 3.11: Original U-Net architecture. Illustration taken from [13]

The contracting path is a typical CNN architecture consisting of two 3x3 convolutions followed by a a ReLU activation function and a 2x2 max pooling operation to downsample the input. At each downsampling operation, the number of feature channels is doubled until reaching the bottom the network. There, the same two 3x3 convolutions each followed by a ReLU are applied. The expanding path is similar, except that the features maps are upsampled thanks to 2x2 “up-convolutions” halving the number of features channels and concatenating them with the corresponding feature maps from the contracting path in order to better incorporate the context. This transfer of contextual information from the encoding path to the decoding path is done thanks to long skip connections. At the final layer, a last 1x1 convolution is applied in order to map each component of the last feature vector to the desired number of classes [13]. The soft-max activation function on this last layer assigns for each pixel the probability of belonging to the positive class and the negative class in the case of binary segmentation. Applying a threshold on either the positive probabilities or the negative probabilities gives a predicted segmentation mask.

3.3.2 Segmentation metrics

As stated previously, binary segmentation can be viewed as a per-pixel binary classification. Thus, all the metrics described at Section 3.2.4 can be used if now TP represents the number of positive pixels that were correctly classifed, TN represents the number of negative pixels that were correctly classified, FP represents the number of misclassified positive pixels, and FN represents the number of misclassified negative pixels. However,

as [21] states it, there is no consensus on one specific metric to use in the context of binary segmentation.

For segmentation in general, the Intersection over Union (IoU), the Mean Intersection over Union and the Frequency (MIoU) and the Weighted Intersection over Union (FWIoU) are also used. The IoU is defined as the area of the intersection of the positive ground truth (Y) with the positive prediction ($\hat{Y}$) divided by the area of the union. In other words, it gives the overlap between the ground truth and the predicted output [24]. Mathematically, this gives:

$$IoU(Y, \hat{Y}) = \frac{|Y \cap \hat{Y}|}{|Y \cup \hat{Y}|} = \frac{TP}{TP + FN + FP} \tag{3.9}$$

When dealing with multi-class problems, one can use MIoU or FWIoU. MIoU refers the IoU on a per-class basis and then averaged and the FWIoU weights the importance of each class score with its frequency of occurrence [24]. Mathematically, this gives:

$$MIoU = \frac{1}{k} \sum_{i=0}^{k-1} \frac{n_{ii}}{\sum_{j=0}^{k-1} n_{ij} + \sum_{j=0}^{k} n_{ji} - n_{ii}} \tag{3.10}$$

$$FWIoU = \frac{1}{\sum_{i=0}^{k-1} \sum_{j=0}^{k-1} n_{ij}} \sum_{i=0}^{k-1} \frac{\sum_{j=0}^{k-1} n_{ij} n_{ii}}{\sum_{j=0}^{k-1} n_{ij} + \sum_{j=0}^{k} n_{ji} - n_{ii}} \tag{3.11}$$

where k is the number of classes, n_{ij} is the number of pixels of class i predicted as belonging to the class j[24].

In the context of segmentation, the *Dice similarity coefficient* is also commonly used. It is used as a spatial overlap index and ranges from 0 (no spatial overlap) to 1 (complete overlap) [24] [25]. However, with the task of binary segmentation, as it is the case in this thesis, the F1-score and the Dice coefficient are equivalent [26]. Mathematically, this gives:

$$Dice = \frac{2|\hat{Y} \cap Y|}{|\hat{Y} + Y|} \tag{3.12}$$

where $\hat{Y}$ is the set of pixels belonging to the ground truth, Y is the set of pixels of the segmented image, and |.| point out the number of elements of a set[24] [25].

3.4 Evaluation strategy

In machine learning in general, a K-fold cross validation strategy validation strategy is commonly used to assess the performance of a model. The idea is to randomly split the

entire data set into K non-overlapping "folds" (constituting of a training & test set) of equal size. Then the model is trained K times until all folds have been used once as a test set and the model's overall performance is the average model performance across the K folds. However, in DL, the data sets become huge and the training time long thus the parameter tuning becomes too impractical with K-folds cross validation. In this case, it is preferable - as an evaluation strategy - to perform a single random split of the data set into 3 partitions: a training set, a validation set, and a test set. The models are trained with the training set and their performance assessed with the validation set to determine the best training setting possible (hyper-parameters, choice of pre-processing, architecture choice, number of epochs, etc.). Once the best setting is found, it is used to train the the final model on both the training and validation sets and its performance is assessed on the test set. The latter is considered as the estimate of the true model performance, as it is done on previously unseen data [17].

Over-fitting is a core problem with CNNs and FCN. A model is said to be over-fitted if it performs particularly well on training data but not on unseen data. On the contrary, a model is said under-fitted if it does not learn sufficiently from the training data. When training a DL model, the aim is to find a just-fitted/balanced model which performs well on both training and testing data[16], i.e. generalizes best. An illustration of the idea of over-fitting is represented in Figure 3.12.

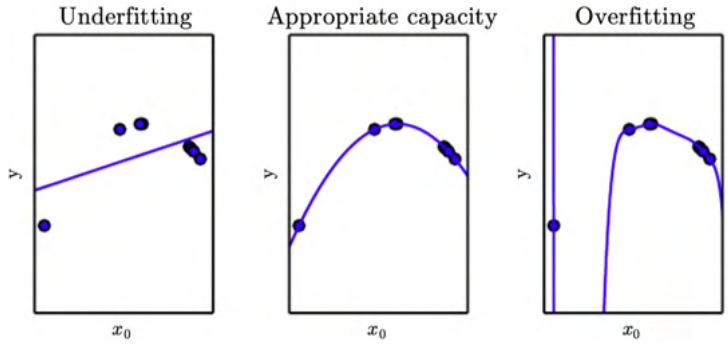


Figure 3.12: Example of under-fitting, just-fitting, and over-fitting of a model trying to model a few data points. Illustration taken from [18]

To have more generalization and thus avoid over-fitting, various concepts are used such as dropout, drop-weights, data augmentation, and batch normalization. Dropout consists in randomly dropping some neurons during each training on one epoch in order to force the network to learn different independent features. Drop-weights is similar to dropout but instead of dropping neurons, weights are dropped at each epoch. Data augmentation refers to the creation of artificial data to increase the available data to train on, since DL models require a sizeable amount of data to train correctly. Finally, batch normalization consists in subtracting the mean and dividing by the standard deviation to normalize the output at each layer. Not only does it reduces over-fitting, but it also reduces the time needed for the network to converge[16].

Over-fitting can be detected by monitoring the performance on the training

data and on the validation data. It happens when the validation performance reaches an optimum while the training performance continues to increase. The training can be automatically stopped thanks to early stopping and only the model at the epoch that attained the best performance on the validation set is saved.

Chapter 4

Related Works

Previous works in the literature have already attempted to automatically segment breast tumorous regions in histopathological images. While some papers addressed some of the constrains imposed by WSIs, others tried to address many of them in a comprehensive pipeline. Few used data sets using Ki-67 labelled tissue, and more papers used H&E staining. In this review of the previous works, both staining will be considered.

4.1 Tumor detection on breast cancer WSIs

In order to train deep learning models to segment tumorous regions on on Ki-67 breast cancer WSIs, all previous works [8] [12] [27] [24] [28] have a similar pipeline. The main elements of the latter are

- A data pre-processing/preparation step, where some of the intrinsic problems of WSIs are tackled as well as ways to accelerate the training/inference procedure. This is reviewed at Section 4.1.1.
- A deep learning architecture. This is reviewed at Section 4.1.2.
- A post-processing step, where the predicted tumor regions are further refined to get a cleaner segmentation. This is reviewed at Section 4.1.3.

4.1.1 Data pre-processing/preparation

An inherent problems of WSIs mentioned in Chapter 2 is the presence of artifacts such as pen demarcations. Another important aspect to consider is that the background (glass slide) takes a lot of space in WSIs. Those problems can be tackled using pre-processing. Indeed, since it is relatively easy to discriminate between regions that contain tissue and

those who do not, [8] [12] [27] generated a mask to remove the non-tissue regions in order to accelerate the inference time. [8] used a threshold based on the mean intensity, while [12] [27] used Otsu adaptive thresholding on the saturation channel of the HSV image. Pen demarcations are considered as noise on a WSI thus [12] generated a mask using a threshold on each RGB channel in order to remove them from the image. It is worth noting that the same masks that are applied on the original WSIs must also be applied on the annotation masks. As those steps are based on high level features, they can be done at a low magnification in order to be computationally less demanding. Furthermore, as a significant portion of the WSI is removed from the analysis, only a fraction of the WSI will be considered for the training, which will lead to a training data set exempt from this useless part of the image. The same argument holds for when later a new unseen WSI will need to be predicted (during the test phase or in practice), as only a fraction of the image will need to be predicted, alleviating the computation load. The inference time is thus faster which is crucial for pathologists in practice.

As mentioned in Chapter 2, one of the intrinsic problems with WSIs are their high-dimensionality. It is not possible to give directly a high-resolution WSI as input to a DL model. In fact, even at low magnification, WSIs remain with a too high dimensionality for direct processing on a CNN [12]. Thus, a way to reduce the dimensionality needs to be found. To overcome this problem, all works used a sliding-window approach. This consists in tiling the WSI into many tiles of a given width (W) and height (H), at a given magnification. An example of the tiling process is given in Figure 4.1.

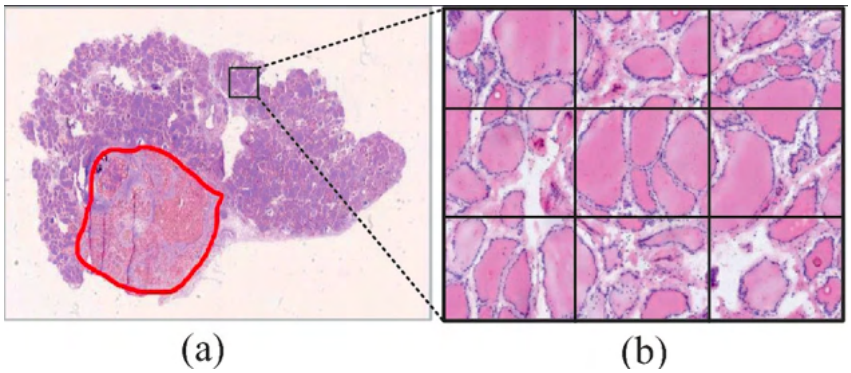


Figure 4.1: Tiling process: a WSI (a) is divided into patches of the same size (WxH), at a certain magnification (b). Illustration taken from [29]

The majority of works chose to work directly at the highest magnification possible (20x for [8], 40x for [27] [24]).
Others chose to combine different resolutions. [28] tried working at x5, x10 and x40 magnifications, but obtained its best segmentation when combining the different resolutions in its model. [12] on the other hand worked at 5x magnification to first get a rough estimate of the ROI and then further refined the segmentation by working on patches at 40x magnification extracted from this identified region.

The tile size used in the previous works varies, going from as little as 64x64

[8] to as big as 1280x1280 [27]. The context each tile incorporates depends on the size and magnification. However, as [28] states it: "A sliding window approach to crop fixed-sized images from WSIs is promising, but dividing the large structures limits the context available (...) and affects the segmentation performance". A particular attention must thus be placed on getting the right trade-off between magnification and tile size. Indeed, when pathologists diagnose breast cancer on WSIs, not only do they check the characteristics of cells but they also observe the tissue around [27] .

Another intrinsic problem of WSIs mentioned in Chapter 2 is their scarcity. However, [9] rightly points out that in DL large data sets are needed in order to properly train the large number of neural network parameters. To overcome this problem, all works [8] [12] [27] [24] [28] used realistic data augmentation techniques for the training phase. Common techniques were: horizontal/vertical flipping, rotation, color jittering, random zooming and cropping.

Finally, another challenge inherent to the data is the extreme class imbalance. Indeed, the positive class (tumorous regions) is underrepresented. This means that there are much more negative (non-tumorous) tiles than there are positive (tumorous) tiles. It is therefore crucial to guide the sampling of data. In that regard, [27] sampled randomly negative tiles in order to get a 1:1 ratio between the positive and negative class during the training process. Another work [15] used a K-means negative sampling strategy. It consists in using features generated by EfficientNet-B0 model trained on ImageNet and then cluster the features in a number C of clusters using the mini-batch K-means algorithm. Then, a number n of samples are extracted from each cluster to form a set of $N_{negatives} = C.n$, where $N_{negatives}$ is the number of negative samples considered for the training. Their findings is that this clustering allows to sample the negative space more evenly, which improved the overall performance, especially reducing the false positives rate. Figure 4.2 compares the random negative sampling versus the feature based K-means negative sampling.

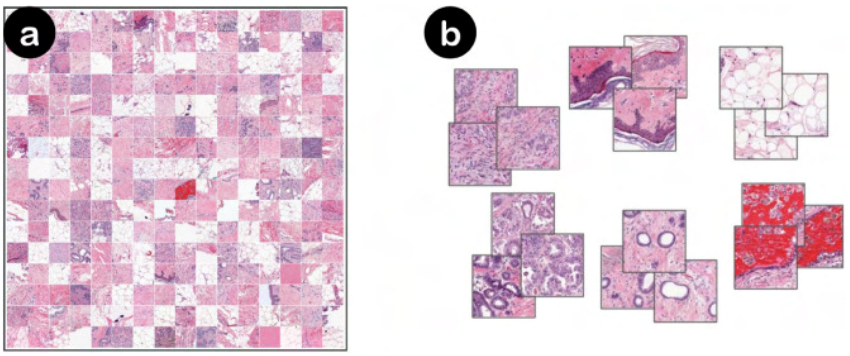


Figure 4.2: Two different negative sampling strategies for selecting the negative patches extracted from a WSI (a) Random negative sampling (b) K-means negative sampling. Illustration taken from [15]

Another approach is to employ hard negative mining [7], which consists in adding iteratively false positives to the training data for further training. This was done by [8],

but requires the help of a pathologist *during* the training to identify the false positives and false negatives generated by the model on the test set.

4.1.2 DL Architecture

The previous subsection (4.1.1) reviewed the main aspects regarding the data preparation for WSIs and their importance. Indeed, as [12] puts it, "a well-prepared data set is a crucial factor in defining a good generalization for DL techniques".
The present subsection deals with the different DL techniques that have been used in the literature to train on such well-prepared data. They differ in the aim (classification or (multi-class) segmentation) and in the architecture. Some works even combined different models, either to speed-up the inference process or to try to get better performances thanks to ensemble learning. While binary segmentation models aim at only finding the ROIs in the image, multi-class segmentation models go one step further and attempt to classify each pixel to a particular type of cancer category.

[8] developed a custom DL model (PTM-NET) based on a CNN architecture of 4 layers to *classify* each patch as tumorous or non-tumorous of Ki-67 WSIs. They used a combination of convolutions, and max-pooling operations to reduce the feature map size as well as to provide the feature map resistance to the translation invariant. Also, to reduce the computational time during the training as well as to avoid overfitting, a dropout ratio of 0.5 was used on the full connection layer to exclude non-active neurons. Since PTM-NET is a classifier, it outputs a label (tumorous or non-tumorous) for each patch. However, as PTM-NET takes as input small 64x64 patches at 20x magnification, assigning the predicted label of the patch to all of its pixels and aggregating all the results is a segmentation process as such. It is interesting to note that the authors of this paper compared their 4-layers CNN architecture (PTM-NET) with another 16-layers CNN architecture (VGG-NET). PTM-NET, despite having less layers and requiring less computation and GPU requirements, outperformed VGG-NET, indicating that "histopathology tumour images possess colour and textual properties that are captured only using a few convolution filters".

[27] also used a classifier but integrated it in a cascade framework they named v3_DCNN. The authors first used a slimmed-down version of the Inception_v3 classifier to preselect tumor regions (acting as a pre-processing step) and then employed a semantic segmentation model Deep Convolutional Neural Network (DCNN) to get a refined semantic segmentation on those identified regions. Their DCNN is based on ResNet-101, where they replaced most of the traditional convolution and pooling layers by atrous convolutions. Considering the computational and memory limitations of general computers, the bilinear interpolation operation was abandoned and the final predicted probability map is of size (W/8 x H/8) compared to the original WSI size of WxH. The authors of this paper experimented with several high-resolution patch sizes as input for their DCNN (321x321, 768x768 and 1280x1280) and got the best performances with the largest patch size (1280x1280). Their interpretation is that larger patch sizes contain more contextual information, explaining thus the better segmentation performance. The ensemble

model consisting of both v3_DCNN with 768x768 patches and v3_DCNN with 1280x1280 patches was able to perform slightly better on one metric while others declined slightly. Given the significant increase in inference time, ensemble models might not be useful in practice.

[12] on the other hand did not make use of any classifier but as [27] they implemented a cascade framework. The first module is a U-Net architecture to detect the ROIs in low magnification WSIs (x1.25). Then, the second module is a ResNet-50/U-Net architecture to get a refined semantic segmentation at high magnification (x40) on the ROIs identified by the previous module.

[24] and [28] both used CNNs for binary and multi-class segmentation, respectively. The first work used a publicly available model Deeplabv3+, which is a DCNN along with an encoder-decoder with a separable atrous convolution architecture. Several backbone networks (Resnet v1, Mobilenet v2 and Xception 65) were tried, and using Xception 65 yielded the best results. The second work used a CNN architecture with a residual encoder-decoder where new features were added, namely an input-aware encoding block, a densely connected decoding network, an additional sparsely connected decoding network, and a multi-resolution network (for context-awareness learning).

In their work, [6] showed that it is better to pre-train models with different histopathology data sets when training models where the pathology data sets are limited, as it is the case in this thesis. This transfer learning approach was used in [12], [24] and [28] for the initialization of weights in the networks, although not on medical images. [24] used a network pre-trained on the Pascal VOC 2012 data set, and [28] initialized its encoder weights with those of ResNet-18 trained on ImageNet.

4.1.3 Post-processing

Some works did not consider any post-processing of the model predictions, apart from applying a threshold to each predicted tile to retrieve the corresponding tile mask of the tumour and then stitching the masks together to retrieve the whole slide mask.

Since context information is very important for DCNNs to segment tumour tissues, other works tried to incorporate it by taking overlapping patches during the inference process. However, depending on the stride/overlap, this can greatly increase the inference time due to a greater number of patches to be processed [27]. In their work, [24] considered two approaches for the merging of 50% overlapping 500x500 patches. This overlap means that each pixel is predicted multiple times in the overlapping areas. Their first approach consisted in applying applying a 0.5 threshold on the patches to extract the masks and then applying an OR operation on the pixel predictions. However, the resulting segmentation maps were noisy and there were discontinuities in between the boundaries of overlapping regions. Their second approach tried to overcome those problems by using the logit maps of the last block of their architecture (Deeplabv3+),

normalize them thanks to a softmax function and averaging the probability maps in the overlapping areas. Then the probability maps were combined with a fully connected Conditional Random Field (CRF), which "helps defining the edge details and capturing the long-range dependencies" [24].

4.1.4 Summary

The Table 4.1 summarizes the related works and gives their results in terms of metrics. However, since those works were not done on the same data sets and with exactly the same tasks, nor with the same IHC labelling, a straight comparison of the metrics but are given here to give a hint of what is achievable.

	Joseph J. et al (2019) [8]	**Zeiser F.A. et al. (2021) [12]**	**Guo Z. et. al. (2019) [27]**	**Priego-Torres B.M. et. al.(2020) [24]**	**Metha S. et. al.(2017) [28]**
Cancer	Breast	Breast	Breast	Breast	Breast
Data	Ki67 WSI	H&E WSI	H&E WSI	H&E WSI	H&E WSI
Pre-processing	Background removal (mean intensity) Exclusion of lymphocytic cells Mean normalized input RGB	Background removal (Otsu) Pen demarcations removal (Otsu)	Background removal (Otsu)	/	/
Data preparation	Sliding-window Data augmentation Hard negative sampling	Sliding-window Data augmentation	Sliding-window Data augmentation Random negative sampling	Sliding-window Data augmentation	Sliding-window Data augmentation
Architecture	PTM-NET: CNN classifier	DeepBatch: Cascade CNN: 1) U-net to segment ROIs 2) Resnet-50/U-Net to get a refined multi-class segmentation	v3_DCNN: Cascade CNN: 1) Inception_v3 classifier 2) DCNN to get a refined multi-class segmentation	Deeplabv3+ (DCNN) with Xception 65	CNN with encoder-decoder
Transfer learning	No	Yes: unknown	No	Yes: Pascal VOC 2012	Yes: ImageNet
Post-processing	Stitching of the patch predictions Pixel label = label of its patch	Stitching of the patch predictions 0.5 threshold	Stitching of the patch predictions Overlapping patches 0.5 threshold on avg. prob. per pixel	Fully connected CRF	Stitching of the patch predictions 0.5 threshold
Results	$Dice = 0.74$ $PPV = 0.70$ $NPV = 0.883$	ROI segmentation: $IoU = 93.43\%$ $Acc = 91.27\%$ $Sn = 90.77\%$ $Sp = 94.03\%$ $F1 = 84.17\%$ $AUC = 93\%$ Refined segmentation (multi-class): $IoU = 88.23\%$ $Acc = 96.10\%$ $Sn = 71.83\%$ $Sp = 96.19\%$ $F1 = 82.94\%$ $AUC = 88\%$	$AUC = 96.6\%$ $FROC = 82.9\%$ $mIoU = 80.69\%$	$Acc = 95.62\%$ $Sp = 97.39\%$ $Se = 88.58\%$ $MIoU = 64.52\%$ $FWIoU = 92.52\%$	$PA = 70\%$ $F1 = 58.8\%$ $mIoU = 44.2\%$

Table 4.1: Summary of the related works

This literature review was useful to guide the methods that will be used in this work and that are detailed in Chapter 6. In the task of tumor segmentation of breast cancer using histopathology images, the previous works presented in this chapter used WSIs of H&E stained tissues, with the exception of [8] which used the same Ki-67 indicator than in this work. All works used a sliding-window to overcome the dimensionality problem of the data by decomposing the original - large - images into smaller tiles of equal size, so this approach will also be followed in this work. Also, [12] [8] [27] used pre-processing to focus their analysis on the tissue tiles only and discard the glass slide region, which will also be done in this work. The resulting tiles will be subject to data augmentation, as it was done in all previous works. Moreover, random negative sampling will be used during the training as in [27] since more tiles of the negative class are generated compared to the positive class. The different works used a CNN architecture to either classify or segment the tiles, or a combination of both. As [8] already investigated the use of a CNN architecture to classify the Ki67 stained WSI tiles, this work proposes instead to investigate the use of a U-Net architecture - a CNN segmentation architecture presented in Section 3.3.1 - to learn to segment tumors on similar Ki-67 labelled WSIs as this architecture - or similar - showed conclusive results in other previous works on H&E WSIs and performs remarkably in biomedical image analysis in general [12] [13]. For the inference pipeline, all works investigated the stitching of individually predicted tiles of a WSI together and the application of a 0.5 threshold on the probability maps. This will also be done in this thesis, but the effect of using overlapping tiles in the predictions and averaging the probability maps in the overlapping areas will will be investigated as in [24], but without using a CRF.

Chapter 5

Material

The aim of this chapter is to present the materials that were used for this work. Section 5.1 specifies the computer that was used, its specifications, the programming language used and the DL library that was used to carry out this work. Section 5.2 presents the data that was used to train the DL model on and how this data was prepared to be usable in this work.

5.1 Computer and environment

All the code[1] in this thesis was written in the development environment Python v3.9.7. An open-source ML library called Tensorflow (version 2.5.0) [30] was used in order to facilitate the writting and development of the DL algorithms. This means that core DL algorithms did not need to be re-implemented from scratch.

All the computations in this work were carried out on a computer with the processor Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz with 32GB of installed RAM, running Windows 10. The computer is supplemented by a NVIDIA GeForce GTX 1080 Ti GPU, with 11GB of dedicated GPU memory.

5.2 Data

The data set provided for this master thesis consists of annotated WSIs with an IHC labeling of the Ki-67 protein. An example of such WSI can be found below on Figure 5.1.

[1] `https://github.com/NicolasWa/MasterThesisFinal`

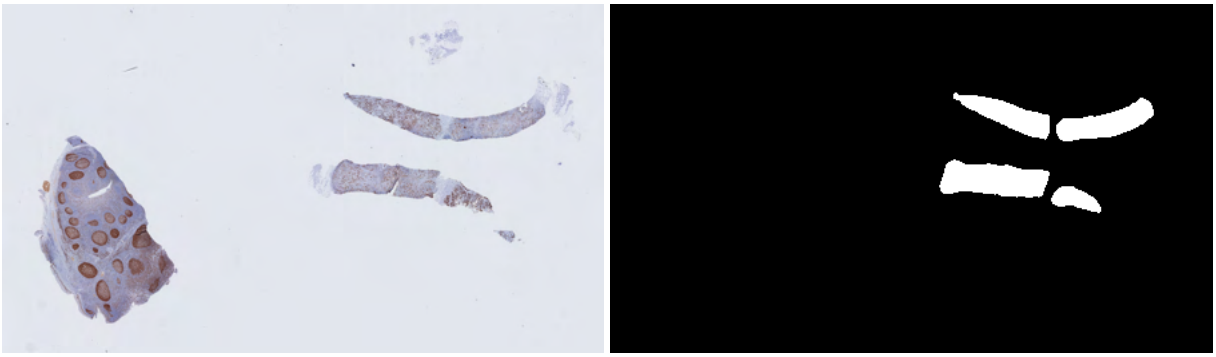


Figure 5.1: Example of WSI (left) with its corresponding annotation mask (right)

Since human Ki-67 protein is present in all proliferating cells (normal and tumor cells) but absent from resting cells, it makes it an excellent proliferation marker to determine the growth fraction of a cell population. Pathologists use the Ki-67 positive tumor cells (Ki-67 labeling index) as a quantitative measure for the breast cancer diagnosis and prognosis [31]. However, aiming to automatically segment tumors using Ki67 stained WSIs still constitutes a challenge since the Ki-67 marker expression is non-uniform, varies a lot and more importantly, is not limited to tumor. Moreover, within a tumor area there can be Ki-67-negative cells as well as Ki-67-positive cells, rendering the task even more difficult [8].

The breast biopsies of the data set were scanned using the Hamamatsu NanoZoomer S360MD and the resulting WSIs of this scanner have the .ndpi format. The WSIs can either be seen at the magnification x1.25 (lowest resolution), x2.5, x5, x10, x20, or x40 (highest resolution). Due to memory constraints when loading and working with the images in the RAM, the analysis in this work can not use magnifications above x10. A pathologist, Prof. X. Catteau from the *Centre Universitaire inter Régional d'Expertise en Anatomie Pathologique Hospitalière* (CurePath) located in Charleroi, Belgium, annotated the WSIs to indicate the tumor region and the hotspot inside it but only the tumor region labeling will be used as ground truth to train the DL to segment the tumor region. The corresponding annotations are stored in the .ndpa format.

The annotation corresponding to the WSI present on Figure 5.1 can be found next to it. Such images are presented at low resolution. However, the particular format of such images allows a visualization at different magnifications. Indeed, it is possible to zoom in and capture more granularity. Figure 5.2 show the WSI at a medium resolution, for both a non-tumorous zone and a tumorous zone.

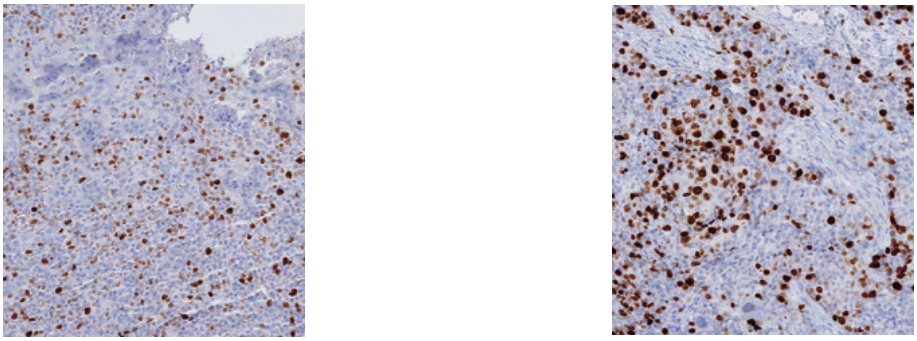


Figure 5.2: Tissue extracts of a non-tumorous region (left) and of a tumorous region (right), both with zoom = x6.35

In the same fashion, it is possible to work at higher resolution with WSIs. On Figure 5.3 are given 2 examples extracted from the same WSI at a higher resolution: one in a non-tumorous region and another in a tumorous region.

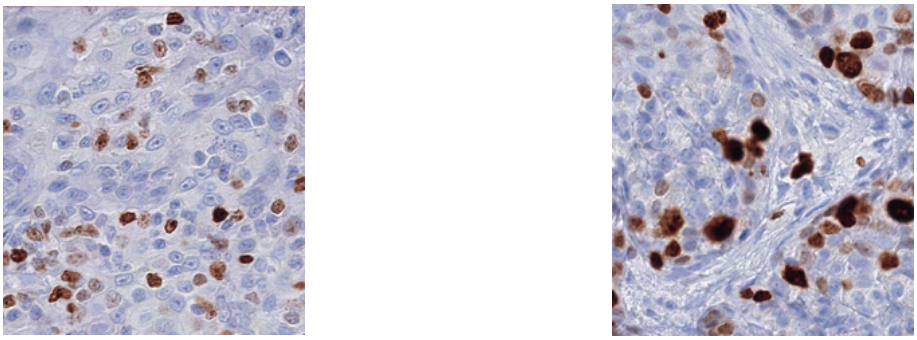


Figure 5.3: Tissue extracts of a non-tumorous region (left) and of a tumorous region (right), both with zoom = x20.94

The data set was originally composed of 153 annotated WSIs. However, there was a severe lack of annotation in the latter, as only representative tumorous regions had been annotated. While tumorous regions were specifically designated, they only accounted for a small fraction of the represented tissue and the rest of the tissue was unlabeled. Considering the unlabeled tissue as being non-tumorous led to no signs of learning for the DL model and it was later decided to work with another data set comprised this time of only 40 annotated WSIs but were the tumor annotations were supposed to be more reliable.

At first glance, it was observed that most images were composed of a majority of glass slide and a minority of tissue. Around half of the tissue seemed to be labelled as tumorous, and the rest of the tissue -unlabelled- was considered non-tumorous. This observation was quantified by computing 2 statistics on the data set: the average fraction of tumor on the WSIs (+/- its standard deviation) and the average fraction of tissue on the WSIs (+/- its standard deviation). Those statistics can be found in Table 5.1.

Data (40 Ki67 WSIs)	Avg. fraction of tumor (in %)	Avg. fraction of tissue (in %)
Original	8.44 ± 7.11	14.42 ± 8.31

Table 5.1: Average fraction of tumorous tissues and tissue (both tumorous and non-tumorous tissue regions) in the images of the entire WSI data set

The visual observations made were confirmed, with only 14.42% of tissue on average on the images and 8.44% of tumor on average on the images. However, it is observed that there is a lot of variance in the data set, as can be seen with the standard deviation of 8.31% and 7.11% for the fraction of tissue and the fraction of tumor, respectively.

This data set needed to be separated into a training set, a validation set, and a test set, as explained in Section 3.4. It was decided that 25% of the data set (10 WSIs) would be used as a test set, and out of the 30 remaining WSIs, 5 would be used as a validation set (cf 5.2).

Data set	Number of WSIs
Training set	25
Validation set	5
Test set	10
Total	40

Table 5.2: Division of the 40 WSIs in the different data sets (training, validation, and test sets)

However, to take into account the inherent variance in the data set, the following approach was carried out to divide the original data set into 3 representative sets. The fraction of tumor on the tissue was computed for each WSI by removing the background (or, conversely, by extracting the tissue mask) and dividing the area of annotated tumorous regions by the area of tissue present in the image.

$$Fraction\,of\,tumor_{tissue} = \frac{Tumor\,area_{image}}{Tissue\,area_{image}}$$

The technique used to extract this tissue mask will be explained in Chapter 6. The Table 5.3 summarizes this by presenting the average fraction of tumor **on the tissue** (after removing the background). It shows that tumors represent around half of the tissue (52.54% on average for the whole data set), but still with a high standard deviation (23.75% for the whole data set) which is in concordance with what was previously said.

The WSIs were thus ranked according to their the percentage of tumor **on the tissue**. First, the test set was extracted by taking the $k * 10th$ percentiles (for $k = 0, 1, ..., 9$). The remaining 30 WSIs were also ranked according to the same criteria and this time the $l * 20th$ percentiles (for $l = 0, 1, ..., 4$) were extracted to constitute the validation set. The remaining 25 WSIs were used as the training set. The Table 5.3 summarizes the mean and standard deviation of the fraction of tumor on the tissue for each set.

	Avg. fraction of tumor (in %)	Avg. fraction of tissue (in %)
A. Training set (25 Ki67 WSIs)		
Original	9.56 ± 7.45	15.58 ± 9.02
+ background removal	54.86 ± 23.07	100
B. Validation set (5 Ki67 WSIs)		
Original	5.98 ± 7.03	11.07 ± 7.16
+ background removal	46.77 ± 27.60	100
C. Test set (10 Ki67 WSIs)		
Original	6.90 ± 6.33	13.19 ± 6.92
+ background removal	49.62 ± 25.36	100

Table 5.3: Average fraction of tumor or tissue regions in the images, with or without background removal, for (A) the training set (B) the validation set, and (C) the test set

As can be seen on Table 5.3, the followed procedure for the separation of the original data set into training, validation, and test sets allowed to have, for each set, a similar mean and variance.

It was found out later that, unfortunately, this new data set contained annotation lacks. The reader is *strongly* encouraged to read Appendix A to understand why/how this annotation problem occured, how it was discovered, and how it impacted the student's work. Indeed, several WSIs presented, on the same slide, serial cuts from the same tissue but from a layer 5µm above with only one cut annotated as being tumorous and the other one left unlabelled even though the tumor should be present on both tissues, despite the 5µm distance. The Figure 5.4 shows an example of this problem on a WSI. As can be observed on the images, the upper tissues are annotated while the serial cuts displayed

below, presenting the exact same shapes and similar textures/visual characteristics are not annotated although they also contain tumor.

Figure 5.4: WSI containing serial cuts (left) presenting the annotation problem on the annotation mask provided (right)

This annotation problem is present in 6 out of the 25 images of the training set, 1 out of 5 images of the validation set, and 4 out of 10 images of the test set. It surely impacted the partition of the WSIs in the different sets, as the concerned WSIs were seriously penalized in terms of percentage of tumor present on the tissue.
Since this annotation problem was identified in August, it was not feasible to retrain the models with a training and validation sets free of those problems due to time constraints. Instead, the performance assessed on the test set was corrected by keeping the current test WSIs but discarding the concerned non-annotated tumorous tissues. Note that after discarding the concerned regions, the average fraction of tumor on the tissue of the test set became $63.90 + -24.93\%$. The test set contains thus a proportion of tumorous tissue greater than in the training or validation set. It is expected, however, that this annotation lack present in some slides of the training and validation sets seriously impacted the training/learning of the model and its performances as, for the concerned regions, the network tried to associate tumorous features to the features supposed to detect the negative class (non-tumorous tissues).

Chapter 6

Methods

The previous chapter detailed the materials that were provided to carry out this work. More particularly, the data set was reviewed, statistics were drawn out of it and used in order to divide the data set into 3 sets (the training, validation and test sets), and a critic of the annotated data was done.
The aim of the present chapter is to describe the methods that were used to attempt achieving the objective of this thesis, i.e. detecting tumor areas in histological sections of breast cancer using deep learning thanks to these materials. To do so, the training and validation sets will be transformed into tiles in order to be fed to the DL architecture that was chosen in this work. The transformation of the WSIs into tiles will be done thanks to a sliding-window approach, and tiles will be generated at a x5 magnification for different contexts (tile sizes of 256x256, 128x128, and 64x64) and at x10 magnification with the corresponding tile sizes incorporating the same contexts (512x512, 256x256, and 128x128). The idea is to assess the importance of context and resolution in the training of a model on IHC Ki67 stained histological sections of breast images and the influence of using overlapping tiles in training data set. As stated in Chapter 4, the context each tile incorporates depends on its size and the magnification. It will thus be important to find the right trade-off between both as a too small tile at a high magnification might miss important contextual information while a too big tile size at low magnification might miss important local information [28].

Those methods are more thoroughly explained in the following sections. Section 6.1 details how the data is prepared into (overlapping) tiles ; Section 6.2 details what augmentation techniques are used to cope with the lack of annotated data ; Section 6.3 details the DL architecture used, how the models are trained and how to compare and choose the best model(s) ; Section 6.4 details the post-processing techniques that can be used during the inference process when testing the model on the test set ; finally, Section 6.5 presents the training and inference pipelines.
The next chapter, Chapter 7, will detail the results obtained with the help of all the aforementioned methods.

6.1 Data pre-processing/preparation

The first step of the whole training process is to have usable images of dimensions acceptable by the DL network. As discussed in Chapter 2, the dimensions of WSIs are significantly bigger than what DL architectures can support.

To counter the scarcity and the dimensionality problem of the data, all the previous works [8] [27] [24] [28] [12] adopted a **sliding-window** approach, i.e. the tiling of the original image into tiles of smaller dimensions. That way, the size of the input image that will be provided to the neural network will be small enough and there will be more input data. However, it should be noted that the overall information contained in the training data remains the same. This sliding-window approach will thus be used in this work. Moreover, the background mask, or conversely the tissue mask, can be easily determined thanks to standard image processing techniques. Such mask will be used in conjunction with the sliding window technique to only extract tissue tiles from the original WSIs and restrict the learning and inference process only to tissue regions. Indeed, the real difficulty lies in discriminating between tumorous and non-tumorous tissues. Thus, in order for the NN to specialize in this, it is better to feed the neural network only with tiles that contain tissue in the training pipeline and to discriminate between the glass slide (background) and tissue (containing both tumorous and non-tumorous regions) thanks to standard image processing. In a similar fashion, such tissue mask will be used later during the inference pipeline to speed up the inference time as non-tissue regions will not need to be predicted by the network. This was also done in previous works [8] [12] [27]

6.1.1 Tissue extraction

Previous works [8] [12] [27] applied pre-processing to the original WSI to obtain a tissue mask and limit their work on the information contained in this zone of the image. Both [12] and [27] used Otsu thresholding. [12] does not specify on which channel it used it but [27] first converted the RGB images to HSV and then applied the Otsu threshold on the saturation channel.
Otsu thresholding is an image processing technique used to perform automatic image thresholding an input image. Here, the goal is to separate the tissue from the glass slide. The Otsu threshold looks at the histogram of the studied channel values and finds the value that best separates the pixels into 2 classes (background and foreground) by maximizing their inter-class variance [32].

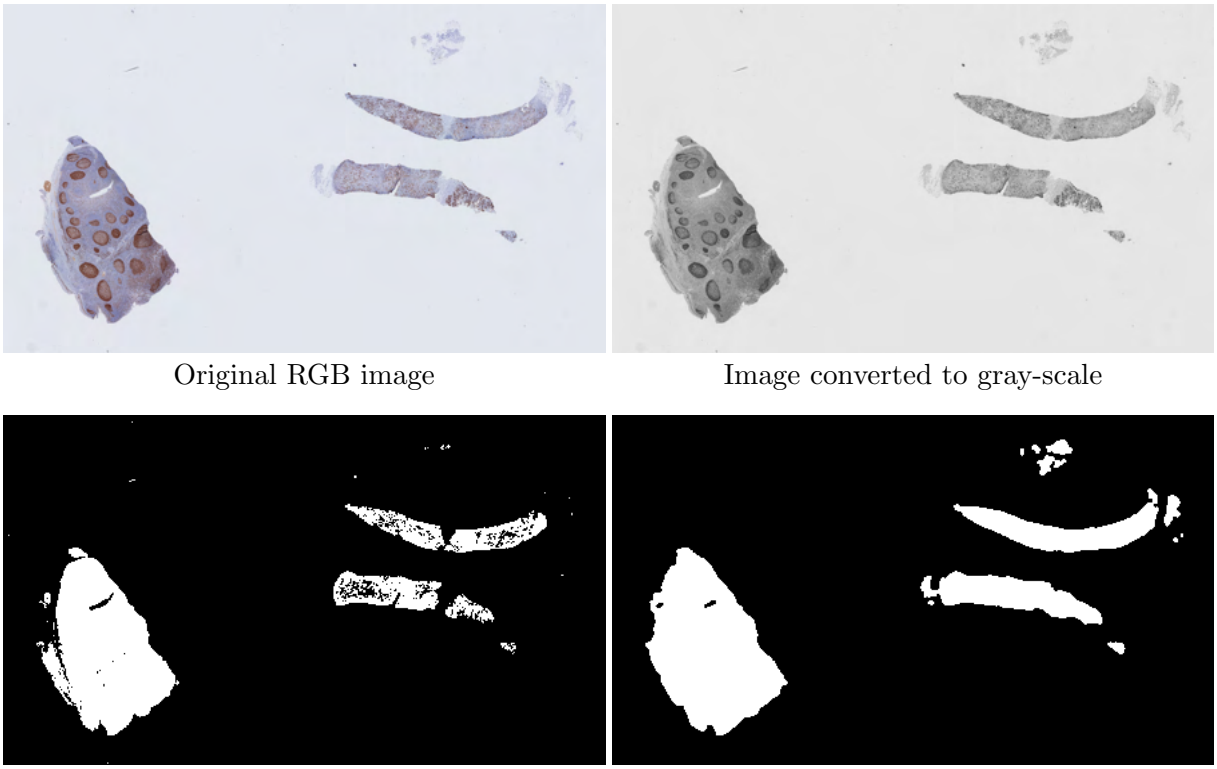
To obtain the tissue masks in this work, the original WSIs were first extracted at the lowest resolution (x1.25 magnification) since the tissue is a high level feature. Then, since Otsu thresholding only works on one channel images, the WSIs were converted from RGB to gray-scale images with the following conversion for each pixel: $Y = 0.2125R + 0.7154G + 0.0721B$, where Y denotes the final gray-scale pixel value, R the red value of

the pixel, G the green value of the pixel, and B the blue value of the pixel [1]. Y ranges from 0 (black) to 255 (white). At this point, the Otsu threshold t_{Otsu} is computed with the expectation that the white class (glass slide) will be separated from the gray class, i.e. the tissue. When applying the Otsu threshold on the gray-scale image, the result is a binary mask with Truth values where there is tissue, and False values where there is background, as the following condition states:

$$tissue_{mask}(x, y) = \begin{cases} True & \text{if } gray - scale\, WSI(x, y) < t_{Otsu} \\ False & \text{otherwise} \end{cases}$$

The result of this operation can be noisy. This noise needs to be removed and to do so, morphological operations such as *opening* and *closing* were applied with a disk size of 10 to clean the noise in the tissue mask obtained with Otsu. On the one hand, the morphological closing operation fills small holes in the tissue mask that fit the structuring element (the aforementioned disk) while keeping the big holes intact. On the other hand, the morphological opening operation removes the small objects in the background while preserving objects larger than the structuring element [33]. For more information about morphological operations, the interested reader should refer to [34].
The Figure 6.1 illustrates the previously mentioned steps for the tissue extraction of a WSI.

[1] Done using rgb2gray from the scikit library `https://scikit-image.org/docs/dev/api/skimage.color.html#skimage.color.rgb2gray`



Original RGB image | Image converted to gray-scale

Tissue mask generated with the Otsu threshold | Tissue mask cleaned thanks to morphological operations

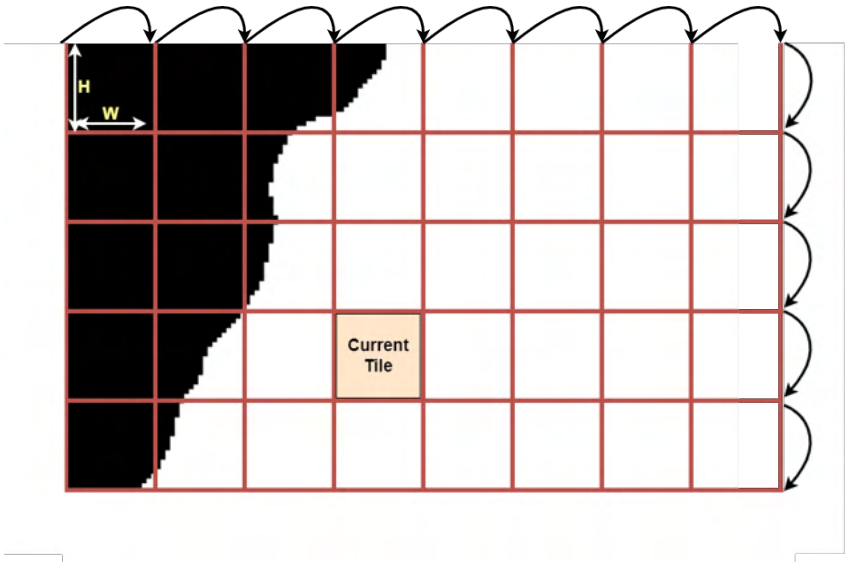
Figure 6.1: Generation of a tissue mask thanks to the Otsu threshold. The original RGB image is first converted into gray-scale. The Otsu threshold computed on the gray-scale image is used to generate a binary mask. Morphological operations are finally used to clean the noise in the mask

The resulting tissue mask is the one that was used to compute the data set statistics in Section 5.2, more specifically to compute the area of tissue on the images. This tissue mask will be useful for the training process as it will allow to only select tiles in the tissue regions, but it will also be useful for the inference process as only tiles in the tissue regions will need to be predicted.

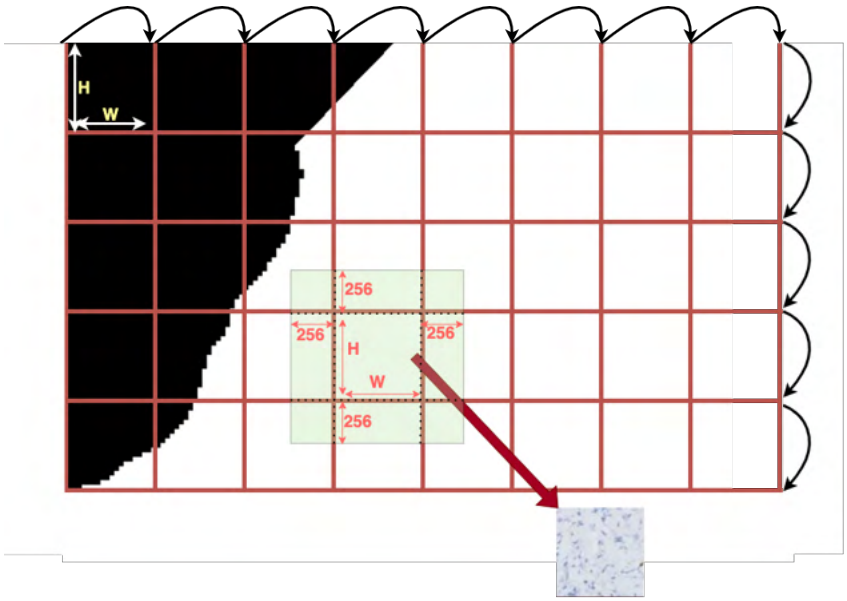
6.1.2 Sliding-window

With the sliding-window approach, the original image is divided into tiles of width W and height H significantly smaller the original image. There are two ways to tile the original image: either **with** or **without** overlapping tiles. In this work, the sliding-window was used on the training set and validation set to generate tiles. A similar approach will be used to evaluate the optimized network performance on the test set but will be explained later in Section 6.4. One parameter that will be studied in this work is the effect of using overlapping tiles (with a 50% overlap) or not *on the training set*.

Without overlap, once a tile size $(W * H)$ is defined, the tissue mask of the WSI is travelled along its width and length by steps of W and Y, respectively. An illustration of this tiling is shown in Figure 6.2.



Sliding window: condition on the tissue



Sliding window: condition on the annotation mask

Figure 6.2: Illustration of the sliding-window principle. The first illustration (above) illustrates the condition on the tissue. A test is conducted on the current tile to assess if it is a tissue tile. The second illustration (below) illustrates the condition on the annotation mask. A test is conducted on the current tile to assess if it is far enough from the tumor border thanks to the contextual tile. Here, the tile satisfies both conditions and the corresponding (positive) tile is extracted from the original WSI along with its annotation

Tiles encompassing at least 85% of tissue, i.e. $\frac{Area\,tissue_{current\,tile}}{W*H} > 85\%$ are considered tissue tiles. If the encountered tile in the tissue mask satisfies this condition, as it is the case on Figure 6.2, another test is made on the tile to segregate it between *core* tumor tile, tumor tile close to the annotation border, non-tumor tile *far* from the annotation border, and non-tumor tile close to the annotation border. The distinction between the tiles is made due to the fact that it was proven in previous studies that different pathologists tend to give different annotations for a same WSI and have a hard time to reach a consensus and agree on a diagnosis [10] [9]. To be able to still work with the data provided but to take this factor into account, the following hypothesis was done: the tissue regions close to the border of the annotation are prone to discussion and should thus be discarded. Hence, only the *core* tumor tiles (positive tiles) and non-tumor tiles (negative tiles) *far* from the annotation border are kept. So, once a tile is considered a tissue tile, another condition is tested to determine whether it should be discarded or not. For a candidate tile of size WxH, a contextual tile of size $(W + 512) * (H + 512)$ at x10 magnification is constructed and centered on the candidate tile. If the magnification is x5, the contextual tile is of size $(W + 256) * (H + 256)$ so that it encompasses the same context than its x10 magnification counterpart.
For a tile to be considered a *core* tumor (positive) tile, the whole contextual tile needs to be located inside the annotation mask. In a similar fashion, for a tile to be considered a non-tumor (negative) tile *far* from the annotation border, the whole contextual tile needs to be located outside the annotation mask. The Figure 6.2 illustrates this idea of contextual tile on the annotation mask.

The coordinates of the candidate tiles satisfying these conditions are retrieved and their corresponding tiles are extracted from the original WSI image. That way, non-overlapping tissue tiles are generated.

Similarly, it is possible to use the sliding-window to generate overlapping tissue tiles. Only this time, the tissue mask and the annotation masks are travelled along their width and height with steps of $W/2$ and $H/2$, respectively, to get an overlapping of 50%. The net effect of using this overlapping is to generate more or less 4 times more tiles than the non-overlapping sliding window approach. The information inside the overlapping zone is not used to improve the training output in this work. Instead, it is used as a natural data augmentation technique. Note, however, that this overlapping mechanism does not create any new information, it only represents the same total information in a different way and with repetition of information.

The Table 6.1 summarizes the result of the sliding-window approach (with *and* without a 50% overlap for the training set) in terms of number of positive or negative tiles generated. This data preparation was done at 2 different magnifications (x5 and x10) and at different tile sizes encompassing 3 different contexts: a large context (mag. x5 256x256 tiles and mag. x10 512x512 tiles), a medium context (mag. x5 128x128 tiles and mag. x10 256x256 tiles), and a small context (mag. x5 64x64 tiles and mag. x10 128x128 tiles). At x5 magnification and at x10, the scales are $1.8416\mu m$ per pixel and $0.9208\mu m$ per pixel, respectively. This means that the large, medium and small contexts encompass an area of $2.2226 * 10^{-1} mm^2$, $5.5567 * 10^{-2} mm^2$, and $1.3892 * 10^{-2} mm^2$, respectively.

The idea is to study the influence of the context and the resolution, and determine what setting is favourable and yield to the best results.

The Figure 6.3 displays side by side examples of tiles encompassing the same context but at the 2 different magnification: x5 and x10.

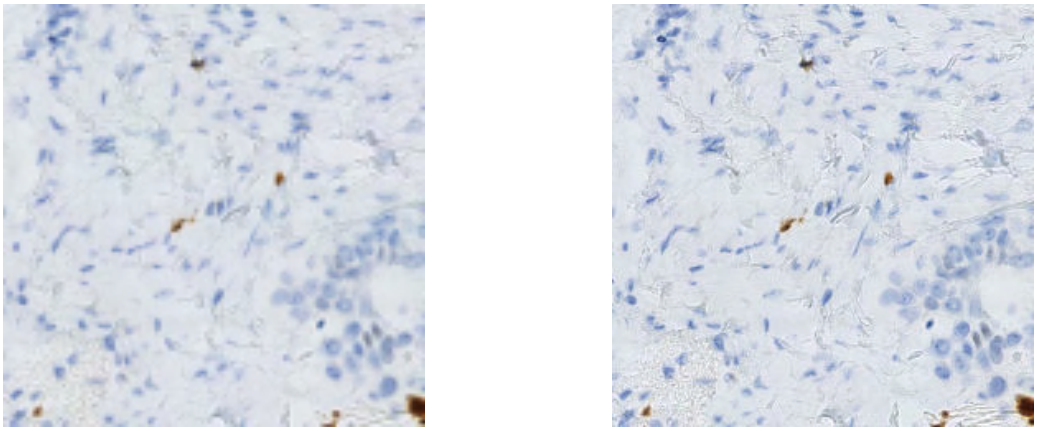


Figure 6.3: 2 tiles encompassing the same context but at different magnifications (resolutions): tile of size 128x128 pixels at x5 magnification (left) and tile of size 256x256 pixels at x10 magnification (right)

Set	Positive tiles	Negative tiles	Total tiles
Large context (mag. x5 256x256 tiles and mag.10 512x512 tiles)			
Training	170	884	1054
Training (with overlap)	704	3548	4252
Validation	19	228	247
Medium context (mag. x5 128x128 tiles and mag.10 256x256 tiles)			
Training	1564	4427	5991
Training (with overlap)	6206	17692	23898
Validation	176	1075	1251
Small context (mag. x5 64x64 tiles and mag.10 128x128 tiles)			
Training	8368	20073	28441
Training (with overlap)	33471	80277	113748
Validation	1011	4778	5789

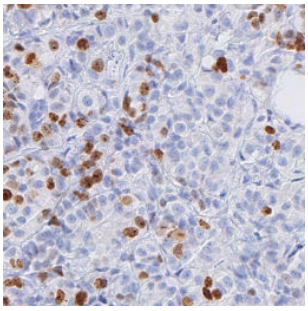
Table 6.1: Number of tiles generated by the sliding window approach on the WSIs at x5 and x10 magnification for different contexts: large (mag. x5 256x256 tiles and mag.10 512x512 tiles), medium (mag. x5 128x128 tiles and mag.10 256x256 tiles), and small (mag. x5 64x64 tiles and mag.10 128x128 tiles). For the training set, the sliding window was done once without overlap and once with a 0.5 overlap

One particular observation is that for a big context, only 19 positive tiles are generated. This is extremely limited and the results of the models obtained with this context should be interpreted with extra care. Something else that stands out from Table 6.1 is that, for all contexts, there is a smaller proportion of positive tiles generated in the validation set compared to the proportion of positive tiles generated in the training set. This must be due to the way the data set was partitioned into training, validation, and test set according to the proportion of tumor encountered on each WSI, as described in Chapter 5. This did not take into account how tiles would be generated. Indeed, the contextual tile condition defined earlier is more restrictive for the positive tiles, as it requires positive tiles to be at the core of the tumor while negative tiles only need to be far enough from the tumor border but can be located at the tissue extremities.

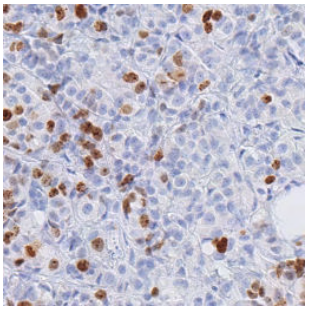
6.2 Data augmentation

All the tiles given as training data in the network will be prone to on-the-fly stochastic data augmentation, each time the training loop is fed with a new batch. All previous works [8] [12] [28] [27] [24] made use of this technique as a way to overcome the scarcity of annotated data by enriching the diversity of the training samples. Since the data augmentation is done stochastically on-the-fly, i.e. it is done randomly each time a batch is fed to the network for further training, the number of augmented tiles for a training depends highly on the number of epochs accomplished by the training process.

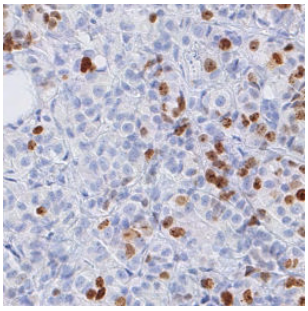
In the same fashion as previous works, similar traditional augmentation techniques were used in this work: horizontal and vertical flipping, random rotations between 0 to 45°, random gamma correction to slightly change the brightness in the image, and noise introduction. All those data augmentation techniques are illustrated in Figure 6.4, where they are individually presented.



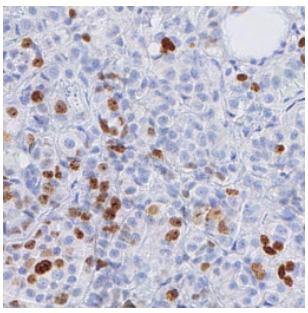
Original tile (OT)



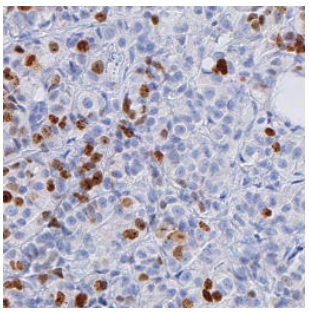
OT & vertical flipping



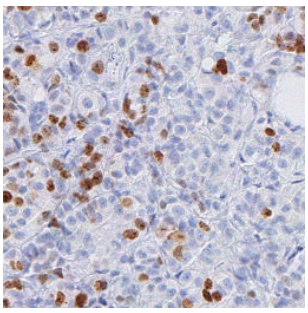
OT & horizontal flipping



OT & rotation



OT & gamma correction



OT & noise introduction

Figure 6.4: Data augmentation techniques used: vertical/horizontal flipping, rotation, gamma correction and noise introduction

6.3 Model: architecture, training and evaluation strategy

This section aims at explaining the DL architecture used in this work, how it was trained on the training set, how the best model will be chosen based on the validation set, and how its performance will be assessed on the test set.

6.3.1 Architecture

The segmentation DL architecture chosen for this work is a U-Net, as this model (or similar) showed conclusive results in previous works and performs remarkably in biomedical image analysis in general [12] [13].

The original U-Net architecture was introduced in Chapter 3.3.1 and represented in Figure 3.11. It used unpadded convolutions, leading to an output segmentation map size smaller than the input image size. In the present work, padded convolutions were used such that the output size of the network is of the same size than the input size. That way, the reconstruction of the full image from the predicted tiles is facilitated, with

the methods described in Section 6.4. Moreover, the ReLU activation function after each convolution was replaced by the *leaky ReLU* activation function. The leaky ReLU is defined as:

$$g(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if} x \leq 0 \end{cases} \tag{6.1}$$

Using this activation function with $\alpha = 0.3$ (default Tensorflow value) over the precedent one has the benefit of avoiding the "dying" ReLU problem, a problem occuring when a neuron is not activated anymore due to negative values [16].
The subsequent modified U-Net architecture is presented in Figure 6.5 for an input RGB image tile of size $W * H$.

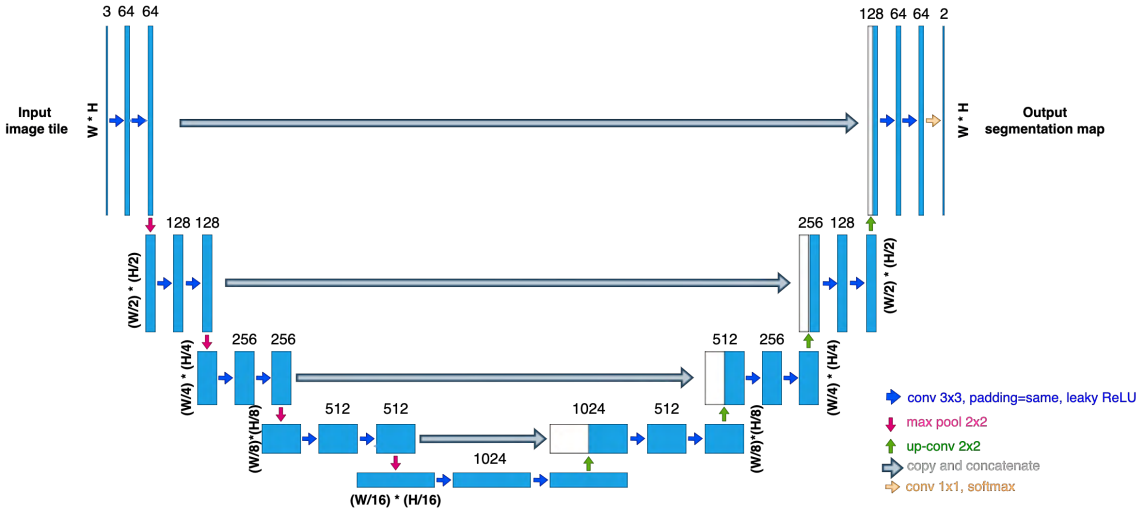


Figure 6.5: Modified U-Net architecture. Illustration inspired by [13]

At the final layer, a soft-max activation function was used to get, for each pixel of the output map, a probability of belonging to either the non-tumor class ("negative" class) or the tumor class ("positive" class). That way, the probability map can be displayed if needed, and applying a 0.5 threshold on the probability of belonging to the positive class gives the segmentation mask predicted by the network.
In total, there are 34,513,410 trainable parameters in the network.

6.3.2 Training of models

Models studied

Different models will be trained with the same U-Net architecture presented in Section 6.3.1, but with training and validation tiles of different context and resolution. The idea is to study :

- the importance of resolution in the training: is there an advantage, in addition to having a faster inference, to work at a lower resolution (x5) ? Or is there a

significant increase of performance observed when working at a higher resolution (x10) ?

- the importance of the context in the training: it better for the tiles to encompass a big context to take into account large tissue structures ? Or are the features necessary for a good segmentation better learned on a small or medium context ?

To do so, models trained on tiles with the same context but at different resolution (x5 and x10) will be trained and compared. The *smallest* context is studied here either at x5 magnification and tile size of 64x64 or at x10 magnification and tile size of 128x128 ; the *medium* context is studied either at x5 magnification and tile size of 128x128 or at x10 magnification and tile size of 256x256 ; the *largest* context studied is either at x5 magnification and tile size of 256x256 or at x10 magnification and tile size of 512x512. Unfortunately, for the largest context, it was **not** possible to train a model on tiles of 512x512 at x10 magnification due to GPU constraints. This large context will still be studied, but only at x5 magnification.

Another factor that will be studied is the potential benefit of using overlapping tiles in the training set. A new cohort of models with all the aforementioned resolution and contexts will be trained but this time on all their respective training set comprised of overlapping tiles. However, in this work the overlap in the tiles is not used to improve the output of the network during the training. Instead, the overlapping tiles are treated independently and serve mainly as a form of data augmentation.

Training batches
During the training process, the batch size is set to 16. Each batch is comprised of an even number of positive tiles and negative tiles to avoid any problem due to class imbalance. Indeed, as summarized in the Table 6.1, there is a significant class imbalance towards the negative class at any resolution or context. As done in [27], at each epoch, all the positive tiles are used and a random sampling is thus used to select an equivalent number of negative tiles. Following the work of [24] the 8 positive tiles and the 8 negative tiles of each batch are randomly augmented, as presented in Section 6.2 : there is a 50% chance of vertical flipping, 50% chance of horizontal flipping, 50% chance of being rotated from angle $x \in [0, \pi/4]$ (x is also randomly chosen). Random gamma correction to adjust the brightness and random noise following a Gaussian distribution are also added to each tile of the batch.

Training parameters
The training was done using the Adam optimizer, a stochastic gradient descent method [19]. The default Tensorflow values of the optimizer were used, with the exception of the learning rate which was set in this work to $\alpha = 0.0001$, as used in [12].

Loss function
As the U-Net outputs, for each pixel, probabilities of belonging to each of the 2 classes (positive or negative), the loss function used in this work is the sparse categorical cross-entropy from the TensorFlow library (sparse because the classes are provided as integerers: 0 for the negative class and 1 for the positive class).

Training monitoring
The maximum number of epochs set for the training process is 100. However, the cross-entropy loss and MCC are monitored during the training and, after each epoch, they are also computed on the validation set. An early stopping of $patience = 10$ is set on the loss function, i.e. the learning stops automatically if the loss score of the model at the last epoch is not lower than any of the 10 previous scores on the validation set: the loss score reached a plateau, indicating that the model seems to have stopped improving on the validation set. Plots of the evolution of the loss and the MCC across the different epochs are generated for both the training and the validation and the weights of the model that are saved are those of the epoch that led to the minimum observed cross-entropy loss on the validation set.

6.3.3 Evaluation strategy

Each model is trained at each resolution and context, with and without overlapping tiles in the training set. For each saved model, the performances on the validation set are computed thanks to the following metrics: the loss, the accuracy, the precision, the specificity, the recall, the F1-score, the intersection over union, and the Matthews correlation coefficient. For the segmentation task of this work, the MCC will be favored. The advantages of using this metric over the others were outlined in Chapter 3. So, when comparing the performances of the different models on the validation set, the decision of which model to select will be mostly based on this metric, while the other metrics will serve as support to better understand how the models behave.

However, the validation performance is not a true indicator of the model performance since the model was optimized in relation to the validation set. The true model performance is assessed on previously unseen data, i.e. the WSIs that were put aside in the test set [17]. This time, the metrics will not be computed tile per tile and then averaged, but will be computed on the whole predicted tissue region. Section 6.4 details the post-processing techniques that can be used to go from the tile prediction to the WSI prediction. Usually, in ML, the best model in terms of validation performance is retrained with the same parameters on the training *and* validation sets before its performance assessment is made on the test set (cf. Section 3.4). However, it was decided in this work to keep the model trained only on the training set since stochastic events happen in the learning procedure and early stopping was used. The addition of the validation tiles in the training set might lead to a slower training curve with more epochs needed to reach a plateau. However, for the comparison to be fair, the training would need to be stopped at the same epoch than during the training on the training set only, which *could* be sub-optimal.

6.4 Post-processing

During the training and the validation, the model metric performances were assessed on the tiles themselves by comparing the ground truth with the corresponding predicted tile mask directly. Such mask is obtained by applying a 0.5 threshold on the probability map of belonging to the positive class. This section, however, describes the post-processing used in the inference pipeline used for evaluating the model performances on the test set. This time, the metrics are evaluated on the whole predicted region (the tissue region) of the WSI. To go from the tile prediction to the WSI prediction, post-processing needs to be done. This can either be done thanks to the stitching of the tile predictions, with or without overlap. Both approaches will be investigated and their results compared to assess whether overlapping predictions improve the segmentation metric results.

6.4.1 Stitching of the tile predictions

The stitching of predicted tiles **without** overlap is done in a completely analogous way non-overlapping tiles were generated thanks to a sliding-window restricted to the tissue areas, i.e. the WSI to predict is travelled across by steps of W in its width and H in its height (for a model trained with tiles of size $W * H$), the non-tissue tiles are by default not considered but instead of extracting the tissue tiles, those are predicted by the model and stitched together by simply assigning to the predicted tile the coordinates of the tile it just has predicted. If the information retrieved for each pixel is its probability of belonging to the positive (tumor) class, the result is a probability map of the positive class of the same size than the input WSI. Previous works [8] [12] [28] adopted this inference approach and applied a 0.5 threshold to get a predicted segmentation mask. For an example of prediction using this post-processing technique, please refer to Appendix B.

6.4.2 Stitching of overlapping tile predictions

The second - similar - way of predicting a complete WSI was done in previous works [27] [24] by stitching overlapping tile predictions together. In this work, this was implemented in the following manner for a model trained on tiles of size $W * H$. The stitching of predicted tiles **with** overlap reuses the previous technique, but applies it 4 times to the WSI, each time initializing the tiling process at a different starting point: $(0; 0), (W/2; 0), (0; H/2),$ *and* $(W/2; H/2)$. Then, for a WSI of size $W_{WSI} * H_{WSI}$, the final prediction $\hat{Y}_{overlap}$ is a $W_{WSI} * H_{WSI}$ matrix where each of element is the element-wise average of the corresponding elements of the 4 resulting probability maps ($\hat{Y_1}$, $\hat{Y_2}$, $\hat{Y_3}$, and $\hat{Y_4}$ respectively) of the same size than the input WSI. This average is computed by taking the sum of the 4 probability maps and doing the element-wise division with $D(x; y)$, the matrix that takes into account the number of times each pixel was predicted through this process, also of size $W_{WSI} * H_{WSI}$. Mathematically, this gives:

$$\hat{Y}_{overlap} = (\hat{Y_1} + \hat{Y_2} + \hat{Y_3} + \hat{Y_4}) \oslash D \tag{6.2}$$

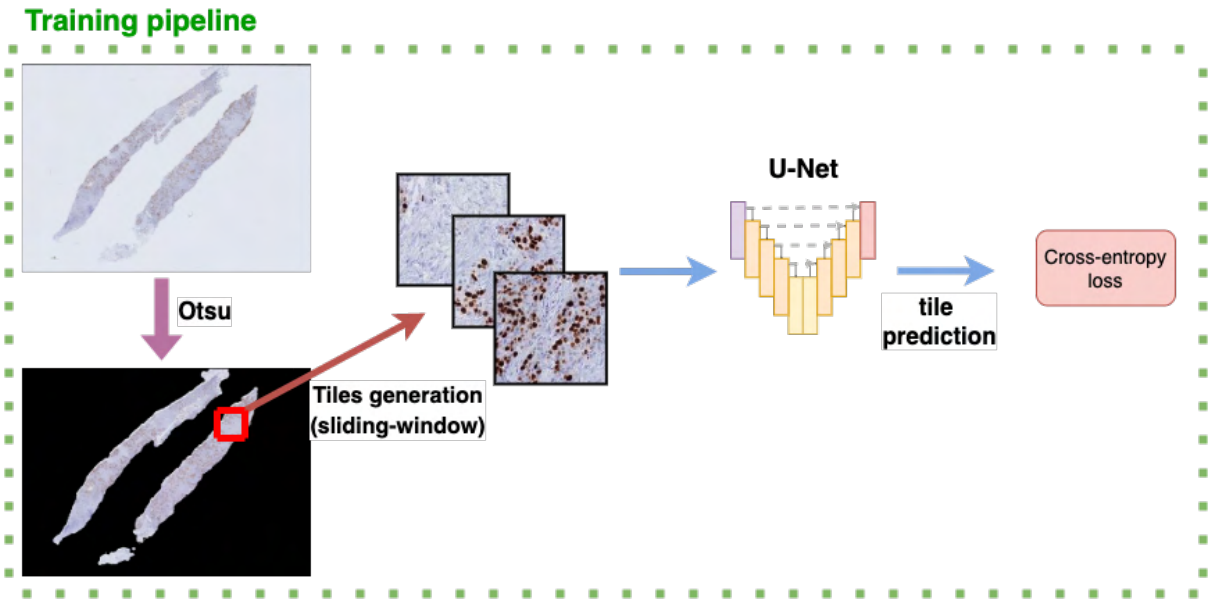
where $\oslash$ denotes the element-wise division and D is defined as :

$$D(x;y) = \begin{cases} 1 & \text{if } (x < \frac{W}{2} \wedge y < \frac{H}{2}) \vee (x < \frac{W}{2} \wedge y > H_{WSI} - \frac{H}{2}) \\ & \vee (x > W_{WSI} - \frac{W}{2} \wedge y < \frac{H}{2}) \vee (x > W_{WSI} - \frac{W}{2} \wedge y > H_{WSI} - \frac{H}{2}) \\ 2 & \text{if } (x < \frac{W}{2} \wedge \frac{H}{2} < y < H_{WSI} - \frac{H}{2}) \vee (x > W_{WSI} - \frac{W}{2} \wedge \frac{H}{2} < y < H_{WSI} - \frac{H}{2}) \\ & \vee (\frac{W}{2} < x < W_{WSI} - \frac{W}{2} \wedge y < \frac{H}{2}) \vee (\frac{W}{2} < x < W_{WSI} - \frac{W}{2} \wedge y > H_{WSI} - \frac{H}{2}) \\ 4 & \text{otherwise} \end{cases} \tag{6.3}$$

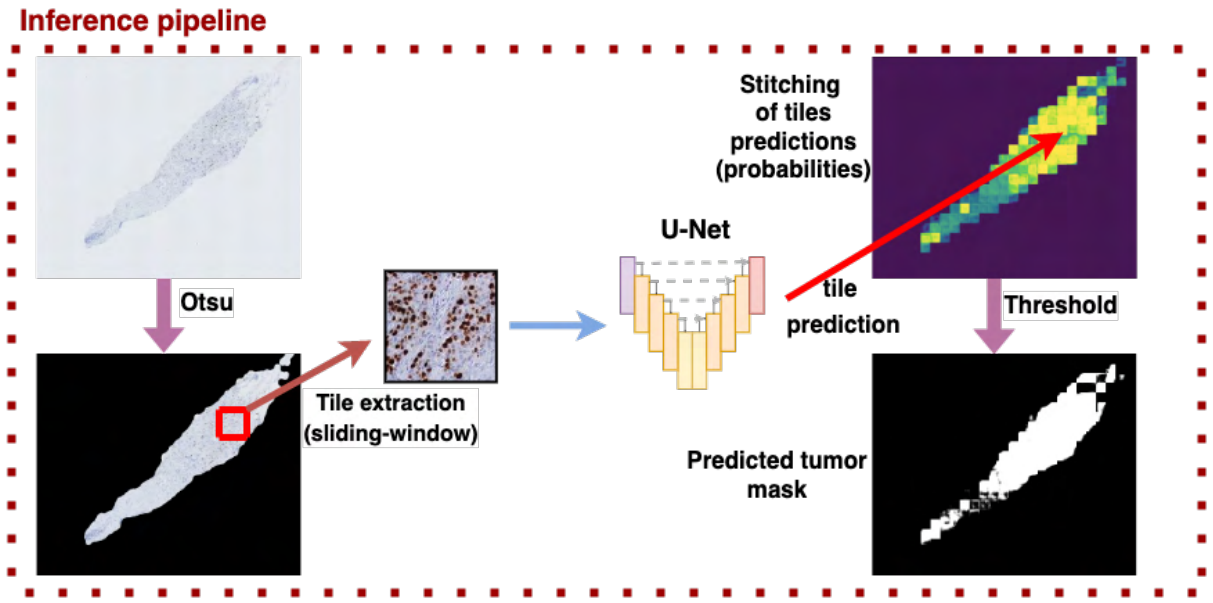
The effect of this post-processing technique, compared to the non-overlapping version, should be to smooth the probability maps. However, this comes at the cost of multiplying the inference time by a factor 4, so its effect should be assessed to determine if the effect it has on the segmentation metrics is worth the fourfold inference time. Again, to retrieve the predicted segmentation mask, a 0.5 threshold must be applied on the average probability map. For an example of prediction using this post-processing technique, please refer to Appendix B.

6.5 Training and inference pipelines

The training and inference pipelines of this work using the methods described previously are illustrated in Figure 6.6.
For the training pipeline, the process is the following: the WSIs of the training set and their corresponding annotation masks are loaded at either at x5 or x10 magnification; corresponding lower resolution WSIs (at x1.25 magnification) are also loaded and converted to gray-scale to then extract their tissue masks thanks to the Otsu threshold and morphological operations; a sliding-window approach (with or without a 50% overlap) is applied on each WSI to extract tumorous (positive) and non-tumorous (negative) tissue tiles (encompassing a large, medium, or small context) at a certain distance of the annotation borders; at each epoch, all the positive tiles and a same number of randomly sampled negative tiles are fed to a U-Net by balanced batches of 16 tiles which are randomly augmented with classical data augmentation techniques; the network tries to minimize the categorical cross-entropy loss function and assesses its performance on the validation tiles (generated in the same fashion but without overlap and without data augmentation) at the end of each epoch; the maximum number of epochs is 100 but the training stops if no new minimum of the loss function is found on the last 10 epochs.
For the inference pipeline, the WSI to be predicted follows a similar beginning only the network is not trained but used to infer each tile independently. Post-processing is applied on the predicted tiles by stitching them together to obtain a positive probability map. If overlapping tiles were used, the probabilities in the overlapping zones are averaged. Finally, a threshold of 0.5 is applied the the positive probability map to a extract a predicted tumor mask.



Training pipeline



Inference pipeline

Figure 6.6: Training & inference pipelines

Chapter 7

Results

In the present chapter, the results of the training of the models using different resolutions and contexts, with or without overlapping tiles in the training set, that were outlined in Section 6 are presented and compared. The models are first compared according to their segmentation metrics on the validation set. Then, for the best model(s) selected, predictions on the test set are carried out with or without overlapping tile predictions. Segmentation metrics are computed to quantitatively assess the performances of the model(s), but prediction masks and probability maps are also presented to offer a qualitative analysis of the results. Finally, the results are discussed.

7.1 Models comparison, selection and testing

The first models were trained on the tiles of the training set generated *without* overlap. Their average performance on the individually predicted tiles of the *validation* set are shown on Table 7.1.

Then, the same models were trained on the tiles of the training set generated *with* overlap. Their average performance on the individually predicted tiles of the *validation* set are shown on Table 7.2.

Model	Loss	Acc	P	Sp	R	F1	IoU	MCC
Small context								
mag x5, size=64x64	0.5154	73.84%	**73.32%**	**70.40%**	77.28%	**78.60%**	**65.70%**	0.4871
mag x10, size=128x128	0.5421	72.24%	66.51%	50.78%	**93.69%**	72.35%	57.70%	0.4951
Medium context								
mag x5, size=128x128	**0.4741**	**77.83%**	72.32%	62.72%	92.80%	73.80%	59.41%	**0.5864**
mag x10, size=256x256	0.5082	75.15%	69.84%	60.31%	89.91%	75.86%	61.77%	0.5273
Big context								
mag x5, size=256x256	0.6547	61.81%	59.14%	45.98%	77.64%	65.65%	49.24%	0.2514

Table 7.1: **Validation metrics** of the different models for different contexts (small, medium, and large) and resolutions (x5 and x10 magnifications), trained on the tiles of the training set generated **without** overlap. The best results are shown in bold.

Model	Loss	Acc	P	Sp	R	F1	IoU	MCC
Small context								
mag x5, size=64x64	0.5366	72.78%	70.33%	64.20%	81.35%	78.60%	65.66%	0.4706
mag x10, size=128x128	0.5414	72.58%	72.58%	71.09%	74.08%	**80.60%**	**68.42%**	0.4604
Medium context								
mag x5, size=128x128	0.4564	**79.89%**	**83.70%**	**84.69%**	75.02%	74.32%	60.17%	0.6075
mag x10, size=256x256	0.4973	75.31%	71.81%	64.45%	86.12%	76.37%	62.37%	0.5215
Big context								
mag x5, size=256x256	**0.4429**	78.29%	70.02%	57.14%	**99.45%**	77.53%	64.04%	**0.6246**

Table 7.2: **Validation metrics** of the different models for different contexts (small, medium, and large) and resolutions (x5 and x10 magnifications), trained on the tiles of the training set generated **with** overlap. The best results are shown in bold.

From Table 7.1 and Table 7.2, it can be concluded that - with the data set provided - the models trained with tiles characterized by a small context lead to unsatisfactory performances in terms of MCC on the validation set, whether overlapping tiles are used in the training set or not. The medium and big contexts however benefit greatly from the strategy of the sliding-window with overlapping tiles for the training set. This could be explained by the fact that the number of tiles generated with a small context were already consequent, even without the overlapping strategy (cf Table 6.1). Another explanation for the bad performance of the small context models is that it seems that the features necessary to distinguish between a tumorous tissue and a non-tumorous tissue on the data provided are located at a higher level than the small context defined in this work. The latter contains more local information but the features learned from them by the model do not perform as well as those from the bigger contexts.
It can also be observed on Table 7.1 and Table 7.2 that the model trained on a big context benefits from a substantial increase in all its metrics when training on overlapping tiles. This is explained by the fact that the number of training samples were initially insufficient without the overlapping strategy: only 170 positive tiles and 884 negative

tiles to train on. When training with overlapping tiles, those numbers rose to 704 and 3548, respectively (cf Table 6.1). However, as discussed in Section 6.1.2, one must be cautious when reading the performances of this model on the validation set because this set consists of only 19 positive tiles and 19 randomly sampled negative tiles.

Based on the MCC metric, 2 models stand out from the rest of the cohort, with better results on this metric than the other models. Both of those models are trained with a sliding-window *with* overlapping tiles and both operate at a x5 magnification. Those 2 models are: the model trained on a medium context at x5 magnification (tile size of 128x128) which will be referred from now on as **Model A** and the model trained on a big context also at x5 magnification (tile size of 256x256) which will be referred from now on as **Model B**.
To sum up the performances of the models on the validation data, it stands out that: a) the small context does not capture enough context to segment well the tumorous tissues from non-tumorous tissues, b) the other models encompassing a medium and a big context benefit from the overlapping strategy, and c) working at a lower resolution leads to better segmentation performances in this work.

Model A and B show nearly identical MCC metrics, with Model B showing a slightly better performance ($MCC = 0.6246$) against Model A ($MCC = 0.6075$). However, in light of the annotation lack problems that were discovered recently in the data set provided but more specifically here in the validation set (cf. Section 5.2), one must be cautious with those results. Furthermore, Model B has a recall excessively high ($R = 99.45\%$) with a low specificity ($Sp = 57.14\%$). Thus, Model B seems to be really good at detecting the positive class but seems bad detecting the negative class. Model A, on the other hand, shows the best specificity metric ($Sp = 84.68\%$) and the best precision metric ($P = 83.70\%$) from the cohort, while having a relatively acceptable recall ($R = 75.02\%$). Therefore, it seems that Model A is good at predicting the positive class while being relatively good at predicting the negative class too. For those reasons, coupled with the fact that - as discussed previously - the validation performances of Model B should be handled with extra care since the number of validation tiles used to validate this were extremely low, it was decided to exclude Model B and only retain Model A, i.e. the model working at x5 magnification and a tile size of 128x128 pixels encompassing what was defined as a *medium* context.

The Table 7.3 shows the mean and standard deviation of the prediction results on the test set (the metrics are computed on the entire tissue regions of each WSI) of the selected model (Model A) with different post-processing (see Section 6.4): once without the stitching of *non-overlapping* tile predictions (from now on refered to as **Model** A_1) and once with the stitching of *overlapping* tile predictions (from now on refered to as **Model** A_2).

Post-processing used	Acc	P	Sp	R	F1	IoU	MCC
Model A: mag x5 , size 128x128, sliding window with overlap							
Non-overlapping pred.	$65.30 \pm 19.46\%$	$78.97 \pm 22.86\%$	$68.94 \pm 26.34\%$	$62.93 \pm 27.27\%$	$67.14 \pm 20.95\%$	$53.75 \pm 23.01\%$	0.3112 ± 0.3136
Overlapping pred.	$64.68 \pm 21.05\%$	$78.53 \pm 25.80\%$	$71.35 \pm 26.28\%$	$60.17 \pm 30.78\%$	$64.62 \pm 24.49\%$	$51.85 \pm 25.74\%$	0.3117 ± 0.3317

Table 7.3: **Test metrics** of the selected model (Model A) computed on the 10 WSIs of the test set, once without (Model A_1) and once with (Model A_2) overlapping tiles in the post-processing phase

A first question is to ask oneself is whether a post-processing technique is better than the other. As can be observed on Table 7.3, the use of overlapping tile predictions in the inference process do not lead to any global improvement trend on the metrics' averages, and each metric remains with a high standard deviation. The high standard deviation observed indicates a high variance in the results of the predicted WSIs. The Table 7.4 compares the performance observed on each WSI of the test set. The index 1 on the metrics refers to Model A_1 and the index 2 refers to Model A_2. Box plots were generated to represent visually in a single chart how the data points regarding a specific metric are spread. Those box plots are shown on Figure 7.1, and the immediate observation is that the data distribution is similar for each metric.

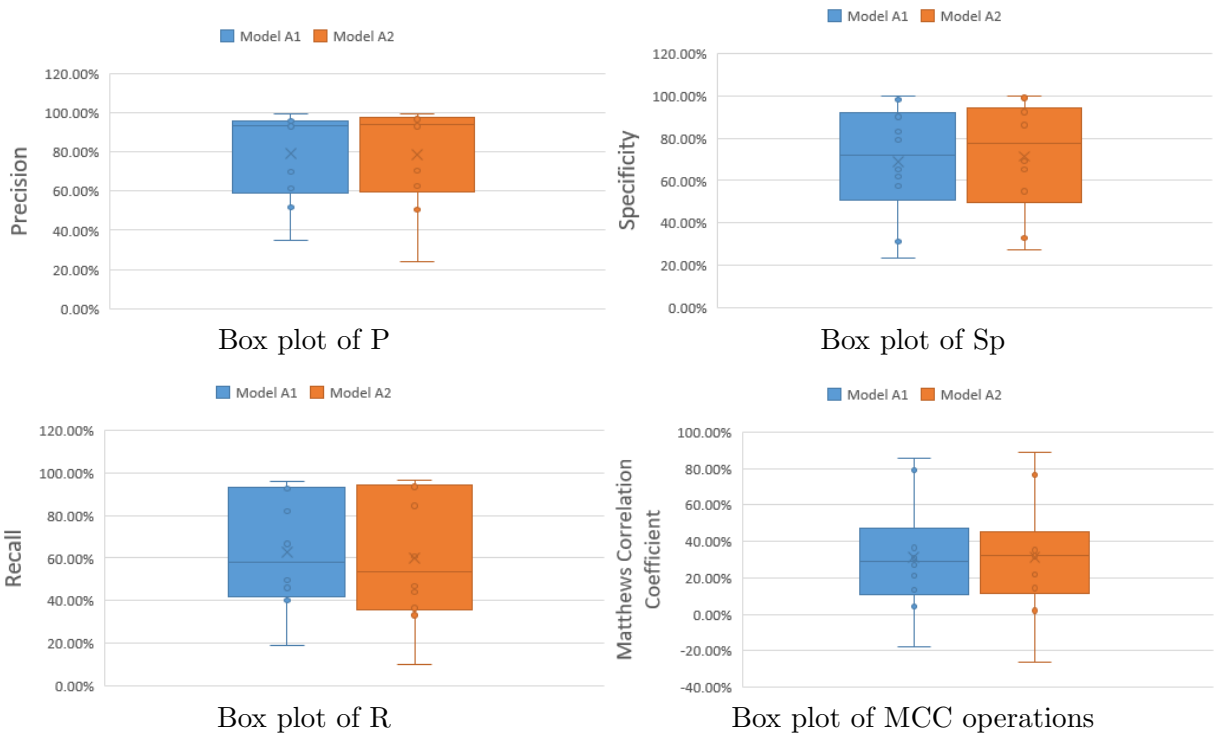


Figure 7.1: Box plots of the precision, specificity, recall and MCC of Model A_1 and A_2. The box plots show a high degree of similarity between the distributions

A statistical test, the Wilcoxon Signed-Ranks test, was conducted to assess whether the results are statistically different. This was done on the following metrics: the precision (P) , the specificity (Sp) , the recall (R) , and the matthews correlation coefficient (MCC). This non-parametric test uses data from matched pairs (in this case,

the matched pairs are the performance of Model A_1 and A_2 according to a metric, for each WSI of the test set) and is used to test the hypothesis of a zero-median difference between 2 paired populations (null hypothesis) [35].

WSI name	$P_1(\%)$	$P_2(\%)$	$Sp_1(\%)$	$Sp_2(\%)$	$R_1(\%)$	$R_2(\%)$	MCC_1	MCC_2
19H15527-Ki67	61.28	62.62	23.35	27.18	94.91	95.44	0.2686	0.3189
19cu009680-Ki67	96.68	98.96	99.80	99.95	66.62	60.92	0.7898	0.7633
19h11173-Ki67	69.84	70.47	31.37	32.81	92.63	93.50	0.3142	0.3444
19h14023-Ki67	93.22	93.07	57.37	55.04	82.11	84.58	0.3092	0.3234
20CU003108-Ki67	34.94	23.99	65.23	69.05	19.02	9.95	-0.1775	-0.2598
20CU006014-Ki67	93.06	94.50	79.21	86.60	40.23	33.21	0.1330	0.1426
20CU017239-Ki67	93.89	94.74	83.02	86.15	45.98	44.02	0.2100	0.2203
20H01437-Ki67	51.82	50.62	61.85	65.37	42.37	36.66	0.0431	0.0212
20cu022767-Ki67	95.76	96.75	90.08	92.40	95.72	96.71	0.8578	0.8909
20cu034252-Ki67	99.23	99.53	98.11	98.92	49.66	46.73	0.3642	0.3516

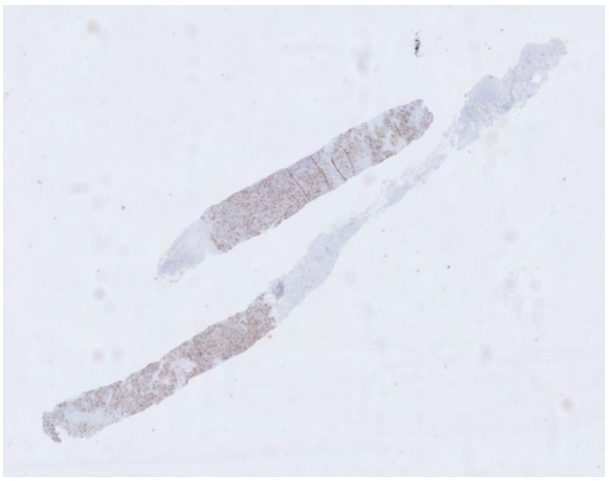
Table 7.4: Individual **test metrics** of the selected model (Model A) computed on the 10 WSIs of the test set, once without (1) and once with (2) overlapping tiles in the post-processing phase. The different metrics are : precision (P) , specificity (Sp) , recall (R) , and Matthews correlation coefficient (MCC)

The details of the computation of the Wilcoxon Signed-Ranks test for each metric is given in the Appendix C. It results from this test that there is no significant difference between the Model A_1 and A_2 with respect to the metrics P, R, and MCC (the null hypothesis is not rejected). However, there is a small significant difference observed between Model A_1 and A_2 with respect to Sp, the specificity metric (the null hypothesis can be rejected with $p = 0.02202$). The effect of using overlapping tiles as post-processing for the Model A is thus to slightly improve the specificity, i.e. it helps the model to better detect the negative class. However, this is not enough to discard one model from the other.

The standard deviation of the test results for each metric was already shown in Table 7.3 but it is even more striking when looking at the individual results in Table 7.4. The Model A shows great or mild performances on some WSIs from the test set, and really bad performances on others. The remainder of this section aims at assessing qualitatively the prediction results by showing the prediction masks and probability maps of key WSIs of the test set.

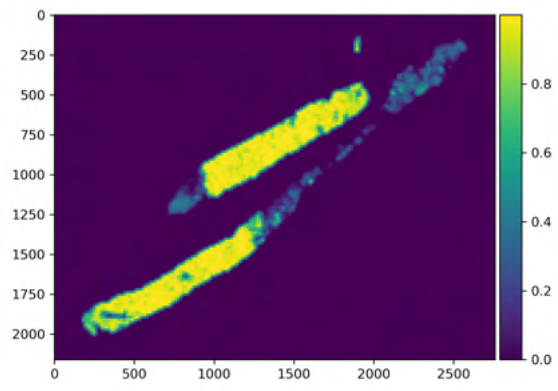
First, the model performed especially well on the following test images: 20cu022767-Ki67 (with $MCC_1 = 0.8578$ and $MCC_2 = 0.8909$) and 19cu009680-Ki67 (with $MCC_1 = 0.7898$ and $MCC_2 = 0.7633$). For the first one, 20cu022767-Ki67, it is observed in Figure 7.2 that the model predicted really well the tumorous regions and shows a high degree of confidence (mostly more than 90%) for the predicted tumorous tissues on the predicted positive probability map, while the non-tumorous tissues are correctly predicted as such but without as much confidence as for the positive class.

On the second image, 19cu009680-Ki67, the model correctly distinguished the tumorous region from the non-tumorous region (cf. Figure 7.3). This time, the probability map shows that the negative class was predicted with a particularly high degree of confidence (confidence in the range of $0 - 5\%$ of being positive, i.e. $95 - 100\%$ confidence of being negative).
It can be observed by looking at the qualitative performance on both images that on 20cu022767-Ki67, the correctly tumorous region contained a high density of Ki67 positive nuclei (manifested as brown hue [36]). A first thought would be that the network used this feature (but others as well) to detect positive areas. However, it is worth noting that results on 19cu009680-Ki67 show that the network was also able to predict correctly the left tissue as being non-tumorous, despite having large zones of high density of Ki-67 positive nuclei. It seems that the model understood that the Ki-67 expression is not limited to tumor, as it was discussed in Chapter 5.



WSI to predict

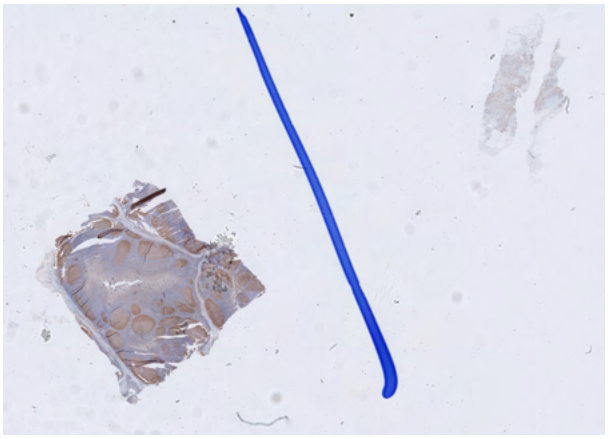
Annotation mask (ground truth)



Predicted positive probability map

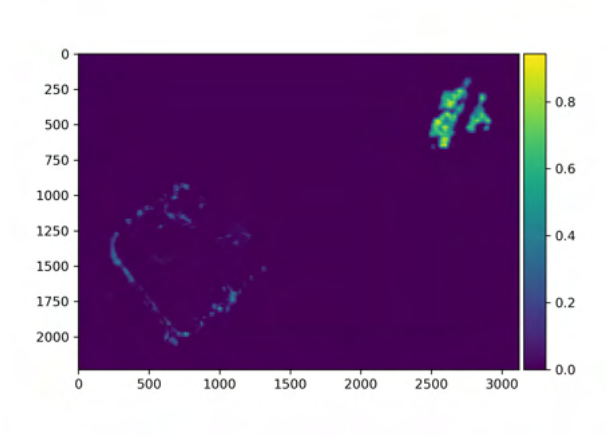
Predicted tumor mask

Figure 7.2: Prediction of the WSI 20cu022767-Ki67 from the test set by the Model A_2



WSI to predict

Annotation mask (ground truth)

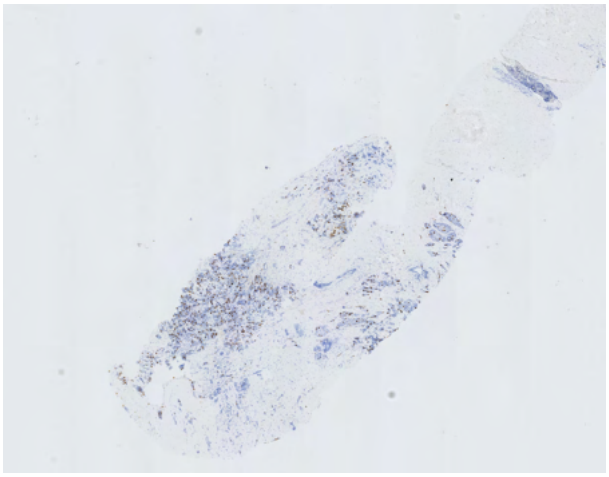


Predicted positive probability map

Predicted tumor mask

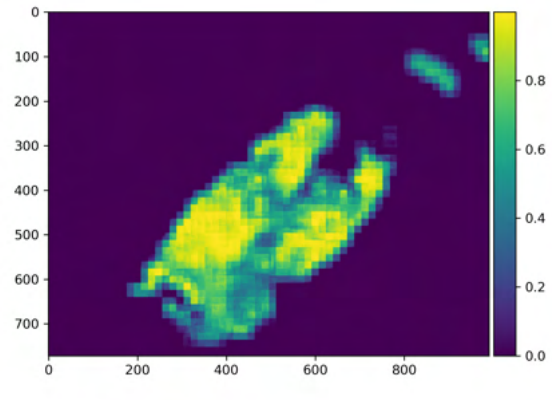
Figure 7.3: Prediction of the WSI 19cu009680-Ki67 from the test set by the Model A_2

On other predictions, however, the network shows mitigated results, as in Figure 7.4 for 19h11173-Ki67. The predicted tumor mask go slightly beyond the tumor, impacting the segmentation metrics. However, as can be observed on the predicted positive probability map, there is a noteworthy higher confidence of the network (around 90%) in the 3 annotated tumor regions. Hence, it seems that playing with the detection threshold could prove itself to be useful.
There are however other WSIs where the model had poor metrics, especially for 20CU003108-Ki67, as seen in Figure 7.5. There, not only are the tumorous regions nearly not detected, but other areas annotated as non-tumorous are falsely predicted positive. The wrongly predicted regions are however predicted without a high confidence of the network, as seen on the predicted positive probability map.



WSI to predict

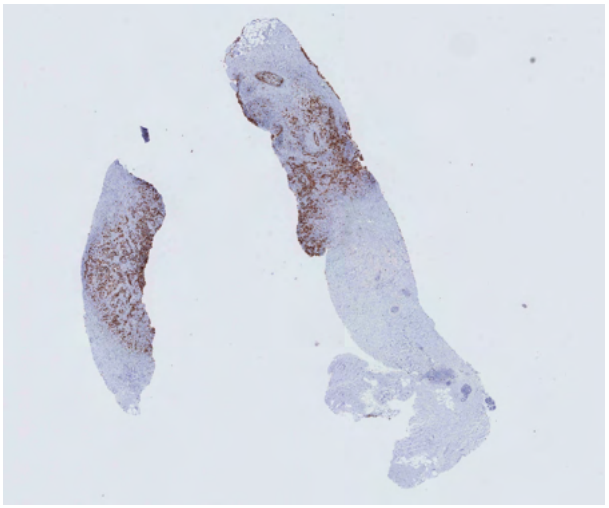
Annotation mask (ground truth)



Predicted positive probability map

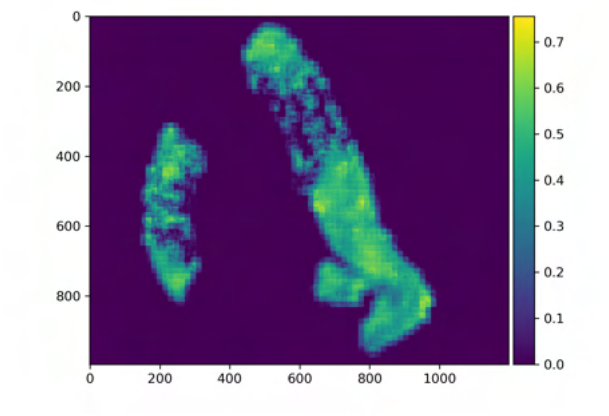
Predicted tumor mask

Figure 7.4: Prediction of the WSI 19h11173-Ki67 from the test set by the Model A_2



WSI to predict

Annotation mask (ground truth)



Predicted positive probability map

Predicted tumor mask

Figure 7.5: Prediction of the WSI 20CU003108-Ki67 from the test set by the Model A_2

The presented predictions, as well as the prediction of the remaining WSIs of the test set are given in Appendix D.

7.2 Discussion

Throughout Chapter 6, the following questions were raised: is it better to work at a lower magnification/resolution (magnification of x5 instead of x10) ? In addition to the resolution, how much context do the tiles need to encompass to carry out the segmentation task of this thesis: small, medium, or large ? What is the influence on the training of using overlapping tiles in the sliding-window approach? Is it possible to improve the segmentation results by using overlapping tiles in the inference process? Is this fourfold increase in inference time worth it?

From the experiences that were carried out, it turns out that using overlapping tiles in the sliding-window approach leads to better performances on the validation set for the medium and large contexts. It was also shown that tiles encompassing a medium context led to the best performances on the validation set, and that working at a x5 magnification was preferable with the data used in this thesis. Finally, it was shown that using overlapping tiles in the inference process led to no statistically different results, thus the fourfold increase in inference time is useless in this case.
The best model achieved in this work, Model A_1, led to the following performances on the test set : $accuracy = 65.30 \pm 19.46\%$, $precision = 78.97 \pm 22.86\%$, $specificity = 68.94 \pm 26.34\%$, $recall = 62.93 \pm 27.27\%$, $F1 - score = 67.14 \pm 20.95\%$, $IoU = 53.75 \pm 23.01\%$, and $MCC = 0.3112 \pm 0.3136$. Compared to previous works, the *means* of those metrics are similar to the ones obtained in [28] but are unsatisfactory compared to other works [8] [12] [27] [24] (cf. Table 4.1). However, the metrics of those other works should be only looked upon as indicators that give a hint on what is achievable, as each work was carried out on a different data set so the metrics can not be directly compared. Also, the high variance observed on the test metrics in this work make it even more difficult/less relevant to compare this work's model with others as the predictions made by the trained model of this work are unstable, with some WSIs really well predicted and others completely not.

Plausible explanations for the lack of performance of this model were identified. The first and main one is the fact that the data set used turned out to contain non-annotated tissue parts, as explained in Section 5.2 and Appendix A. While corrections of this problem could easily be handled on the test set by discarding manually the prediction results of the non-annotated regions, the annotation lack present in 6 out of the 25 training WSIs most probably altered the learning process of the model as, for the non-annotated regions, the network tried to learn tumorous features when trying to learn to to recognize non-tumorous tissues. Furthermore, the validation of the model was also impacted by similar annotation problems in the validation set. In addition to impacting the metrics, it could also have had the effect of stopping the learning earlier than necessary due to the early stopping used. However, it might as well have prolonged it. This could explain in part the high variance observed in the predictions of the WSIs of the test set. It is worth noting, however, that despite the problems in the data set, the model still seems to have derived useful features for its segmentation task, showing a certain robustness of the U-Net architecture used in this work.
The second explanation could be in the lack of annotated samples. Indeed, a substantial amount of tiles can be generated from the original WSIs thanks to the sliding-window approach and data augmentation, but in the end only 40 patient samples (WSIs) were used to carry out this work. It could be that there is not enough diversity in the tissues encountered in the training so that the model generalizes well on previously unseen data. It might explain the good performance of the model on some of the testing samples with confidence/high predicted positive probabilities ($p_+ > 90\%$) of belonging to the tumor class (as it is the case, for example, in the Figure 7.2), i.e. it may be that these tissues are similar to those previously seen by the network during the training. Conversely, it may be that testing samples that were moderately or badly predicted by the model had features that were not present in the training samples and thus not learned by the

network. However, this must be said with high caution and it is only given as a hypothesis as it is not safe to deduce interpretations from such an unstable network that was trained in part with annotations problems, and only from that few testing samples.

One could question the validity of using a segmentation network instead of a classification network for this particular segmentation task when using a sliding-window approach. Indeed, the resulting segmentation model in this work tends to predict most of the time the near majority of the tile pixels as being from only one class or the other, leading to a "tile effect" on the segmentation masks (straight borders on the predicted tumor region borders). This is coherent with the fact that the model was trained on 100% positive and 100% negative tiles, therefore the model was not explicitly trained to segment tumor borders at the tile level. Also, in the end, the information targeted is the tumorous zones (which are continuous regions) so the classification of tiles that are of a size small enough to avoid any inconvenient tile effect at the predicted tumor borders would be sufficient to get overall detection of the tumor regions.

One of the initial goal of this work was that the segmentation of the tumors of the network should be robust to changes in immunohistochemical labeling so that it can be applied to other markers of interest. This was not investigated in this work as the network was not even sufficiently performant on the Ki67 IHC labelling.

Chapter 8

Conclusion and Future Works

This work attempted to use the power of deep learning to segment automatically the tumor areas in IHC Ki67 stained histological sections of breasts. A data set containing 40 WSIs annotated by an expert pathologist was provided to carry out this objective. This data set was split into a training set (25), validation set (5) and test set (10) according to a statistical rule based on the proportion of tumor on the tissues of the image. A major challenge of this work was the dimensionality of the images provided and the low amount of training samples available. Indeed, DL models need extensive amount of training data to derive useful features, and current DL architectures and hardware are unable to deal with such images. In this work, following the previous works [8] [27] [24] [28] [12], a sliding-window approach was used to tackle both problems. The WSIs were tiled in tiles of equal size that were small enough to meet the memory and GPU requirements of the computer provided.
The advantage of this approach was the creation of numerous training samples for the DL model training. Extensive data augmentation techniques (vertical/horizontal flipping, random rotation, random intensity correction and random noise introduction) were also used to counter the scarcity of data. Another data augmentation technique was the use of a 50% overlap in the sliding-window approach. However, no matter the number of tiles generated, it is worth noting that the total data information remains the same as the training samples still only come from 40 patient samples.
The downside of this tiling approach is the loss of contextual information. Therefore, experiences were led by training different models with tiles of different size and resolution encompassing a total of 3 different contexts in order to determine the right amount of context needed for the model to derive useful features for the segmentation task.

The DL network architecture used in this work is the classical U-Net - a CNN used for segmentation tasks - with minor modifications. From the different experiences that were carried out, it resulted based on the validation metrics that:

- In addition to the classical data augmentation techniques, the use of overlapping tiles in the sliding-window approach as a data augmentation technique significantly improved the model performance with respect to the validation metrics

- With the data used in this work, working at a x5 magnification yields better performances than working at a x10 magnification
- The context yielding to the best performances on the validation metrics is a tile size of 128x128 pixels (at x5 magnification). The model trained with input images (tiles) of this context, and generated thanks to a 50% overlap sliding-window approach was retained.

To assess the true performance of the retained model, the latter was evaluated on previously unseen data: the 10 WSIs kept aside in the test set. In this work, before computing the test metrics, two post-processing techniques were used and compared to reconstruct the global prediction mask from the local (tiles) predictions: the first one consisted in simply stitching the individual tile positive probability maps together and applying a 0.5 threshold on it to get the predicted mask; the second one consisted in stitching overlapping tile positive probability maps together, and averaging the probability of belonging to the tumor class for each pixel predicted multiple times. The results on the test set showed a high variance across the different WSIs predictions for each observed metric, and this for both post-processing techniques. Statistical tests were conducted on several key metrics to assess whether the results of both post-processing techniques were statistically different. It results that they were not, apart for the specificity which showed a minor increase with overlapping tiles in the inference process. The fourfold increase in inference time with this overlapping post-processing technique is therefore not worth it. The final infererence pipeline consits in: loading the WSI and its corresponding annotation at x5 magnification, pre-processing the WSI to only work on the tissue regions of the image; on the tissue regions, a non-overlapping sliding window is used to tile the tissue in 128x128 tiles ; each pixel of each tile is predicted thanks to a trained U-Net segmentation model and the probability maps of the tiles are stitched together ; finally, a 0.5 threshold is applied on the reconstructed probability map to extract the predicted segmentation mask.

This model led to the following performances on the test set : $accuracy = 65.30 \pm 19.46\%$, $precision = 78.97 \pm 22.86\%$, $specificity = 68.94 \pm 26.34\%$, $recall = 62.93 \pm 27.27\%$, $F1-score = 67.14 \pm 20.95\%$, $Intersection\,over\,Union = 53.75 \pm 23.01\%$, and $MCC = 0.3112 \pm 0.3136$. As discussed previously in Section 7.2, the final model did not reach a satisfactory level of performances on the test set because of low metric means and big standard deviations, and are unsatisfactory compared to previous works. One big factor was the discovery - too late in the process - that the data set used included annotation problems with the presence of tumorous tissues that were not labelled as such. Another factor that was given, as an hypothesis, is that the lack of training samples impeded the network to generalize well, despite the data augmentation techniques that were applied.

Future works should first and foremost investigate the performance of the model trained with the same training pipeline but on a correction of the data set provided, i.e. the data set free of the annotation problems explained in Section 5.2 and Appendix A. It is obviously expected that the resulting model would outperform the current one. Comparing the performance of the two models will assess the impact that the annotation lack

had on the model performances. Also, complementing the data set with new WSIs should improve the model performance and its generalization.
Then, different thresholds than the 0.5 used in this work on the probability map to generate the predicted tumor mask could be investigated to assess the impact it has on the model performance. Once the optimal threshold is found based on the performance on the validation data, the test metrics should be recomputed using this new threshold.
The overlapping tiles were used as a data augmentation technique in this work, but were treated individually in the training process. Future works on this data set could try to exploit the information located in the overlapping zones of the tiles during the training process to try to improve the training's output.
The influence of other parameters could be assessed such as the learning rate, the optimizer parameters, the number of filters in the U-net (may be the features needed to segment tumor tissues do not need a high number of filters). The potential benefits of using of transfer learning to initialize the weights with those of a pre-trained network could also be assessed.
One drawback of the method used in this thesis to partition the data into training, validation and test sets was the fact that it led in this case to a proportion of generated positive tiles significantly lower in the validation set than in the training set. Other ways of partitioning the data set into training, validation, and test set could be investigated, for example by asking a pathologist to partition them qualitatively with representing samples in each set according to his/her expertise.
One of the initial goals of this work was that the segmentation of the tumors of the network should be robust to changes in IHC labeling so that it can be applied to other markers of interest. This was not investigated in this work as the network was not even sufficiently performing on the Ki67 IHC labelling. However, future works with decent model performances should assess if the resulting U-Net model is robust to those changes in IHC labelling. Also, once the tumor regions are correctly segmented, one could further extend the analysis to detect "hot-spots" of proliferation within the tumors, i.e. areas of high density ok Ki67-positive cell nuclei within the tumor area.
Hardware constraints restricted this work's analysis to x5 and x10 magnification, but future works on this data set could explore the impact of working at higher magnifications, as it was done in all previous works [8] [27] [24] [28] [12].

Appendix A

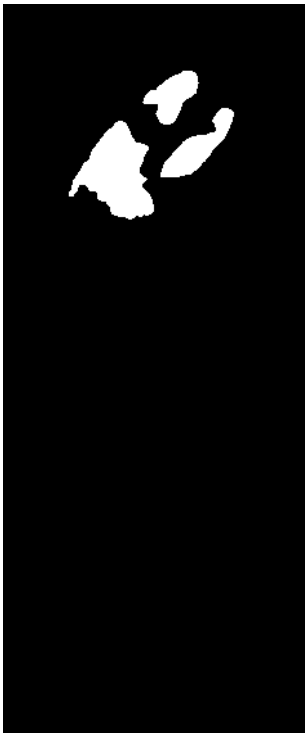
Annotation lack in the data set provided

The goal of this appendix is to briefly explain the difficulties encountered during this thesis that were - in fact - due to an annotation lack in the data set.

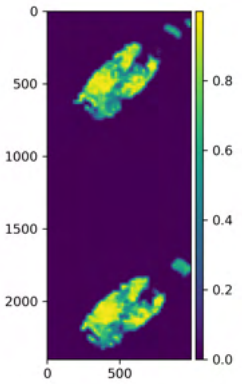
There were a total of 2 data sets that were given to the student.
The first one contained 153 Ki67 WSIs where *representative* tumor regions were annotated (the rest of the tissues were left unlabelled and thus de facto considered non-tumorous). In the coding of the training pipeline, the student struggled a lot because the model showed no sign of learning on the data set provided. Facing this, it was hypothesized by the student that there was some error in the code that he had written and a consequent amount of time was spent rewriting the already existing functions. As the models still showed no sign of learning, it was suspected that the original data set contained too few annotations and that it was not suited to train a model on this data.
Early in July, a new data set containing 40 Ki67 WSIs with tumor annotations was provided to the student. As the data set given from the pathologist to a member of the university and then from that member to the student, a problem arose in the transmission of information. Indeed, it was communicated to the student that this new data set had been more carefully annotated by the pathologist and that *all* the tumor regions had been identified. It was omitted to the student, however, that when serial cuts from a same tissue but at different height were displayed on the same WSI, only one of those tissues had been annotated as tumorous although the other - that was left unlabelled - was also tumorous. Upon receiving this new data set, the student observed that on some WSIs some tissue regions had a similar shape and texture (while still showing different cells and Ki67 expression), but the student thought that the pathologist had a specific reason to consider one tissue as being tumorous and the other non-tumorous. The concerned unlabelled tissues were therefore not discarded by the student and since the hypothesis in this work was that any unlabelled tissue was to be considered as being part of the negative class (non-tumor class), the written code blindly trusted the annotation mask that was fed to provide the network with samples of the positive and the negative class. Although this annotation lack concerned 11 out of 40 WSIs, this time it did not prevent

the model from showing signs of learning. The student was thus comforted in the idea that the new data set was valid as such, since there were - at last - signs of learning from the network. However, the performances on the validation sets were still moderate on the validation set and insufficient on the test set. The student spent a lot of time trying to tweak the parameters with the hope (without success) that it would unblock this situation and create a jump in the performances. It was only in August that the student had a light-bulb moment when looking more closely at the probability maps that were predicted from the test WSIs (and not at the metrics). Indeed, when comparing the input image, annotation mask and probability maps together, the student realized that the predictions of the slides that contained tissues of similar shape and texture (the ones explained above, with the annotation problem) had correctly predicted the annotated tumor, but had also predicted the other tissue as being tumorous too (impacting thus severely the segmentation metrics). This is illustrated in Figure A.1. Before, the student thought it was only an error of the network but as the tweaking of parameters still led to similar performances, the student suddenly realized that both tissues had a texture that was too similar and that if one was considered tumorous, so should the other one. Since the student is not trained to distinguish between tumorous and non-tumorous regions with its eyes (it requires an expert knowledge), the student brought this finding to its promoter and supervisors, who confirmed the student's doubts. This problem was then identified in 11 out of the 40 WSIs provided (6 in the training set, 1 in the validation set, and 4 in the test set). It was too late, however, for the student to clean the data set by discarding the concerned regions and then retrain the model on this new data set. Instead, only the metrics of the test set were corrected by discarding the concerned regions.

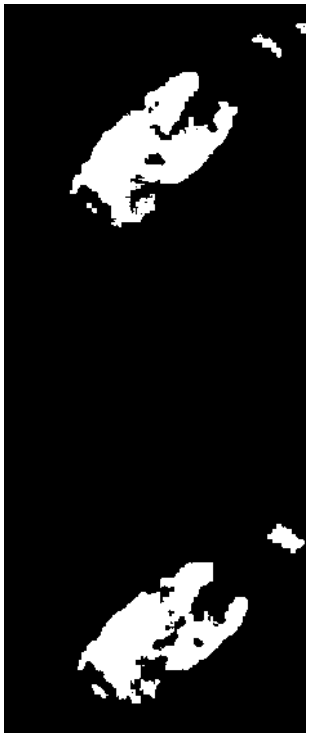
WSI to predict



Annotation mask (ground truth)



Predicted positive probability map



Predicted tumor mask

Figure A.1: Example of annotation lack. The WSI contains 2 serial cuts of the same tissue. The tissue from above was annotated positive and the tissue from below was left unannotated even though it is also tumorous. The network identified tumorous zones in both tissues

Appendix B

WSI prediction example using (non) overlapping tiles in the inference process

The goal of this appendix is to illustrate the difference in the results of the 2 post-processing techniques described in this work in Section 6.4: one with and another without the use of overlapping tiles in the inference process. The Figure B.1 shows an example of WSI to predict and Figure B.2 shows the prediction results using the 2 post-processing techniques.

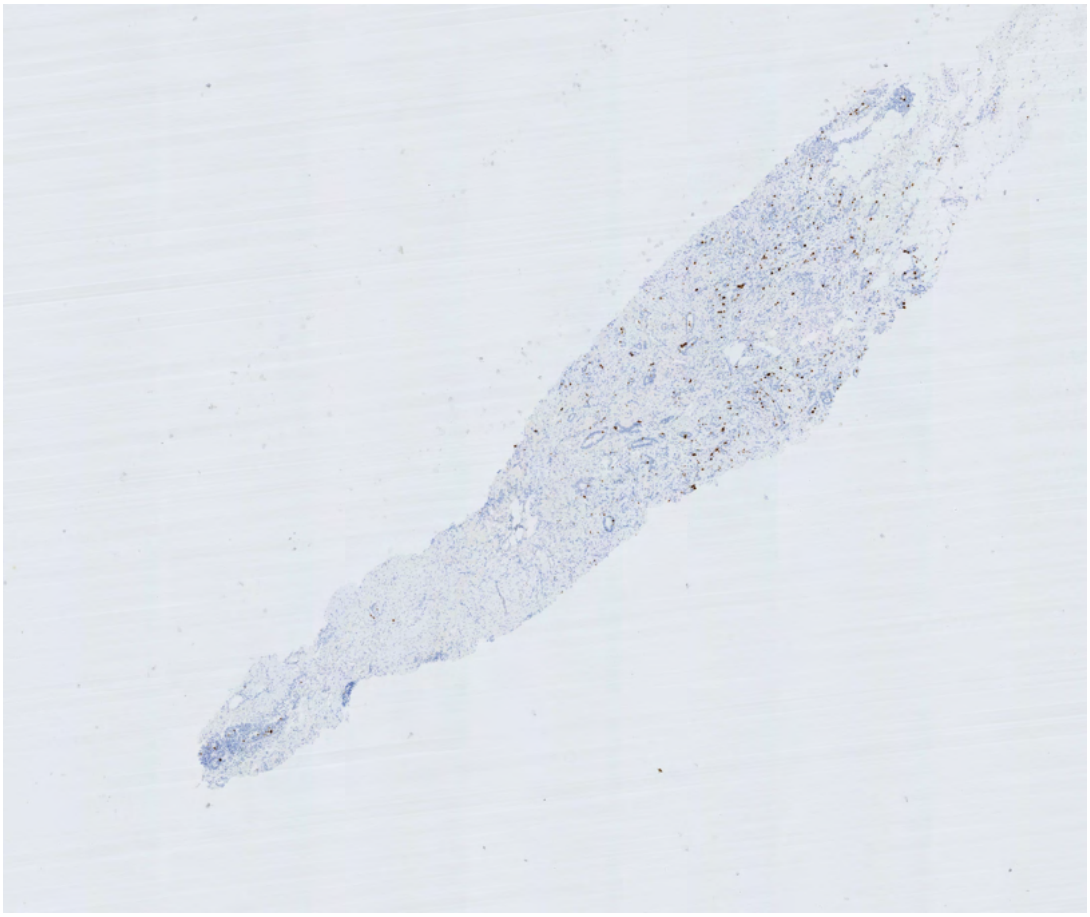
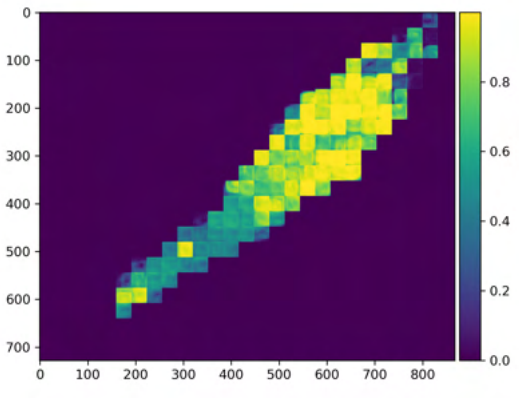
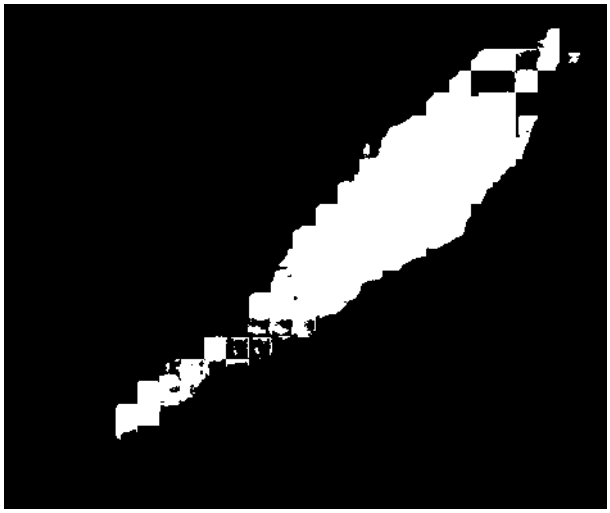


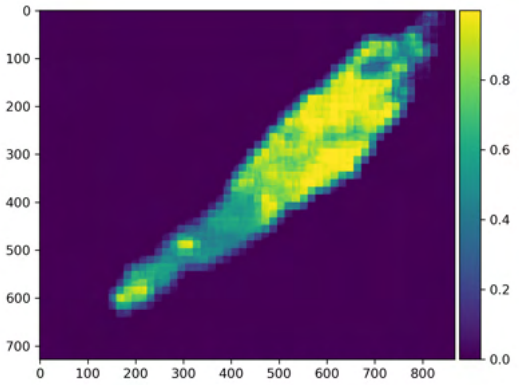
Figure B.1: WSI to predict



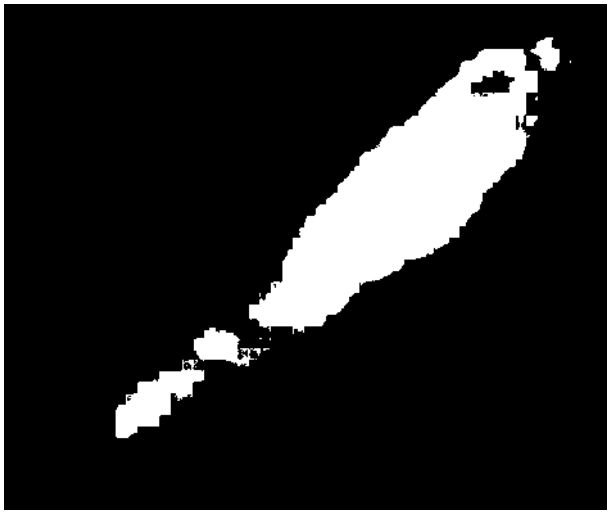
Stitching of non-overlapping tile predictions (positive probability map)



Stitching of non-overlapping tile predictions (predicted mask)



Stitching of overlapping tile predictions (positive probability map



Stitching of overlapping tile predictions (predicted mask

Figure B.2: Comparison of the prediction results on a same WSI using 2 different post-processing techniques: one using non-overlapping tiles in the inference process, and another using overlapping tiles in the inference process

Appendix C

Wilcoxon Signed-Ranks Test on the results of the models on the test set

This appendix shows the results of the Wilcoxon matched-pairs signed-ranks test done on the MCC, P, R, and SP metrics. The computations were done with an online calculator that can be found at the following address: `https://www.socscistatistics.com/tests/signedranks/`):

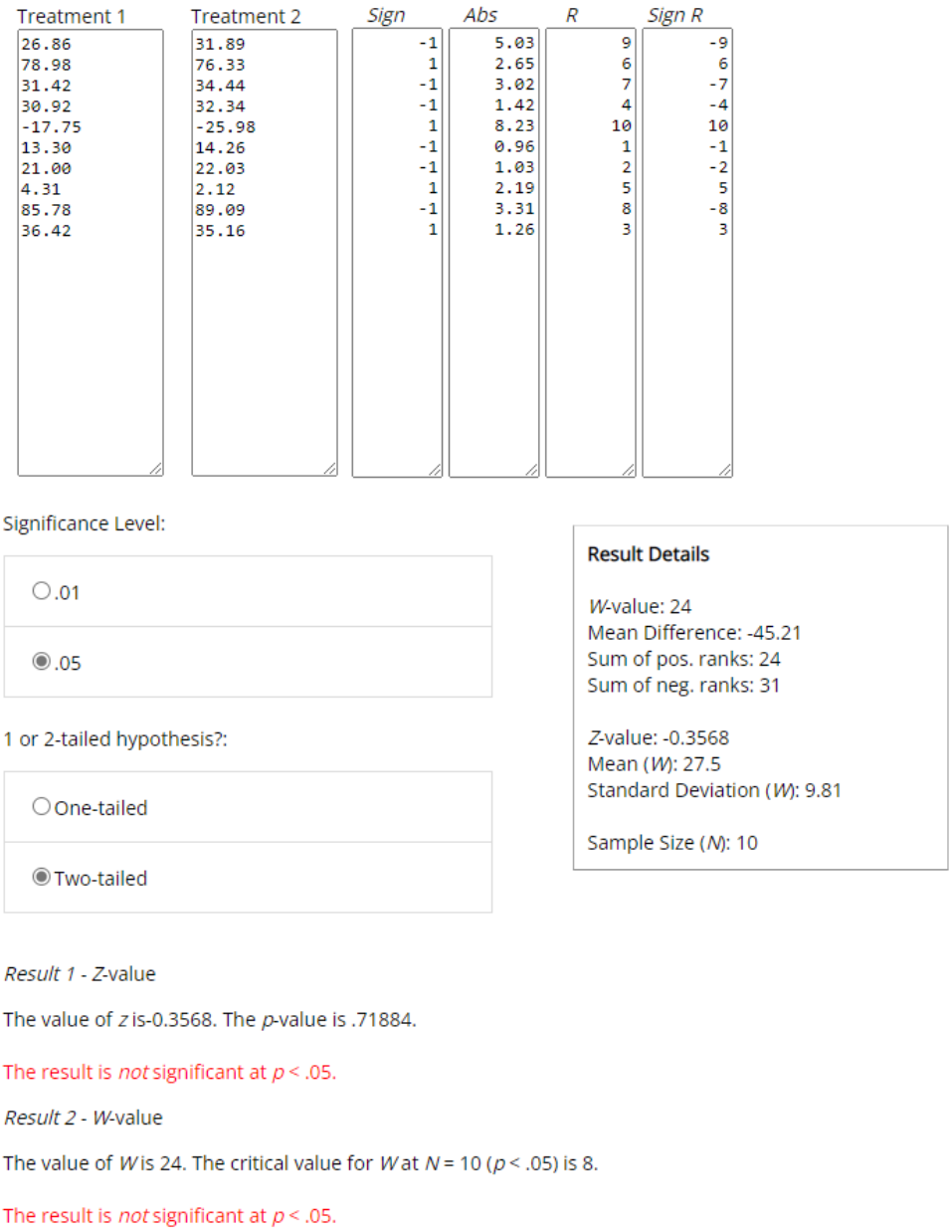


Figure C.1: Wilcoxon Signed-Rank Test computation of Model A_1 and A_2 with respect to the MCC metric. In this case, the null hypothesis can not be rejected

Treatment 1	Treatment 2	*Sign*	*Abs*	*R*	*Sign R*
61.28	62.62	-1	1.34	7	-7
96.68	98.96	-1	2.28	9	-9
69.84	70.47	-1	0.63	3	-3
93.22	93.07	1	0.15	1	1
34.94	23.99	1	10.95	10	10
93.06	94.50	-1	1.44	8	-8
93.89	94.74	-1	0.85	4	-4
51.82	50.62	1	1.2	6	6
95.76	96.75	-1	0.99	5	-5
99.23	99.53	-1	0.3	2	-2

Significance Level:

○ .01

◉ .05

1 or 2-tailed hypothesis?:

○ One-tailed

◉ Two-tailed

Result Details

W-value: 17
Mean Difference: -19.99
Sum of pos. ranks: 17
Sum of neg. ranks: 38

Z-value: -1.0703
Mean (*W*): 27.5
Standard Deviation (*W*): 9.81

Sample Size (*N*): 10

Result 1 - *Z*-value

The value of *z* is-1.0703. The *p*-value is .28462.

The result is *not* significant at $p < .05$.

Result 2 - *W*-value

The value of *W* is 17. The critical value for *W* at $N = 10$ ($p < .05$) is 8.

The result is *not* significant at $p < .05$.

Figure C.2: Wilcoxon Signed-Rank Test computation of Model A_1 and A_2 with respect to the P metric. In this case, the null hypothesis can not be rejected

Treatment 1	Treatment 2	*Sign*	*Abs*	*R*	*Sign R*
94.91	95.44	-1	0.53	1	-1
66.62	60.92	1	5.7	7	7
92.63	93.50	-1	0.87	2	-2
82.11	84.58	-1	2.47	5	-5
19.02	9.95	1	9.07	10	10
40.23	33.21	1	7.02	9	9
45.98	44.02	1	1.96	4	4
42.37	36.66	1	5.71	8	8
95.72	96.71	-1	0.99	3	-3
49.66	46.73	1	2.93	6	6

Significance Level:

○ .01

◉ .05

1 or 2-tailed hypothesis?:

○ One-tailed

◉ Two-tailed

Result Details

W-value: 11
Mean Difference: 2
Sum of pos. ranks: 44
Sum of neg. ranks: 11

Z-value: -1.6818
Mean (*W*): 27.5
Standard Deviation (*W*): 9.81

Sample Size (*N*): 10

Result 1 - *Z*-value

The value of *z* is-1.6818. The *p*-value is .09296.

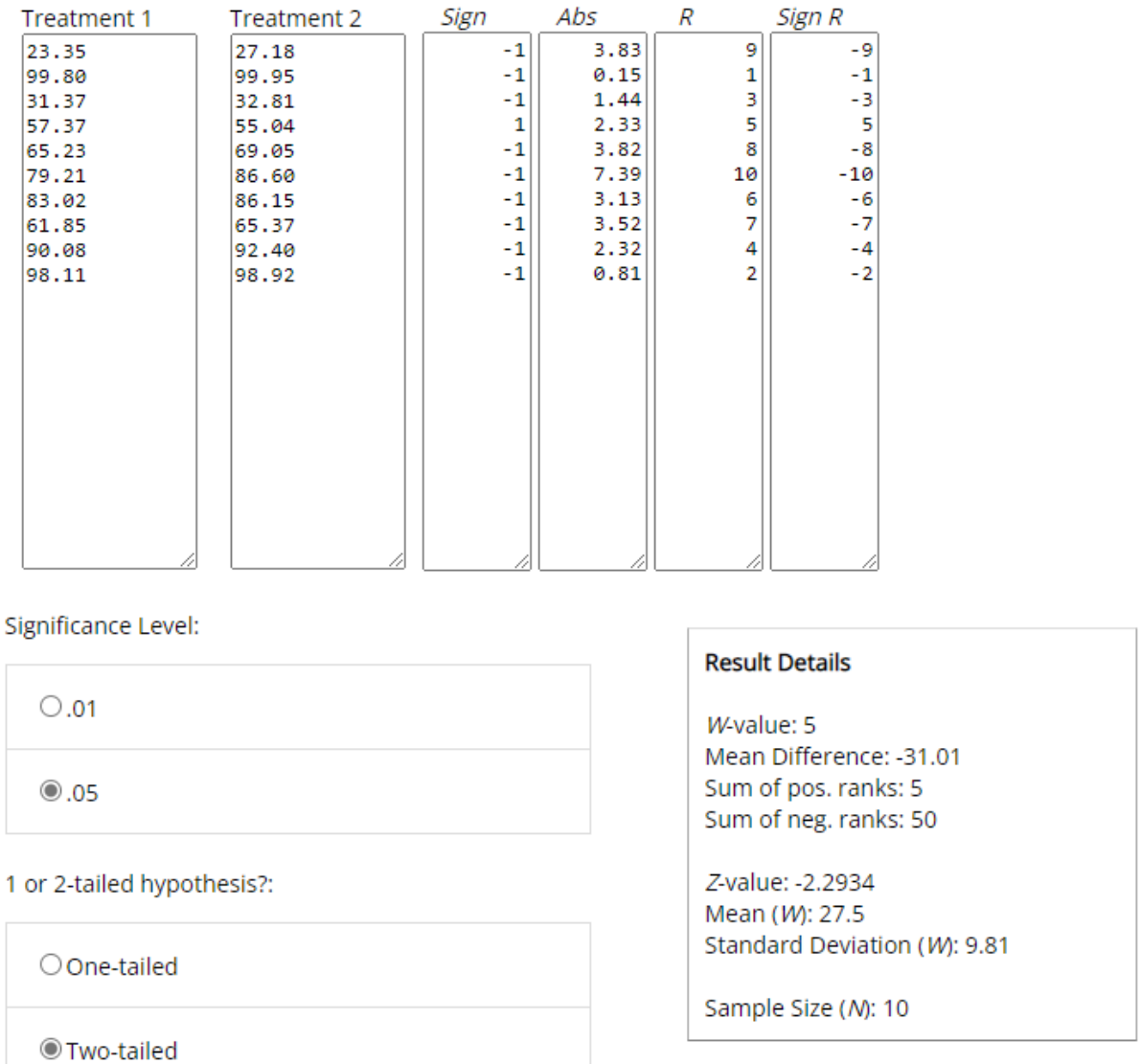
The result is *not* significant at $p < .05$.

Result 2 - *W*-value

The value of *W* is 11. The critical value for *W* at $N = 10$ ($p < .05$) is 8.

The result is *not* significant at $p < .05$.

Figure C.3: Wilcoxon Signed-Rank Test computation of Model A_1 and A_2 with respect to the R metric. In this case, the null hypothesis can not be rejected

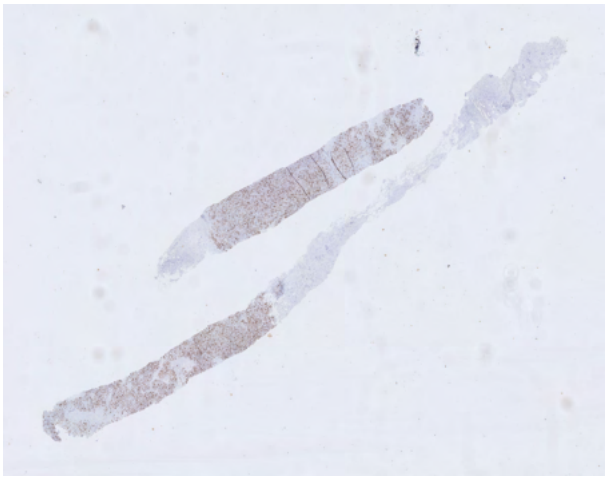


Treatment 1	Treatment 2	*Sign*	*Abs*	*R*	*Sign R*
23.35	27.18	-1	3.83	9	-9
99.80	99.95	-1	0.15	1	-1
31.37	32.81	-1	1.44	3	-3
57.37	55.04	1	2.33	5	5
65.23	69.05	-1	3.82	8	-8
79.21	86.60	-1	7.39	10	-10
83.02	86.15	-1	3.13	6	-6
61.85	65.37	-1	3.52	7	-7
90.08	92.40	-1	2.32	4	-4
98.11	98.92	-1	0.81	2	-2

Figure C.4: Wilcoxon Signed-Rank Test computation of Model A_1 and A_2 with respect to the Sp metric. In this case, the null hypothesis is rejected

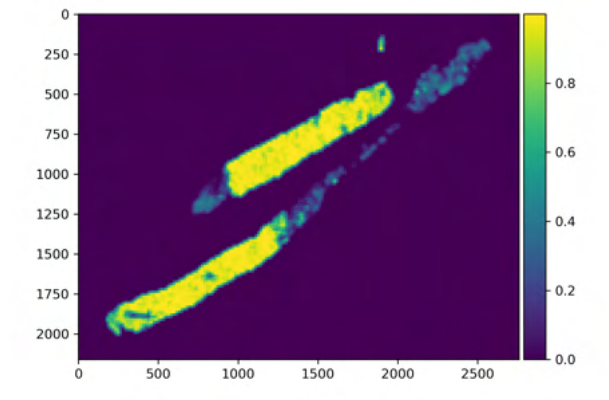
Appendix D

Predictions of the model on the test set



WSI to predict

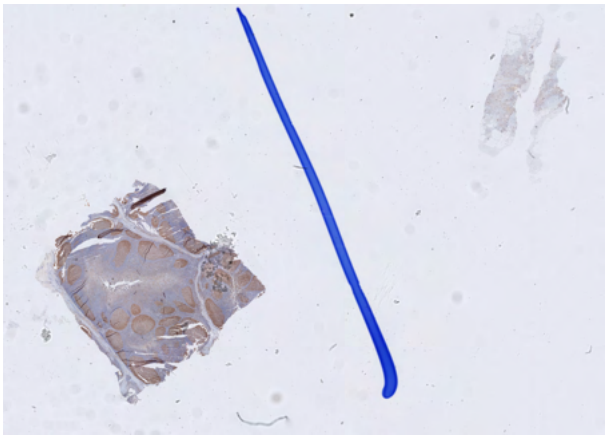
Annotation mask (ground truth)



Predicted positive probability map

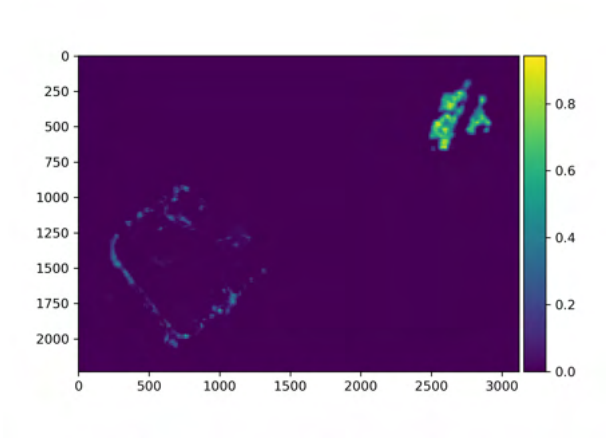
Predicted tumor mask

Figure D.1: Prediction of the WSI 20cu022767-Ki67 from the test set by the Model A_2

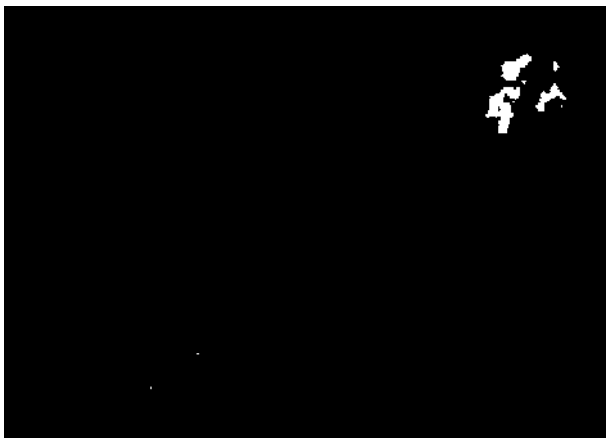


WSI to predict

Annotation mask (ground truth)

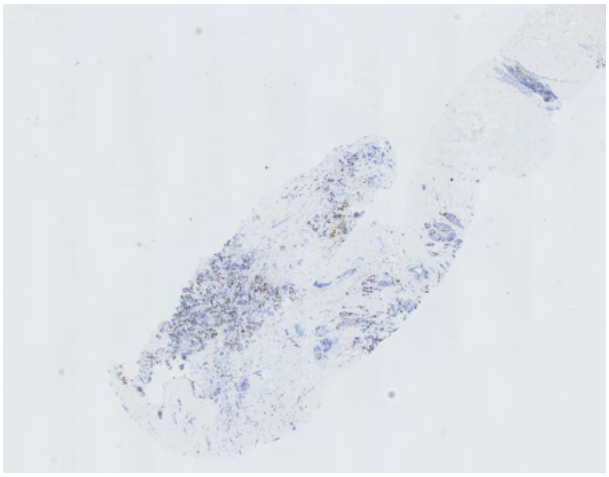


Predicted positive probability map



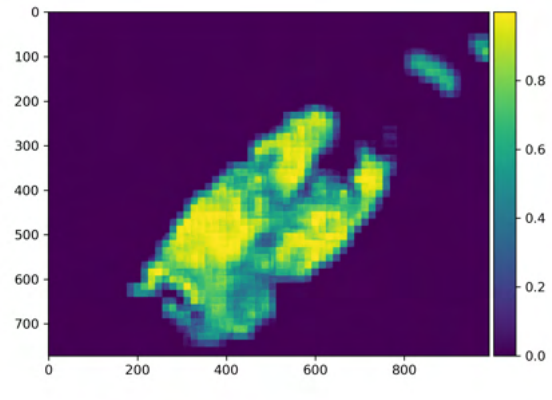
Predicted tumor mask

Figure D.2: Prediction of the WSI 19cu009680-Ki67 from the test set by the Model A_2



WSI to predict

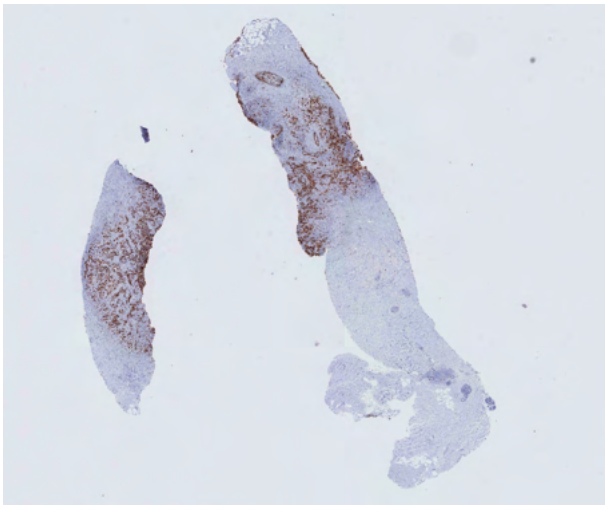
Annotation mask (ground truth)



Predicted positive probability map

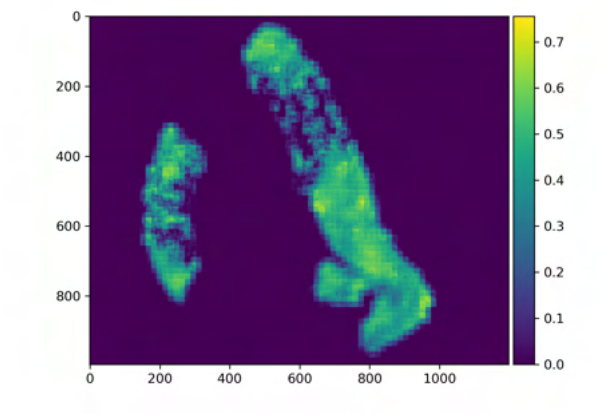
Predicted tumor mask

Figure D.3: Prediction of the WSI 19h11173-Ki67 from the test set by the Model A_2



WSI to predict

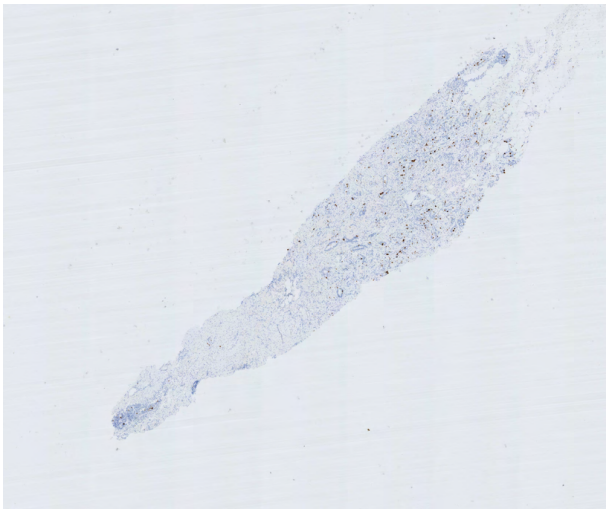
Annotation mask (ground truth)



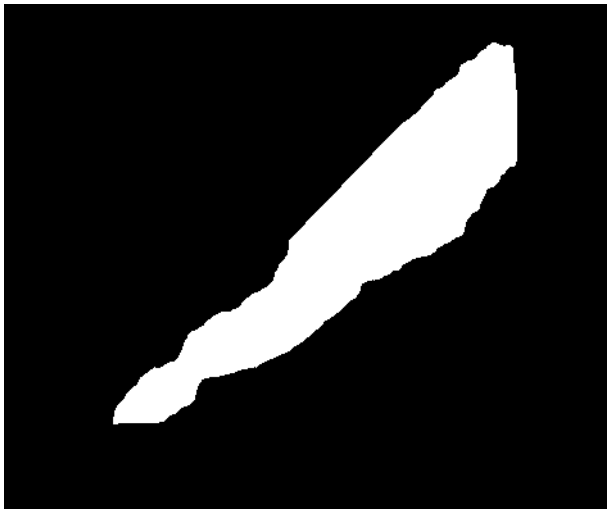
Predicted positive probability map

Predicted tumor mask

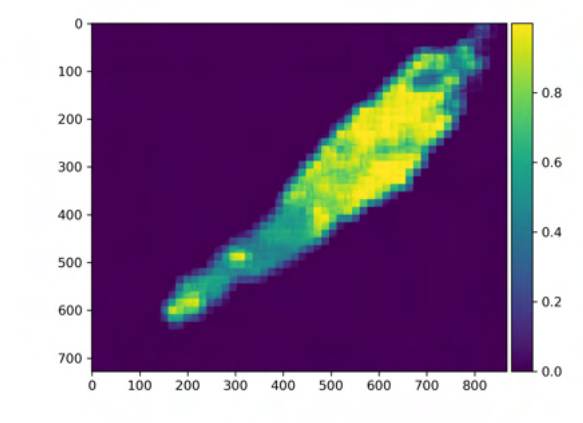
Figure D.4: Prediction of the WSI 20CU003108-Ki67 from the test set by the Model A_2



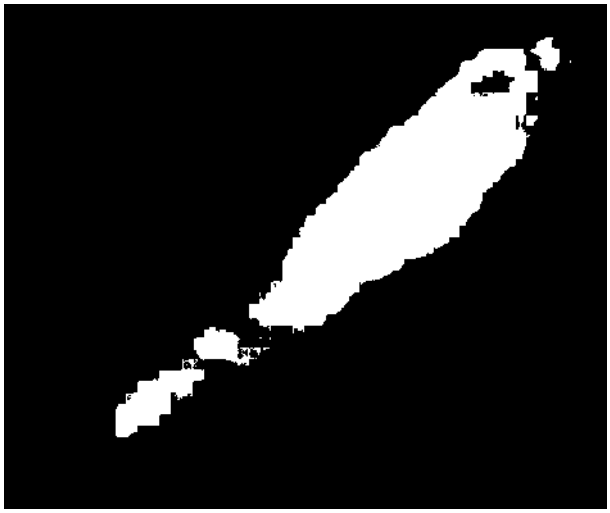
WSI to predict



Annotation mask (ground truth)

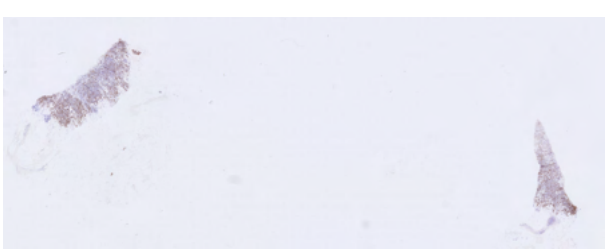


Predicted positive probability map



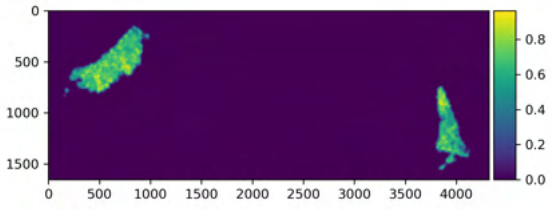
Predicted tumor mask

Figure D.5: Prediction of the WSI 19h14023-Ki67 from the test set by the Model A_2



WSI to predict

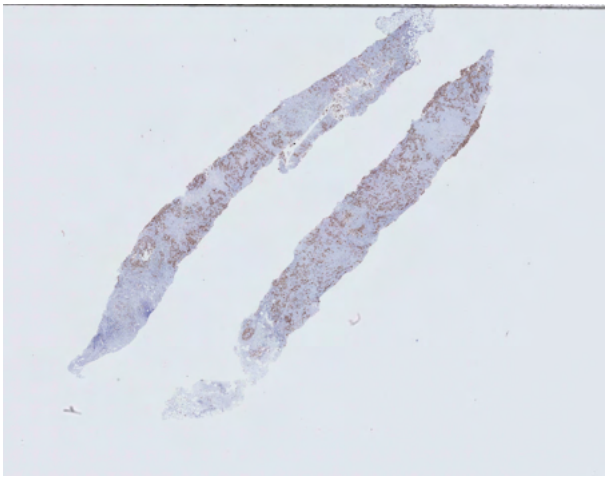
Annotation mask (ground truth)



Predicted positive probability map

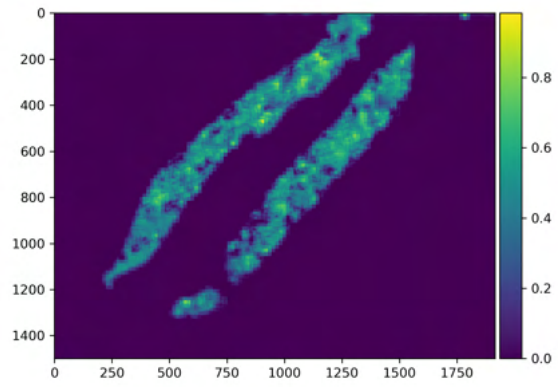
Predicted tumor mask

Figure D.6: Prediction of the WSI 19H15527-Ki67 from the test set by the Model A_2



WSI to predict

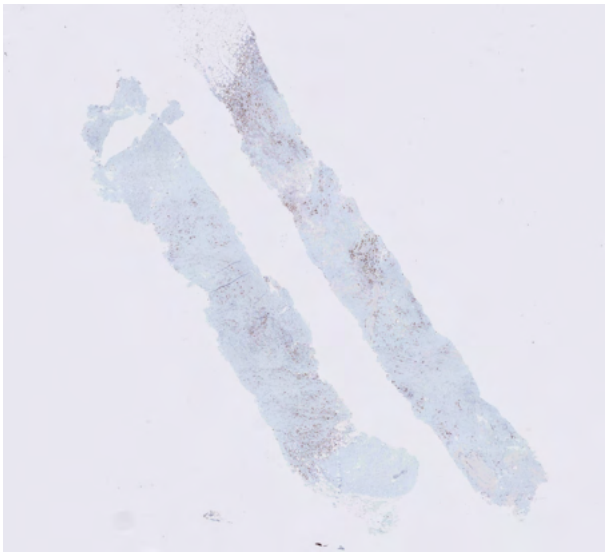
Annotation mask (ground truth)



Predicted positive probability map

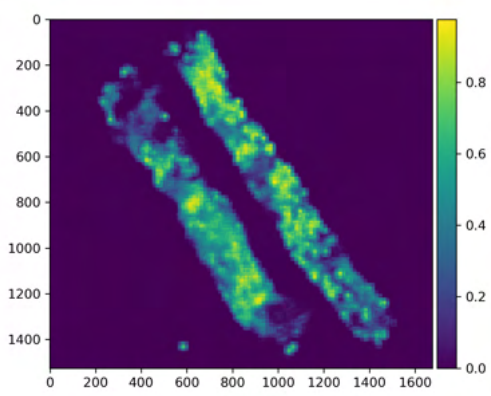
Predicted tumor mask

Figure D.7: Prediction of the WSI 20CU006014-Ki67 from the test set by the Model A_2

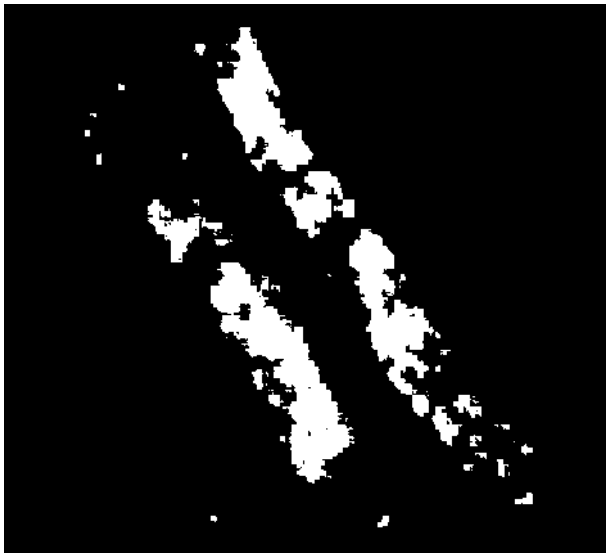


WSI to predict

Annotation mask (ground truth)

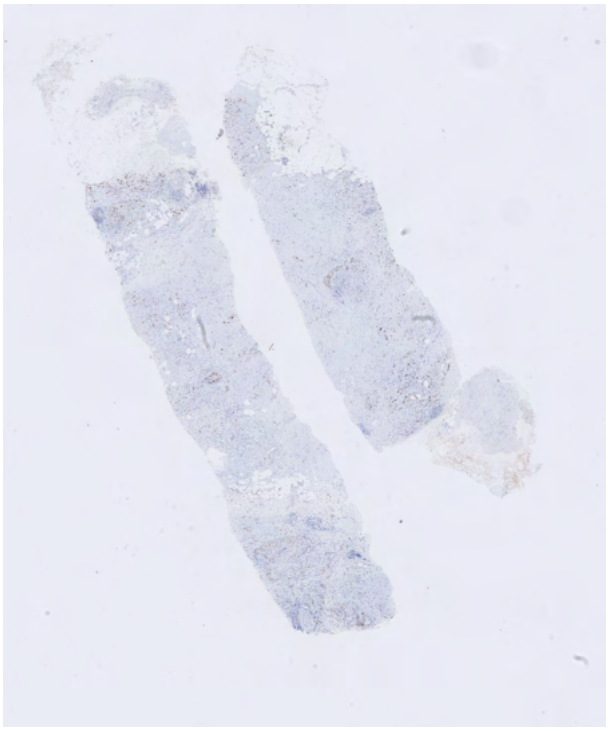


Predicted positive probability map



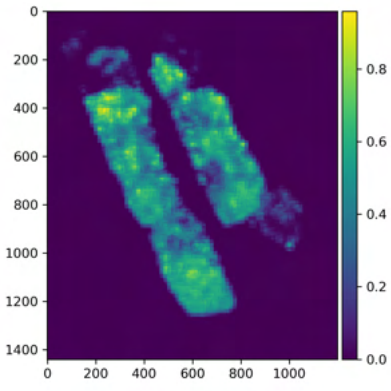
Predicted tumor mask

Figure D.8: Prediction of the WSI 20CU017239-Ki67 from the test set by the Model A_2



WSI to predict

Annotation mask (ground truth)



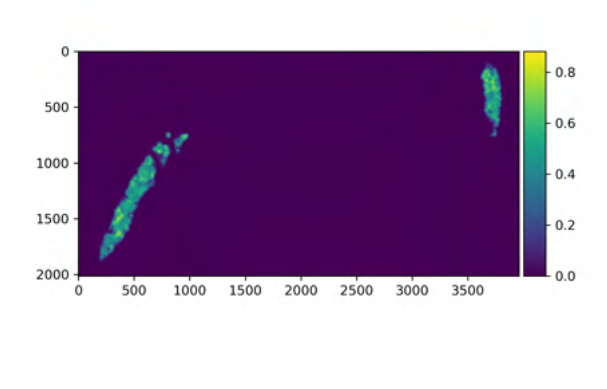
Predicted positive probability map

Predicted tumor mask

Figure D.9: Prediction of the WSI 20cu034252-Ki67 from the test set by the Model A_2

WSI to predict

Annotation mask (ground truth)



Predicted positive probability map

Predicted tumor mask

Figure D.10: Prediction of the WSI 20H01437-Ki67 from the test set by the Model A_2

Bibliography

[1] DeSantis C., Ma J., Bryan L., and Jemal A. Breast cancer statistics. *CA: a cancer journal for clinicians*, 64(1):52–62, 2013.

[2] World Health Organization Global Cancer Observatory, International Agency for Research on Cancer. *Globoscan 2020 (Breast)*, 2020. `https://gco.iarc.fr/today/data/factsheets/cancers/20-Breast-fact-sheet.pdf` [Accessed: 2021-09-28].

[3] Hawkes N. Cancer survival data emphasise importance of early diagnosis. *BMJ*, 364:52–62, 2019.

[4] Debelee T. G., Kebede S. R., Schwenker F., and Shewarega Z. M. Deep learning in selected cancers' image analysis — a survey. *Journal of Imaging*, 6(11):121, 2020.

[5] Weissleder R. and Nahrendorf M. Advancing biomedical imaging. *Proceedings of the National Academy of Sciences*, 112(47):14424–14428, 2015.

[6] Khened M., Kori A., Rajkumar H., Krishnamurthi G., and Srinivasan B. A generalized deep learning framework for whole-slide image segmentation and analysis. *Scientific Reports*, 11(1):1–14, 2021.

[7] Dimitriou N., Arandjelović O., and Caie P. D. Deep learning for whole slide image analysis: An overview. *Frontiers in Medicine*, 6:264, 2019.

[8] Joseph J., Roudier M. P., Narayanan P.L., Augulis R., Ros V.R., Pritchard A., Gerrard J., Laurinavicius A., Harrington E.A., Barrett J.C., and Howat W.J. Proliferation tumour marker network (ptm-net) for the identification of tumour region in ki67 stained breast cancer whole slide images. *Scientific Reports*, 9(1):1–12, 2019.

[9] Meijering E. A bird's-eye view of deep learning in bioimage analysis. *Computational and structural biotechnology journal*, 18:2312–2325, 2020.

[10] Elmore J. G., Longton G. M., Carney P. A., Geller B. M., Onega T., Tosteson A. N., and Weaver D. L. Diagnostic concordance among pathologists interpreting breast biopsy specimens. *JAMA*, 313(11):1122–1132, 2015.

[11] Kulkarni S., Seneviratne N., Baig M. S., and Khan A. H. A. Artificial intelligence in medicine: Where are we now? *Academic radiology*, 27(1):62–70, 2020.

[12] Zeiser F. A., da Costa C. A., de Oliveira Ramos G., Bohn H. C., Santos I., and Roehe A. V. Deepbatch: A hybrid deep learning model for interpretable diagnosis of breast cancer in whole-slide images. *Expert systems with applications*, 185:115586, 2021.

[13] Ronneberger O., Fischer P., and Brox T. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pages 234–241, 2015.

[14] Gehlot S. and Gupta A. Self-supervision based dual-transformation learning for stain normalization, classification and segmentation. *International Workshop on Machine Learning in Medical Imaging. MLMI*, 12966:477–486, 2021.

[15] Ciga O., Xu T., Nofech-Mozes S., Noy S., Lu F.I, and Martel A.L. Overcoming the limitations of patch-based learning to detect cancer in whole slide images. *Sci Rep*, 11(8894), 2021.

[16] Alzubaidi et al. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(53), 2021.

[17] Xiao C.and Sun J. *Introduction to Deep Learning for Healthcare*. Springer Nature, 2021.

[18] Goodfellow I., Bengio Y., and Courville A. *Deep Learning*. MIT Press, 2016.

[19] Kingma D. and Ba J. Adam: a method for stochastic optimization. 2014.

[20] Dumoulin D. and Visin F. A guide to convolution arithmetic for deep learning. 2018.

[21] Chicco D. and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):1–13, 2020.

[22] Hossin M and Sulaiman M. N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5(2), 2015.

[23] Gianluca B. *Statistical foundations of machine learning, 2nd Edition*. 2021.

[24] Priego-Torres B. M., Sanchez-Morillo D., Fernandez-Granero M. A., and Garcia-Rojo M. Automatic segmentation of whole-slide he stained breast histopathology images using a deep convolutional neural network architecture. *Expert Systems With Applications*, 151, 2020.

[25] Zou K. H., Warfield S. K., Bharatha A., Tempany C. M., Kaus M. R., Haker S. J., and Kikinis R. Statistical validation of image segmentation quality based on a spatial overlap index. *Scientific reports: Academic Radiology*, 11(2):178–189, 2004.

[26] Weishaupt L.L., Torres J., and Camilleri-Broët S. Deep learning-based tumor segmentation on digital images of histopathology slides for microdosimetry applications. 2021.

[27] Guo Z., Liu H., Ni H., Wang X., Su M., Guo W., and Qian Y. A fast and refined cancer regions segmentation framework in wsi breast pathological images. *Scientific reports*, 9(1):1–10, 2019.

[28] Mehta S., Mercan E., Bartlett J., Weaver D., Elmore J., and Shapiro L. Learning to segment breast biopsy whole slide images. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 663–672, 2018.

[29] Wu Z., Li H., Cui L., Kang Y., Liu J., Ali H., and Yang L. Interpretable histopathology image diagnosis via whole tissue slide level supervision. *International Workshop on Machine Learning in Medical Imaging*, pages 40–49, 2021.

[30] Tensorflow. `https://www.tensorflow.org` [Accessed: 2022-08-13].

[31] Scholzen T. and Gerdes J. The ki-67 protein: From the known and the unknown. *Journal of Cellular Physiology*, 182(3):311–322, 2000.

[32] Otsu N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[33] MathWorks. Types of morphological operations, 2022. `https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html` [Accessed: 2022-08-13].

[34] Dougherty E. *Mathematical Morphology in Image Processing (Optical Science and Engineering), 1st edition*. CRC Press, 1992.

[35] Harris T. and Hardin J. W. Exact wilcoxon signed-rank and wilcoxon mann-whitney ranksum tests. *The Stata Journal*, 13(2):337–343, 2013.

[36] Niazi M. K. K., Senaras C., Pennell M., Arole V., Tozbikian G., and Gurcan M. N. Relationship between the ki67 index and its area based approximation in breast cancer. *BMC Cancer*, 18(1), 2018.