# UNIVERSITY OF PAVIA

FACULTY OF ENGINEERING
Department of Electrical, Computer and Biomedical Engineering

Ph.D. in Electronics, Computer Science and Electrical Engineering


---

## Conditional Deep Convolutional Neural Networks for Improved Generalization of Automated Screening of Histopathological Images

---

*Candidate*
**Gianluca Gerard**
XXXIII Cycle

*Advisors*
**Prof. Virginio Cantoni**
**Prof. Marco Piastra**

Academic Year 2019/2020

*Can a machine be made to be super-critical?*

*By A. M. Turing*

# Abstract (English)

The increase in computing power of the last two decades has fueled the growth of a new field of pathology, digital pathology, where glass slides are digitized with high-speed scanners that produce multi-gigabyte images, Whole Slide Images (WSIs). Since the advent of AlexNet in 2012, multiple works have successfully applied deep learning based computer vision to histopathology with performances comparable to the ones of human experts. However, such algorithms have seen limited adoption in the clinical practice of histopathology for two main reasons:

1. deep learning algorithms, trained on datasets of WSIs collected from one or more medical centers, give very accurate results, e.g. classifications of specimen as healthy or diseased, when tested on WSIs from the same centers, but tend to perform significantly worse when tested on datasets of WSIs acquired by different medical centers;

2. deep learning algorithms show limited interpretability of the results, i.e. they offer limited insights into how and why particular results were obtained.

In digital pathology, the first concern is significant because there exist large variations in the data characteristics of histopathology datasets acquired by different medical centers, this is known as *domain shift*. Furthermore, the number of curated, publicly available datasets, to be used for training is limited; as such improving the performance of deep learning algorithms through data volume and standard training techniques is not a viable solution to the domain shift problem.

In this thesis, I focus on studying and implementing a method drawn from the *few-shot learning* paradigm, an approach to train and test deep learning algorithms with few examples, to be robust to domain shift. To address the second concern of limited interpretability, I implemented a Fully

Convolutional Network (FCN), adapted to few-shot learning, for automatic segmentation of metastases in WSIs. Because FCNs output a lesion probability heatmap which can be overlaid on top of the input WSI, the interpretability of algorithmic decisions is easier as it can be related to hue and pattern appearances of the underlying image. The goal is to provide a decision support tool that could assist the pathologists in screening WSIs and that could highlight areas where their evaluation is needed.

To achieve this goal, I focus on histopathology images of sentinel lymph nodes for the diagnosis of breast cancer. WSIs in the dataset were fully annotated by expert pathologists to contour metastases. For this research, I selected and studied a FCN based algorithms whose final predictions can be guided at inference time by providing as input, together with the WSI to segment, a set of other images, known as *support* images, that condition the final output. The architecture is a variant of conditional FCNs (co-FCNs). I also identified a protocol, that partially relies on unsupervised learning techniques, to associate each input WSI to the appropriate support images. Finally, I conducted experiments to evaluate the performance of such algorithm and I compared its performance against a standard FCN semantic segmentation algorithm.

My main contributions are:

- I have studied a few-shot learning method to address the issue of domain shift in digital pathology;
- I have identified and implemented the necessary architectural changes to make the chosen co-FCN architecture applicable to segment WSIs;
- I have devised and implemented a method for the selection of the support set necessary to enable effective few-shot learning.

This is also, to the best of my knowledge, the first study of the applicability of co-FCNs to digital pathology.

# Abstract (Italian)

La crescita degli ultimi due decenni in capacità computazionale ha alimentato lo sviluppo di un nuovo campo in patologia, la patologia digitale, dove i vetrini sono digitalizzati da *scanner* ad alta velocità che producono immagini con dimensioni di diversi Gigapixel, *Whole Slide Image* (WSI). Fin dall'avvento di AlexNet nel 2012, più lavori hanno applicato con successo le tecniche di visione artificiale basate sull'apprendimento profondo (*deep learning*) all'istopatologia con prestazioni confrontabili con quelle degli specialisti umani. Tali algoritmi hanno visto però un'adozione limitata nella pratica clinica per due principali ragioni:

1. gli algoritmi di deep learning, allenati su dataset di WSI raccolti presso uno o più centri medici, forniscono risultati, per esempio le classificazioni di campioni come sani o malati, molto accurati quando provati su WSI degli stessi centri, ma tendono ad avere presetazioni significativamente peggiori quando provati su dataset di WSI acquisite da altri centri medici;

2. gli algoritmi di deep learning mostrano un'interpretabilità limitata dei risultati, in particolare offrono indicazioni limitate sul come e sul perché particolari risultati siano stati ottenuti.

Nella patologia digitale, il primo limite è significativo perché esiste una grande varietà nelle caratteristiche dell'insieme di dati acquisiti da centri medici distinti, questo è noto come *domain shift*. Inoltre il numero dei dati di esempio, curati e pubblicamente accessibili, da usare per l'apprendimento è limitato; per tanto migliorare le prestazioni degli algoritmi di deep learning con un approccio di apprendimento tradizionale basato sul volume dei dati non è una soluzione percorribile al problema del domain shift.

In questa tesi, mi focalizzo sullo studio e sull'implementazione di un metodo tratto dal paradigma di apprendimento con pochi esempi, *few-shot*

*learning*, un approccio per allenare e provare gli algoritmi di deep learning con pochi esempi, con l'obiettivo di essere robusto al domain shift. Per indirizzare il secondo limite dell'interpretabilità limitata, ho programmato una rete completamente convolutiva, *Fully Convolutional Network* (FCN), adattata per il few-shot learning, per la segmentazione delle metastasi nelle WSI. Poiché le FCN forniscono una mappa di probabilità delle lesioni che può essere sovrapposta alla WSI di ingresso, l'interpretabilità delle decisioni algoritmiche è semplificata potendo essere sovente correlata alle differenze di tonalità e tessitura dell'immagine istologica. L'obiettivo finale è quello di fornire uno strumento a Sapporo delle decisioni che possa assistere i patologhi nell'effettuare uno *screening* delle WSI e che possa evidenziare aree del vetrino dove una valutazione sia necessaria.

Per arrivare a questo obiettivo, mi concentro sulle immagini istopatologiche dei linfonodi sentinella per la diagnosi del tumore della mammella. Le WSI di esempio sono completamente annotate da patologhi esperti per contornare le metastasi. Per questa ricerca, ho selezionato e studiato un algoritmo basato su FCN le cui predizioni possono essere guidate durante l'inferenza usando come dato d'ingresso, insieme alla WSI da segmentare, un insieme di altre immagini, definite immagini di *supporto*, per condizionare il risultato finale. L'architettura è una variante delle FCN condizionali (co-FCN). Ho inoltre ideato un protocollo, che in parte si affida a metodi di apprendimento non supervisionato, per associare ogni WSI alle sue immagini di supporto appropriate. Infine ho condotto esperimenti per valutare le prestazioni di questo algoritmo e ho confrontato i risultati prestazionali con un algoritmo FCN di segmentazione semantica tradizionale.

I miei contributi principali sono:

- ho studiato un algoritmo di few-shot learning per indirizzare il problema del domain shift nella patologia digitale;
- ho identificato e programmato i cambiamenti architetturali necessari per rendere l'architettura co-FCN scelta adatta alla segmentazione di WSI;
- ho ideato e implementato un metodo per la selezione delle immagini di supporto necessarie per abilitare un few-shot learning efficace.

Infine questo lavoro è il primo, per quanto di mia conoscenza, che applica le co-FCN alla patologia digitale.

# List of Abbreviations

**AUC** Area Under the Curve

**BCE** Binary Cross-Entropy

**BIC** Bayesian Information Criterion

**CI** Confidence Interval

**CNN** Convolutional Neural Network

**CNP** Conditional Neural Processes

**CT** Computer Tomography

**DL** Deep Learning

**FCN** Fully Convolutional Network

**FSL** Few-Shot Learning

**FSS** Few-Shot Segmenter

**GMM** Gaussian Mixture Model

**GPU** Graphical Processing Unit

**ITC** Isolated Tumor Cell

**ML** Machine Learning

**MRI** Magnetic Resonance Imaging

**pAUC** partial Area Under the Curve

**ROC** Receiver Operating Characteristic

**SE** Squeeze and Excitation

**TL** Transfer Learning

**t-SNE** t-Distributed Stochastic Neighbor Embedding

**WSI** Whole Slide Image

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

Advances in digital imaging and computing power have fueled the rapid growth of digital pathology, where glass slides, commonly diagnosed at the microscope, can now also be digitized and diagnosed with Computer Aided Diagnostic tools. In the past decade many papers have shown that machine learning, and in particular deep learning algorithms, can be used to assist pathologists in their daily routine. Still for a wider adoptions of deep learning tools in the clinical practice, especially for small and mid-sized medical centers, I stipulate that it is necessary for such algorithms to adapt their behavior to match the local demographic, instruments and protocols present in each medical center.

In this thesis, I study and apply a recent variant of a Fully Convolutional Network (FCN), conditional FNCs (co-FCNs), for segmenting lesions in histopathology images of sentinel lymph nodes, and I show that such algorithms can adapt their behavior based on a limited set of manually annotated examples. This implementation could be used to assist pathologists in screening regions of specimens most likely to contain metastases.

My main contributions are:

- I studied conditional FCNs performance applied to digital pathology;
- I propose the first method for proper and automated training of a particular variant of such networks on a histopathological dataset;
- I wrote a full implementation for the training and evaluation of the architecture;
- I show that conditional FCNs can outperform non-conditional FCN architectures when tested on a histopathological dataset of a medical

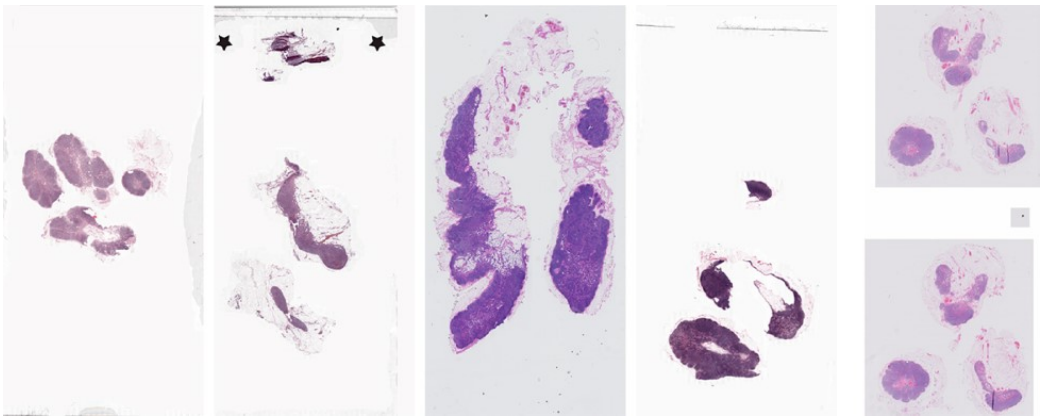


Figure 1.1: Low-resolution examples of WSI as shown in Litjens et al. (2018).

center that use a slide scanner different from the ones used to acquire the training dataset.

To the best of my knowledge this is the first time such networks are used to analyze and segment histopathological images.

## 1.1 Lesions segmentation via Deep Learning in histopathological images: an overview

Histopathologists diagnose diseases with the examination at the microscope of biopsy or surgical specimens processed and fixed onto glass slides. In the past two decades, the exponential growth of computing and storage capabilities have started the field of digital pathology where the glass slides are digitized and then managed as digital images. High-speed scanners digitize glass slides at very high resolution (240 nm per pixel or more) to produce several files whose size are in the order of gigapixels, each called a Whole Slide Image (WSI). Low-resolution examples of WSIs are shown in 1.1.

In this thesis, I focus on the application of the most recent, deep learning based, algorithms to segment lesions, i.e. cancer regions, in WSIs. Since early 2000 machine-learning algorithms have been applied to WSIs for breast cancer classification (Petushi et al., 2006) and since the second half of the past decade deep learning algorithms, Convolutional Neural Network (CNN), have been shown to improve efficiency and accuracy of histopathologic diagnosicts (Litjens et al., 2016).

Breast cancer is the most common form of cancer among women in the Western world. The prognosis depends on whether the cancer has spread to

other organs. *Sentinel lymph nodes*, lymph nodes under the arm near the breast, are the organs which are primarily reached by metastasizing cancer cells and therefore their diagnosis is of critical importance to decide patients treatment.

In sentinel lymph nodes, pathologists sometimes have to isolate tumor cells no larger than 0.2 mm or less than 200 cells in several, 10 or more, slides extracted from each biopsy. This work requires extensive and time consuming sessions at the microscope or in front of a screen. Therefore, the need for tools that can support and assist the work of the pathologists becomes obvious.

A particular variant of deep learning consists in algorithms for the *semantic segmentation* of images. Such algorithms segment images in regions with different semantic meanings. For natural images this could correspond to separating the foreground of an image from the background, in biomedical images, instead, this could be the separation of tumor regions from healthy regions. The first end-to-end and pixel-to-pixel deep learning algorithm for semantic segmentation was the FCN architecture introduced by Long et al. (2015). Since then several deep learning segmentation architectures have been successfully applied to biomedical images (Wang et al., 2019b; Seo et al., 2020).

Semantic segmentation in histopathological images is useful to assist pathologists in locating neoplasia within a tissue. Furthermore semantic segmentation algorithms are computationally efficient in pixel classification when compared to regular CNNs as they are fine tuned to assign a category to every pixel of the input image while traditional CNNs are used for the classification of an entire image. For locating lesions on a WSIs semantic segmentation algorithms provide finer resolution and rely less on scanning through numerous and often overlapping grid of patches superimposed on large WSI. Further details are in Wang et al. (2019b).

## 1.2 Challenges and limitations of traditional deep learning techniques

Domain shifts, i.e. the variations in data characteristics of WSI is a major limiting factor in the widespread deployment of deep learning algorithms in the clinical practice as noted in Stacke et al. (2019). Supervised deep learning, in fact, which entails training on large annotated datasets of images, tend to overfit the training dataset and does not adapt easily to handle images acquired through different protocols.

For histopathological images stain normalization, augmentation and stain transfer techniques are commonly used to improve generalization capabilities, although the usage and actual benefits of such techniques is still debated (see Srinidhi et al., 2021). Standard deep learning regularization strategies, such as data augmentation, drop-out, etc. are also commonly adopted to mitigate the issue.

Another approach relied on adversarial training to regularize a set of WSIs (Ren et al., 2019). According to the authors this technique helps to achieve "significant classification improvement compared with the baseline models".

Google Health approached this and other issues with digital pathology, such as the costs of acquiring the necessary instrumentation, by proposing an integrated Hardware and Software custom image acquisition and processing solution. This device, attached to a microscope, would assist the pathologist by highlighting potentially ill areas in her Field of View (see Chen et al., 2019).

## 1.3 Aim and organization of the thesis

I aim to solve the domain adaptation problem by devising an end-to-end pipeline that learns autonomously to separate local physiological variations in WSIs caused by metastases from global domain specific variations caused by differences of protocols in the preparation of the specimens as well as variances in the digital acquisition pipeline.

My ultimate goal is to design an automated segmentation method that could help the pathologist in screening WSIs areas which need an accurate inspection, and that could adapt and learn, based on a limited number of *supervision* examples, to fit local diagnostic practices and protocols. This scenario applies to many diagnostic centers which do not have the time nor the resources to fully annotate a large set of WSIs to be used for a full 'classical' retraining of a deep learning segmentation model. Therefore it is important to provide an algorithm able to adapt its behavior on a limited set of annotated examples. This approach also satisfies a further requirement: avoiding shifts of the model due to variations over time in the acquisition process and protocol.

Transfer Learning (TL) is the current mainstream approach for training deep learning models on a limited set of annotated examples. Fine-tuning pre-trained initial weights avoids having to re-learn the network weights from scratch and can reduce the time to converge by orders of magnitude. TL has therefore several advantages over training a network from scratch: it is *data efficient*, i.e. less data is needed to train the network, and it can improve

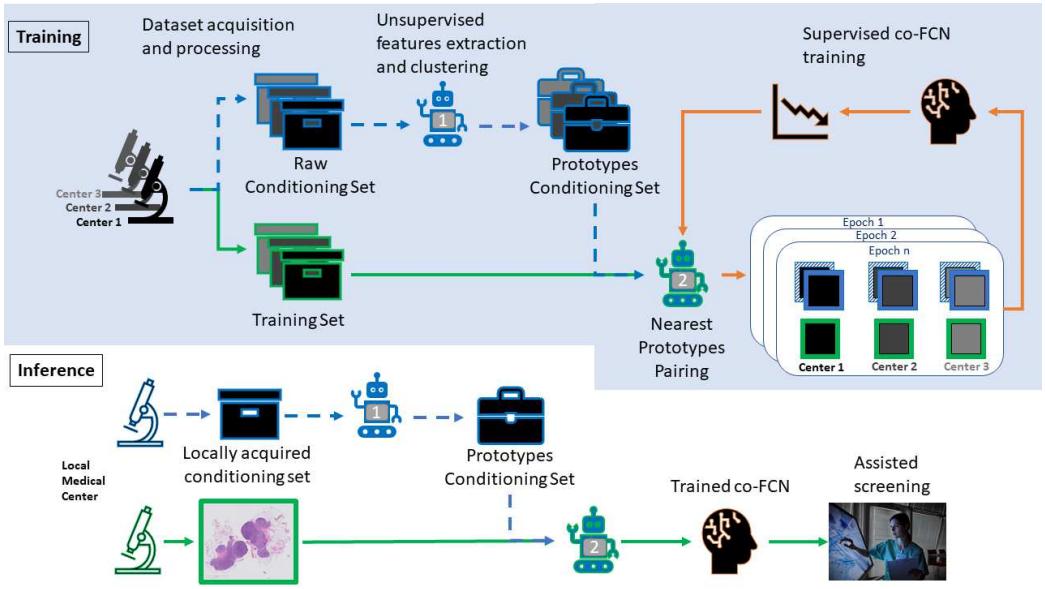



Figure 1.2: A schematic view of the end-to-end process. See the text for an explanation of the process.

both the training speed as well as the model accuracy. However, the benefit of using pre-trained weights greatly decreases as the original task which the network was trained to solve diverges significantly from the target one (see Raghu et al., 2019).

A more general approach that can lead to data-efficient learning is the *few-shot learning* paradigm (Wang et al., 2020). Few-shot learning aims at learning from a limited number of examples with supervised information and from prior knowledge that augments the supervised experience, without incurring the issue of *over fitting* the network to the training dataset. In this research, I used *few-shot learning* methods to achieve the goal of adapting the behavior of the network based on a limited set of supervision examples.

To train and test our algorithm we used the CAMELYON17 dataset as described in Litjens et al. (2018). The WSIs in this dataset are acquired from different medical centers, and it was ideally suited for our purposes as we wanted to train our algorithm under the assumption that it had to perform well even when shown a limited number of annotated samples. Once the CAMELYON dataset is split by medical center, 5 centers contributed just 10 lesion-level annotated slides each, making it perfect to simulate and test a training performed under the few data regime we anticipate should be the target user of our method.

A schematic view of the end-to-end process is shown in Figure 1.2. During

the training a dataset from multiple medical centers is pre-processed to divide each WSI into a grid of patches from which a raw conditioning set and a training set are extracted. With a properly trained autoencoder each patch in the conditioning set is associated with its corresponding feature vector. The feature vectors are then clustered to extract representative prototypes. The same autoencoder extract features from the patches in the training set which allows for the nearest prototypes to be associated with each patch in the training set. The tuples of training patches and associated conditioning patches, are then used iteratively to train the co-FCN. At inference, the trained co-FCN is fed with patches extracted from a single WSI and with a set of prototypes created with the same pre-processing and association pipeline used during the training. The final output segmentation is presented to the pathologist to assist her in the screening and diagnostic process.

This thesis is composed as follows:

- in Chapter 2 I provide a State of Art review of deep learning semantic segmentation for histopathological images, a broad overview of few-shot learning and a summary of related work;
- in Chapter 3 I discuss the methods I use to address the limitations of traditional fully supervised FCN and derived architectures; this chapter details all the processing steps and components used in the diagram shown in 1.2;
- in Chapter 4 I review the experiments I conducted to assess the architecture performance with a comparison against a baseline U-Net model (Ronneberger et al., 2015) trained on the same dataset;
- in Chapter 5 I further discuss the results and provide a summary of the contributions as well as possible future developments.

Chapter 2

# State of Art

In this chapter, I summarize the main applications of Deep Learning (DL) to medical imaging with a focus on biomedical image segmentation and the use of DL in digital pathology. I also introduce, as part of *meta-learning*, the main concepts and techniques on *few-shot learning*, still a relatively new development of DL, and I discuss some of the applications of few-shot learning to the medical domain. Finally, I conclude the chapter by citing the papers that are most closely related to the work described in this thesis.

## 2.1 Deep Learning for medical imaging

DL has had a considerable impact on medical imaging. Litjens et al. (2017) survey and group the applications of DL to medical imaging as such:

- **classification**, including image/exam classification and object/lesion classification;
- **detection**, including organ/region localization and object/lesion detection;
- **segmentation**, including organ segmentation and lesion segmentation;
- **registration**, including spatial alignment of medical images.

Litjens et al. (2017) note that for classification tasks in medical imaging it is very common to fine-tune to the medical task large networks, such as ResNet (He et al., 2016) or Inception (Szegedy et al., 2015), pre-trained on very large, non-medical, datasets, typically ImageNet. This technique is commonly called *transfer learning*. However Raghu et al. (2019) has shown

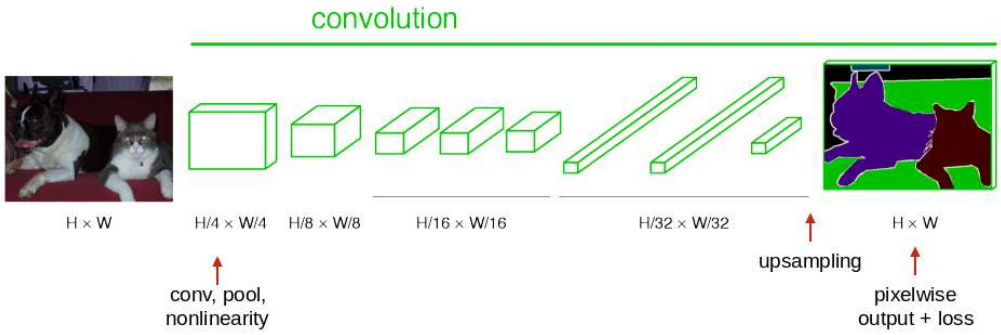



Figure 2.1: The original FCN architecture in Long et al. (2015) as shown in `https://tutorial.caffe.berkeleyvision.org/caffe-cvpr15-pixels.pdf`.

that simpler architectures trained from scratch on medical datasets often yields similar results to the ones trained via transfer learning. The reason may be two-fold:

1. medical imaging tasks usually aim at differentiating between fewer classes;

2. natural images often have distinct boundaries between objects while medical images segmentation is more often based on small changes in the texture of the tissue

In the work described in this thesis we follow Raghu et al. (2019) and we train from scratch our network on two public histopathology datasets: CAMELYON16 and CAMELYON17 (see Litjens et al., 2018).

### 2.1.1 Biomedical image segmentation

For semantic segmentation problem, FCN (Long et al., 2015) and U-Net (Ronneberger et al., 2015) architectures are the most popular in the medical domain as they allow processing inputs of various sizes.

Long et al. (2015) had the insight to replace the fully connected layers of the VGG (Simonyan and Zisserman, 2015) network, a classic CNN for image classification, with convolutional layers whose kernels covered the entire input region. The original FCN is shown in Figure 2.1. Replacing fully connected layers with convolutions enabled the architecture to retain the spatially relevant information and allowed the classification of each image pixels.

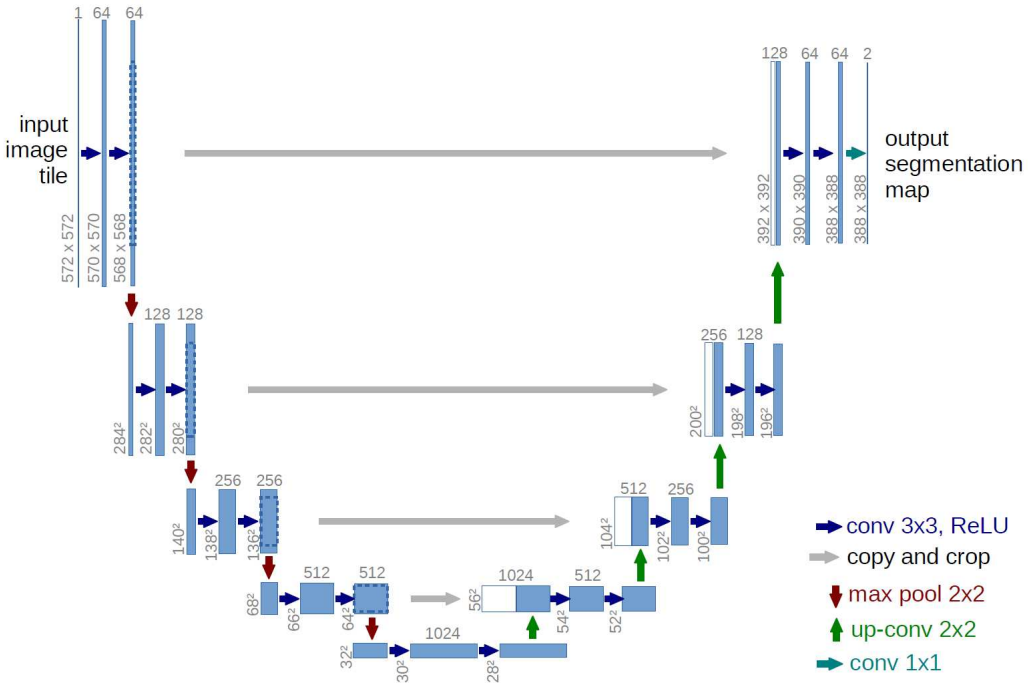



Figure 2.2: The original U-Net architecture in Ronneberger et al. (2015) as shown at `https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/`.

An advantage of FCNs is that, once trained, they will work on input images of any size as long as the computation can fit into memory. This property has been used recently in Chen et al. (2019) that modified an Inception-V3 (Szegedy et al., 2015) network into its fully convolutional form, and applied the resulting architecture to real-time lesion segmentation of histopathology slides through a suitably adapted light-microscope.

The FCN architecture was modified and improved by the U-Net architecture which was originally aimed at segmenting biomedical images. The U-Net, shown in Figure 2.2, has an approximately symmetric shape where the first half of the network, the *contraction* path, is composed by a set of convolutions and pooling layers followed, in the *expansion* path, by another set of convolutions and upsampling layers. The contraction and expansion blocks are connected by skip connections that augment the spatial resolution of the final segmentation output. The original U-Net was initially designed to segment scans from electron microscopy but has since been successfully applied in other medical settings including histopathology images (de Bel et al., 2018). The U-Net architecture, with its two symmetric paths and the skip connections, also represents the basis of the architecture I use in this

work for lesion segmentation of WSIs.

Roy et al. (2019) demonstrated further improvement of FCN based architecture performances on medical datasets by integrating 'squeeze and excitation' (SE) modules, introduced by Hu et al. (2018) for classification networks, into the FCN architectures. Roy et al. (2019), instead of focusing on improving the spatial encoding or the network connectivity of the architecture, investigated how to promote (excite) more informative features, while suppressing the less informative ones, by calibrating the feature maps with appropriate weights. One advantage of SE modules is that they do not increase significantly the model complexity, but, at the same time, allow the model to better learn inter-dependencies between the feature channels. I also evaluated 'squeeze and excitation' modules in my architecture, and as such they are further described in Subsection 3.1.2.

## 2.2 Deep Learning in digital pathology

DL brings a great promise to aid digital pathology. The background, current work, and main challenges have been extensively reported in Dimitriou et al. (2019); Srinidhi et al. (2021). Srinidhi et al. (2021) mention over 130 papers that have been published between 2013 and December 2019. Among these papers eight focus on detection of breast cancer metastases from Haematoxylin and Eosin (H&E) stained slides via segmentation models. The standard approach is by using FCN based models. Most approaches proposed in the literature rely, as I do, on the CAMELYON16 and CAMELYON17 datasets (Litjens et al., 2018).

*Whole Slide Images* (WSIs), glass slides digitized by digital slide scanners, pose some significant challenges that make the adoption of deep neural networks cumbersome compared to other types of medical images: WSI resolutions are very high which results in image files of several gigapixels for each slide. Due to memory constraints and current limitations of the hardware accelerators needed to efficiently execute DL algorithms, including FCNs, WSIs must be divided in patches such that batch of patches can fit into the available memory of the hardware accelerators. Each patch is then processed independently by a forward pass of the network. Dimitriou et al. (2019) further raises the concern of the lack of standardized format that could be adopted by WSI scanners manufacturers, similar to the DICOM standard which is instead adopted widely in radiology.

Persisting problem in computer-aided histopathology analysis is also the domain shift or domain adaptation problem. As noted by Stacke et al. (2019) there can be large differences in data characteristics of WSIs between medical

centers and scanners, making generalization of deep learning to unseen data difficult. Stacke et al. (2019) compare different approaches to resolve domain shift issues including stain normalization and the use of CycleGAN (Zhu et al., 2017) to normalize the WSI to a standard reference dataset. Other approaches are the ones described in Ren et al. (2019) who also describes the use of an adversarial network to solve the domain adaptation problem.

Another challenge of applying standard supervised DL methods to histopathology is the difficulty in obtaining curated and fully labeled datasets for training. Especially with segmentation models the need for accurate dense segmentation requires laborious work from highly trained pathologists. Recently growing attention has therefore been dedicated to applying weakly supervised or unsupervised methods to the field of digital pathology.

Campanella et al. (2019) acknowledges the limitations of current fully supervised models that perform on par with trained pathologists but only on limited and curated datasets. The authors note that these datasets are too small to be representative of the large variability of cases pathologists face in the actual clinical practice. In one experiment they performed, training on CAMELYON17 and then applying the trained model to their own private dataset resulted in a drop of 20% in performances, measured via AUC. Therefore, the authors gathered a very large dataset of over 44732 WSIs from 15187 patients and they proposed to classify slides with a weakly supervised method that only rely on the labels of reported diagnoses for training. According to their evaluations, in order to achieve clinical-grade performance, a training dataset of at least 10000 WSIs is necessary for their method to perform adequately.

Among the weakly supervised method, Lu et al. (2020) introduce an original method, called Clustering-constrained Attention Multiple instance learning (CLAM), a DL based weakly supervised method that uses attention-based learning to identify sub-regions of high diagnostic value to classify WSIs and at the same time be explainable for later review by a pathologist of the decision taken by the algorithm.

Yamamoto et al. (2019) propose an unsupervised approach that starting from a large, over 13000 WSIs, unannotated dataset extract features by using a sequence of two autoencoders. Such features mimic the concept of the Gleason score, thus being explainable features understandable to professionals but also able to identify relevant foretelling features in areas usually not considered relevant. According to the authors the automatically generated features can then supplement human-criteria for accurately predicting the recurrence of prostate cancer. I also rely on a partially unsupervised approach, based on an autoencoder, to extract a representation of the WSI patches suitable to cluster similar WSI patches together as explained in Subsection

3.3.2.

## 2.3 Meta-learning and few-shot learning

As I have already hinted at, the most powerful models based on supervised learning paradigm require large amounts of labeled data to properly solve a specific Machine Learning (ML) problem. This problem not only affects the histopathology domain, but it affects the whole medical domain where biomedical images datasets contain at least one order of magnitude less images than ImageNet (see Raghu et al., 2019). With a "classic" deep learning algorithm, if we use a limited number of training examples, the resulting trained model is prone to overfit the training dataset.

The **meta-learning** paradigm aims to solve these overfitting problems. The idea behind it derives from the observation of us human beings: we are able to learn to solve new tasks with just a few examples. The assumption is that we are able *to generalize* because, by solving previous tasks, we have *learned how to learn*. The *meta-learning problem* can then be defined as follows: *given data/experience on previous tasks, learn to solve a new task more quickly and/or more proficiently*.

Following the formalization provided by a recent survey of meta-learning in Hospedales et al. (2020) we can contrast conventional ML with meta-learning. In conventional ML, we aim at solving a learning task $\mathcal{T}$ defined as the tuple $(\mathcal{D}, \mathcal{L})$ where $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is the *training set* and $\mathcal{L}$ is the loss function that measures the error between ground truth values $y$ and predictions $\hat{y} = f_\theta(x)$ of a model $f_\theta$ parameterized by a set of parameters $\theta$. ML algorithms identify the optimal set of parameters $\theta^*$ solving the following optimization problem

$$\theta^* = \arg\min_\theta \mathcal{L}(\mathcal{D}; \theta, \omega) \tag{2.1}$$

In traditional ML the optimization is performed independently for every task $\mathcal{T}$ and the parameter $\omega$, which denotes the dependence of the solution on external assumptions (the function class for $f$, the choice of the optimizer for $\theta$, etc.) is specified *a-priori*.

Meta-learning, instead, aims at learning $\omega$, i.e. the algorithm for learning itself, hence the common definition of meta-learning as 'learning to learn'. There exist several different approaches to meta-learning each with slightly different theoretical frameworks or views. Hospedales et al. (2020) identify three of them:

- the task-distribution view;

- the bilevel optimization view;
- the feed-forward model view

A common denominator is that they all divide the dataset $\mathcal{D}$ that comes with a task $\mathcal{T}$ into two subsets:

- a training set, $\mathcal{D}^{tr}$, commonly called in the meta-learning literature *support* set;
- a validation set, $\mathcal{D}^{val}$, commonly called *query* set.

The feed-forward model view is the one that provides the best theoretical underpinning for the network I used in this work and that I discuss in Chapter 3. In this view, training of a feed-forward meta-learning model identify the optimal set of parameters $\theta^*$ and $\omega^*$ over a distribution of tasks $p(\mathcal{T})$ such that

$$\theta^*, \omega^* = \arg\min_{\theta,\omega} \mathop{\mathbb{E}}_{\substack{\mathcal{T}\sim p(\mathcal{T}) \\ (\mathcal{D}^{tr},\mathcal{D}^{val})\in\mathcal{T}}} \sum_{(\mathbf{x},t)\in\mathcal{D}^{val}} \mathcal{L}(\mathbf{x}, t; \theta, \mathbf{g}_\omega(\mathcal{D}^{tr})) \tag{2.2}$$

For each task a support set, $\mathcal{D}^{tr}$, and a query set, $\mathcal{D}^{val}$, are drawn. The $\mathbf{g}_\omega$ function embeds the support set $\mathcal{D}^{tr}$ into a vector which is used, along with the model parameterized by $\theta$, to make predictions on the observations $\mathbf{x}$ drawn from the query set $\mathcal{D}^{val}$. In this context we 'learn to learn' by finding the proper embedding function $\mathbf{g}_\omega$ able to extract the information from the support set relevant to reduce the errors on the query set between the predictions and the target values $t$.

At test time, for novel tasks $\mathcal{T}^{te}$ from $p(\mathcal{T})$ (e.g. different domains or different classes from the ones seen at training time), $\mathbf{g}_\omega$ distills the knowledge from the support set of $\mathcal{T}^{te}$ to condition and improve the predictions on the query set of the same task.

One of the main applications of meta-learning in computer vision is on Few-Shot Learning (FSL) methods which, according to Wang et al. (2020), holds promise to "to learn from a limited number of examples with supervised information" . This same author defines formally FSL as "a type of machine learning problems (specified by $E$, $T$ and $P$), where $E$ contains only a limited number of examples with supervised information for the target task $T$" where $E$, $T$ and $P$ have the usual definition of *Experience*, *Task* and *Performance* measure found in Mitchell (1997).

### 2.3.1 Few-shot segmentation

The most common application of FSL in computer vision is for few shot multi-class image recognition, but the technique is also applied for **few-shot**

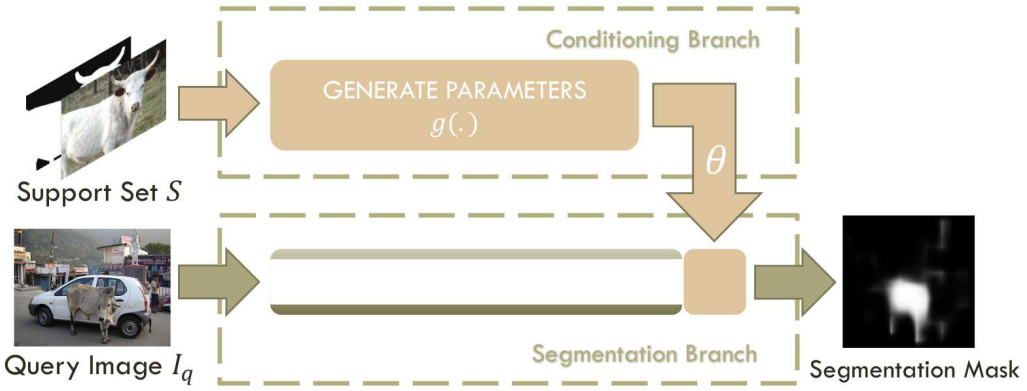



Figure 2.3: The original high level diagram of the two branch architecture in Shaban et al. (2017). The support image with its annotation mask is provided as input to the conditioning branch which derives a set of parameters $\theta$ to be fed into the segmentation branch that takes as input the query image and generates the corresponding predicted mask.

**segmentation**. One of the first attempts at few-shot segmentation is described in Shaban et al. (2017) with a two branches network. One branch receives as input the images of the support set and generates part of the weights of the second branch that segments the images in the query set (see the original high level network diagram in Figure 2.3).

This same approach was further evolved by Rakelly et al. (2018) who use a similar network whose high level architecture is shown in Figure 2.4. The proposed network has again two branches:

- one branch, the conditioning branch, extracts a latent task representation from the support set;
- the other branch, the segmentation branch, segments a query image conditioned on the latent task representation.

Both branches are trained end-to-end in a fully supervised way. In the original model both networks start with the weights of a VGG (Simonyan and Zisserman, 2015) network pre-trained on ImageNet. Contrary to what was done by Shaban et al. (2017), the latent representation extracted from the support set is not used as a set of weights by the guided branch, it is instead tiled spatially and concatenated to the features representation of the query images. A small downstream network subsequently (shown as "Conv" in the segmentation branch of Figure 2.4) learns a metric distance between the query representation and the latent representation to extract the segmentation mask. An original contribution of Rakelly et al. (2018) is to use

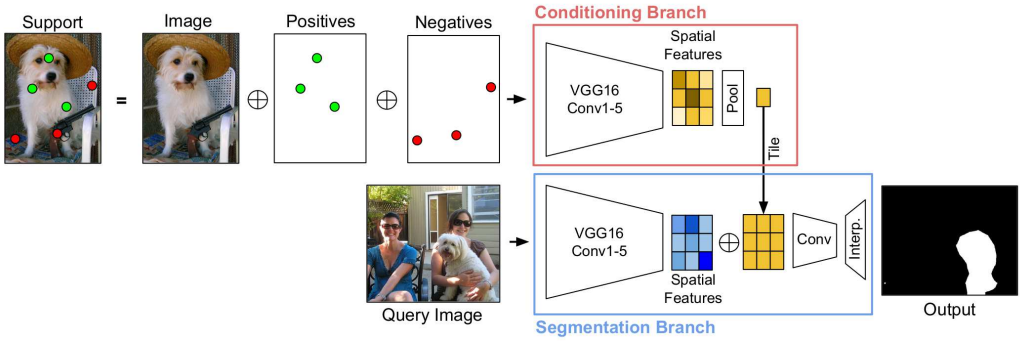



Figure 2.4: The high level diagram of the two branch architecture as shown in Rakelly et al. (2018). In the example the support image has only sparse annotations with the green dots been placed over pixels belonging to the positive class and red dots placed on the negative/background pixels. The features extracted from the support image and the corresponding annotations are pooled spatially and then concatenated to the features extracted by the segmenting branch before been fed to the last layers of the segmenting network that predicts the final segmentation.

annotations of the support set either dense or *sparse*: a dense annotation is the usual annotation where each pixel of an image is classified, a sparse annotation instead classifies only a few pixels of an image with the class of the object the pixels belong to. This is an interesting feature that has the potential to reduce the overhead of annotation and it is a route we have explored in prior work (see Gerard and Piastra, 2019).

## 2.4 Related work

Early work of applying FSL for segmentation tasks, especially by Shaban et al. (2017) and Rakelly et al. (2018), has already been introduced in Section 2.3.

Guha Roy et al. (2020) extend on the work of Rakelly et al. (2018) and introduce a few-shot framework for the segmentation of volumetric medical images. The architecture consists of a conditioning branch, which processes the annotated support input and generates a task-specific representation. This representation is passed on to the segmentation branch that uses this information to segment the new query image. To facilitate efficient interaction between the two branches, the authors propose to use of '*channel squeeze and spatial excitation*' (sSE) blocks (Roy et al., 2019) to enable heavy interaction between both branches with negligible increase in model complexity.

Contrary to the architecture of Rakelly et al. (2018), this contribution allows to perform image segmentation without relying on a pre-trained model. Guha Roy et al. (2020) perform experiments for organ segmentation on whole-body contrast-enhanced CT scans from the Visceral Dataset.

I adopt the high level architecture of Guha Roy et al. (2020) with the necessary changes that make it suitable for the different task of segmenting histopathological images. In particular, I cannot rely on spatial knowledge about the location of the organs, a clearly important information the support set has in the specific case of organ segmentation, and as such I do not use the sSE blocks used by Guha Roy et al. (2020). A full description of my network is provided in Section 3.1.

Other approaches of using limited data to improve the segmentation results in the medical imaging domain are described below. These papers did not influence my research but are provided here for comparison and completeness.

Medela et al. (2019) trained a Siamese Neural Network (van der Spoel et al., 2015) over a dataset of colon tissue images, and applied it as a feature extractor for a dataset composed by healthy and tumoral samples of colon, breast and lung tissue. The resulting low-dimensional representations of the images was used to train a shallow classifier that could perform the classification task with very few samples. The authors claims to achieve a balanced accuracy (BAC), i.e. the mean of sensitivity and specificity, of 90% with 60 training images which outperformed the fine-tune transfer learning approach that obtained 73% BAC with 60 images and required 600 images to get up to 81% BAC. In the process they effectively transferred knowledge from the source domain of colon tissue images to the target domain of colon, breast and lung tissue images.

Wang et al. (2019a) uses mixed-supervised learning, where only a portion of data is densely annotated with segmentation label and the rest is weakly labeled with bounding boxes. The model is trained jointly in a multi-task learning setting. The authors introduce an architecture, Mixed-Supervised Dual-Network (MSDN), with two separate arms for the detection and segmentation tasks. The two networks, similarly to Guha Roy et al. (2020), are connected via 'squeeze and excitation' blocks to transfer information from the branch performing the auxiliary detection task to the branch performing the target segmentation task. The method are tested on CT images for lung nodule segmentation and for cochlea segmentation.

Zhao et al. (2019) present an automated data augmentation method for synthesizing labeled medical images. They demonstrate the method on the task of segmenting Magnetic Resonance Imaging (MRI) brain scans. The method requires only a single segmented scan, and leverages other unlabeled

scans in a semi-supervised approach. They learn a model of transformations from the images, and use the model along with the labeled example to synthesize additional labeled examples. Each transformation is comprised of a spatial deformation field and an intensity change, enabling the synthesis of complex effects such as variations in anatomy and image acquisition procedures. According to the authors, training a supervised segmenter with these new examples provides significant improvements over state-of-the-art methods for one-shot biomedical image segmentation.

Feyjie et al. (2020) propose a model-agnostic few-shot learning framework for semantic segmentation. The model is trained on episodes, which represent different segmentation problems, each of them trained with a very small labeled dataset. Unlabeled images are also made available at each episode. They include surrogate tasks that can leverage supervisory signals, derived from the data itself, for semantic feature learning: according to the authors, including unlabeled surrogate tasks in the episodic training leads to better feature representations, which results in better generalization to unseen tasks. They demonstrate the method in the task of skin lesion segmentation on two publicly available datasets.

## 2.5 Summary

Few-shot semantic segmentation has been increasingly applied to medical images but mostly for radiology (Computer Tomography (CT) scans and MRI). Although their is a clear need, in the histopathology domain, to confront the challenge of acquiring the large annotated datasets traditionally required by deep learning algorithms, few-shot learning to histopathology images is still an underdeveloped area. Most approaches for histopathology have instead explored weakly supervised and unsupervised techniques for lesion classification and segmentation. Data augmentation also seems to play a relevant role in the way the support set is extended and unlabeled data is used. The main motivators been the cost of labelling and the potential of overfitting in low data regime.

When a few-shot segmentation algorithm is applied to the medical domain two branches FCN networks are often used (Guha Roy et al., 2020; Wang et al., 2019a; Feyjie et al., 2020), sometimes in conjunction with ”squeeze and excitation” connection blocks to increase the information transfer between the two networks. From the domain review we also gather that, due to hardware accelerators memory constraints, the current established approach to training on histopathology datasets is to train on patches rather than on full WSIs, effectively treating each patch as independent from each other.

Chapter 3

# Methods

In this chapter, I describe the architecture of the two branch network used for few-shot segmentation. I then discuss how the dataset is reassembled to form the support and query sets and I review the self supervised method I use to associate each query image to its corresponding support images.

## 3.1 Few-Shot Segmenter architecture

The Few-Shot Segmenter (FSS) network architecture, shown in Figure 3.1, is derived from the two branch network of Guha Roy et al. (2020) (see also Section 2.4). The most relevant architectural differences are present in the way the two branches interact, these changes have been introduced to adapt the original architecture, optimized to segment organs in CT scans, to the task of segmenting lesions in histopathology slides.

Both branches are based on the U-Net (Ronneberger et al., 2015) architecture with an approximately symmetrical configuration. The contraction path to the left of each branch $B$, either the *Segmentation* ($S$) or the *Conditioning* ($C$) branch, is a sequence of *encoder* blocks $E_i^B$. The expansion path to the right is a sequence of *decoder* blocks $D_i^B$. Both blocks are shown in Figure 3.2 together with the *bottleneck* $BN^B$ and *classifier* $CL$ blocks. The top branch, the 'Segmentation Branch', takes WSI patches from the query set as input and it outputs their corresponding segmentation masks. The bottom branch is instead the 'Conditioning Branch', it takes as input $k$ WSI patches chosen from the support set according to the policy described in Section 3.3, and it communicates with the segmentation branch through multiple connections. I show these connections in the diagram of Figure 3.1 as inputs to the custom $MX$ blocks which I describe, along with other custom blocks, in Subsection

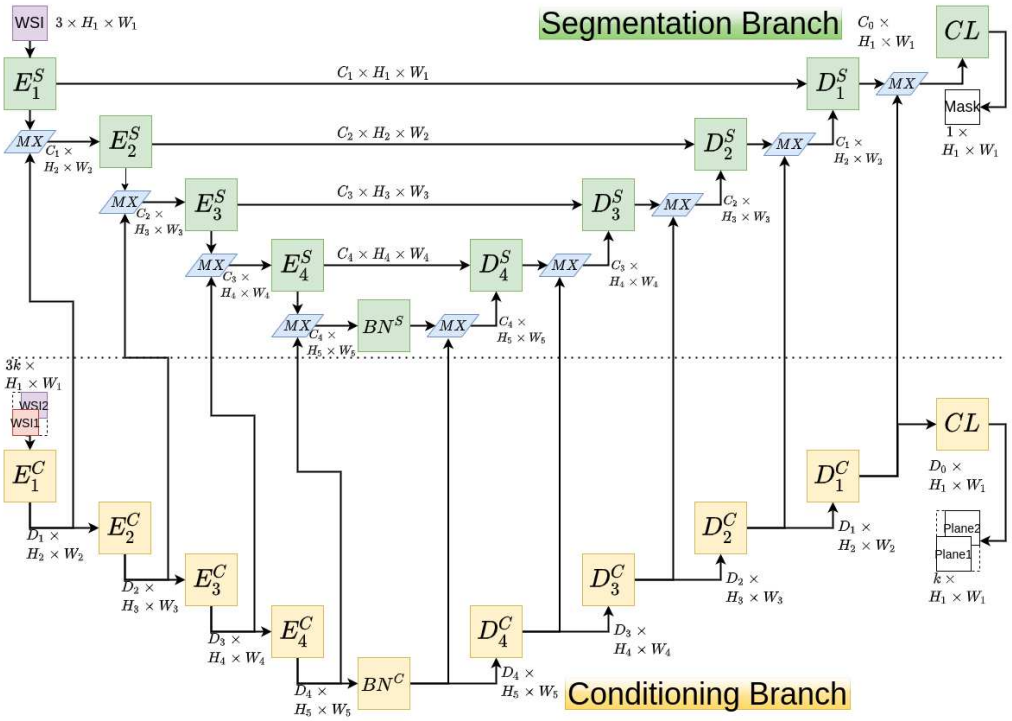



Figure 3.1: The two branch network used for few-shot segmentation. $E_i^B$ and $D_i^B$ are the *Encoder* and *Decoder* blocks for the $B$ branch, either the *Segmentation* ($S$) or the *Conditioning* ($C$) branch. These blocks details are shown in Figure 3.2. The two branches are connected via *MX* blocks, that appear, in the diagram, next to the segmentation branch blocks. The *MX* blocks can take different configurations as shown in Figure 3.3. The $k$ input WSIs to the conditioning branch form the $k$ support *shots*.

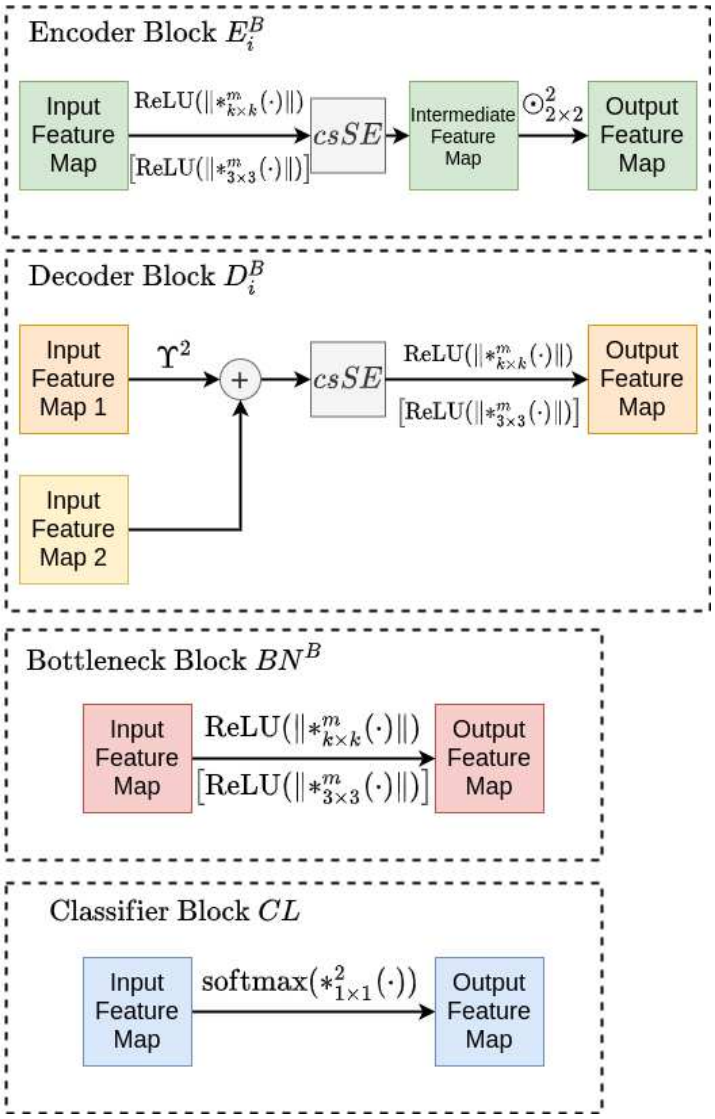



Figure 3.2: The blocks used in the network diagram of Figure 3.1. For the encoder and decoder blocks the *csSE* block (see Figure 3.4) is optional. For the decoder block 'Input Feature Map 2' is optional; when it is present it is concatenated channel-wise with 'Input Feature Map 1' and it is used to handle *skip connections* in the segmentation branch. The meanings of the labels are summarized in Table 3.1, labels in square brackets under the arrows denote optional network layers that can be enabled to sequentially process the output of the layers above the arrows.

3.1.1.

The conditioning branch also outputs, through its classifier $CL$, a feature map of dimension $k \times H_1 \times W_1$. During the network training, this output is averaged across all dimensions and passed through a sigmoid function to output an estimate of the probability that the support patches are extracted from lesion regions (see Subsection 3.3.2 for details on how the probability is derived and Section 3.4 on the training process).

Table 3.1: Operators and functions used in the network blocks shown in Figure 3.2.

| **Symbol** | **Description** |
|---|---|
| $*^{m}_{p\times q}$ | Convolution with $m$ output channels, kernel size $p \times q$ and stride 1 |
| ReLU$(\cdot)$, softmax$(\cdot)$, $\sigma(\cdot)$ | ReLU, SoftMax and sigma functions |
| $\|\cdot\|$ | Batch normalization |
| $\odot^{s}_{p\times q}$ | Maxpooling with kernel size $p \times q$ and stride $s$ |
| $\Upsilon^2$ | Bilinear upsampling with scale factor 2 |

In Figure 3.1 I also include the shape of the feature maps that flows through the blocks connections; a summary of the dimension sizes used for the experiments discussed in Chapter 4 is provided in Table 3.2.

Table 3.2: The dimensions of the features maps flowing in the connections of the FSS diagram shown in Figure 3.1.

| **Parameter** | **Value** |
|---|---|
| $H1$, $W1$ | 128 |
| $H2$, $W2$ | 64 |
| $H3$, $W3$ | 32 |
| $H4$, $W4$ | 16 |
| $H5$, $W5$ | 8 |
| $C1$, $D1$, $D2$ | 32 |
| $C2$, $D3$ | 64 |
| $C3$, $D4$ | 128 |
| $C4$ | 256 |

### 3.1.1 Architectural blocks

The FSS uses five different block types. The first four blocks are shown in Figure 3.2 and are described below:

- the **encoder** block $E_i^B$: it applies once or twice a sequence of a convolution with a variable kernel size ($5 \times 5$ when applied once and $3 \times 3$ when applied twice) and stride 1 followed by a batch normalization layer and a ReLU activation function; the resulting intermediate representation goes optionally through a 'channel and spatial **Squeeze & Excitation** (SE)' block (shown in Figure 3.4 and discussed in Subsection 3.1.2) that outputs the final feature map after a maxpooling operation with kernel size $2 \times 2$ and stride 2. In the segmentation branch, the intermediate feature map can optionally flow through a skip connection to the corresponding decoder block $D_i^S$.

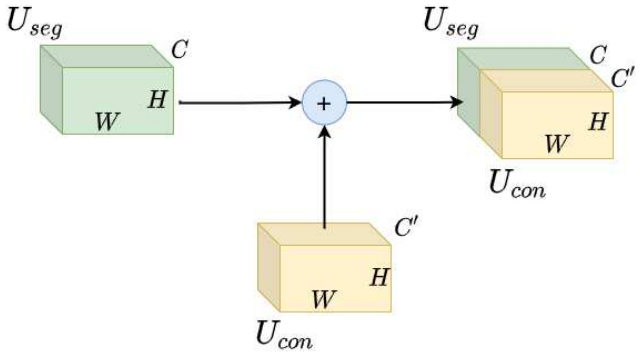
- the **decoder** block $D_i^B$: it applies a bilinear upsampling, $\Upsilon^2$ to the first input map, it optionally adds via channel wise concatenation a second feature map (which in the segmentation branch is the intermediate feature map) and then applies once or twice the same block of convolution, batch normalization and ReLU activation function already described for the encoder block $E_i^B$;

- the **bottleneck** block $BN^B$: it makes only one transformation of the input feature map by applying once or twice the same sequence of convolution, batch normalization and ReLU activation already described for the encoder and decoder blocks;

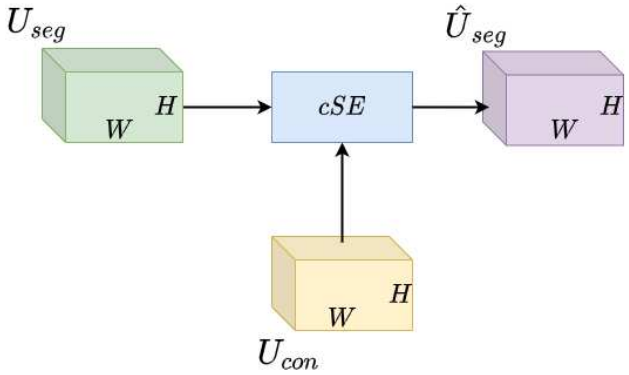
- the **classifier** block *CL*: it applies a softmax function to the output of a convolution with kernel size $1 \times 1$, stride 1 and two output channels; this block effectively outputs one or more planes each containing a binary output mask.

For all blocks, the meanings of the labels used in the architectural diagrams are summarized in Table 3.1 (labels in square brackets indicate optional connections). The presence of the optional *csSE* blocks in the encoder and decoder blocks of both branches is one of the differences between this architecture and the reference architecture described in Guha Roy et al. (2020). In my experiments (ref. Chapter 4) I tested network configurations with and without *csSE* blocks.

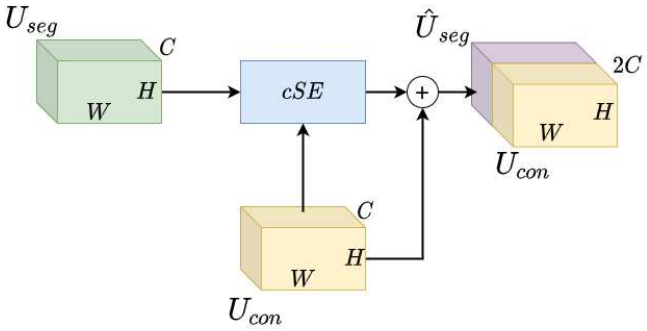
The fifth block type are the *MX* blocks which I show in Figure 3.3. These blocks replace the 'spatial SE' blocks ($SE$) of the reference architecture in Guha Roy et al. (2020). *MX* blocks come in three different flavours:




(a) Features concatenation




(b) Channels weighting




(c) Features concatenation and channels weighting

Figure 3.3: *MX* blocks used in the network of Figure 3.1: Figure 3.3a shows a channel-wise concatenation between feature map $U_{seg}$ and feature map $U_{con}$; in Figure 3.3b a 'channel SE' block, *cSE* (see Figure 3.5 and Subsection 3.1.2 for details) transforms the feature map $U_{seg}$ into a weighted feature map $\hat{U}_{seg}$ with the additional input of the feature map $U_{con}$; Figure 3.3c shows a combination of the previous two configurations with $\hat{U}_{seg}$ concatenated channel-wise with $U_{con}$.

- the *features concatenation MX* block (see Figure 3.3a) takes two input feature maps $\mathbf{U}_{seg}$ and $\mathbf{U}_{con}$ and concatenates them channel wise to output $\hat{\mathbf{U}}_{seg}$;

- the *channels weighting MX* block (see Figure 3.3b) applies a 'channel SE' block (*cSE*), that I introduce in Subsection 3.1.2 (see also Figure 3.5), to the two input feature maps $\mathbf{U}_{seg}$ and $\mathbf{U}_{con}$, and it outputs a weighted feature map $\hat{\mathbf{U}}_{seg}$;

- the *features concatenation and channels weighting MX* block (see Figure 3.3c) applies the *cSE* block to $\mathbf{U}_{seg}$ and $\mathbf{U}_{con}$, and then it concatenates the result channel wise with $\mathbf{U}_{con}$ to produce the final output $\hat{\mathbf{U}}_{seg}$. This configuration has been shortly evaluated but it has not been used for the experiments described in Chapter 4 as it appeared to bring no sensible advantage in the network performances.

The *MX* blocks with concatenation, shown in Figures 3.3a and 3.3c, allow the outputs of the encoder/decoder blocks of the segmentation branch to be concatenated to the outputs of the corresponding encoder/decoder blocks of the conditioning branch. This is partially similar to the way Rakelly et al. (2018) connected the two branches in their architecture. Contrary to Rakelly et al. (2018) though, I use multiple connections instead of just one before the last layers of the segmentation branch.

As an additional difference with the architecture of Guha Roy et al. (2020) I use skip connections in the segmentation branch similar to a U-Net architecture. Skip connections were explicitly removed by Guha Roy et al. (2020) in their architecture as they observed that caused "the network [to copy] the binary mask of the support set to the output". In my case the support input does not contain the binary masks, another important difference in the way the network is used, and the presence of skip connections regain their usual effect of enhancing the spatial resolution of the segmentation output.

I tested various configurations of the network connections to identify the one that could better transfer information between the two branches: a full discussion of the different configuration results is provided in Chapter 4.

### 3.1.2 Squeeze and Excitation blocks

The encoder and decoder blocks discussed previously both optionally use the 'channel and spatial SE' block shown in Figure 3.4; it was introduced by Roy et al. (2017) for CNNs and then adopted in FCN based architectures by Hu et al. (2018) and Roy et al. (2019).

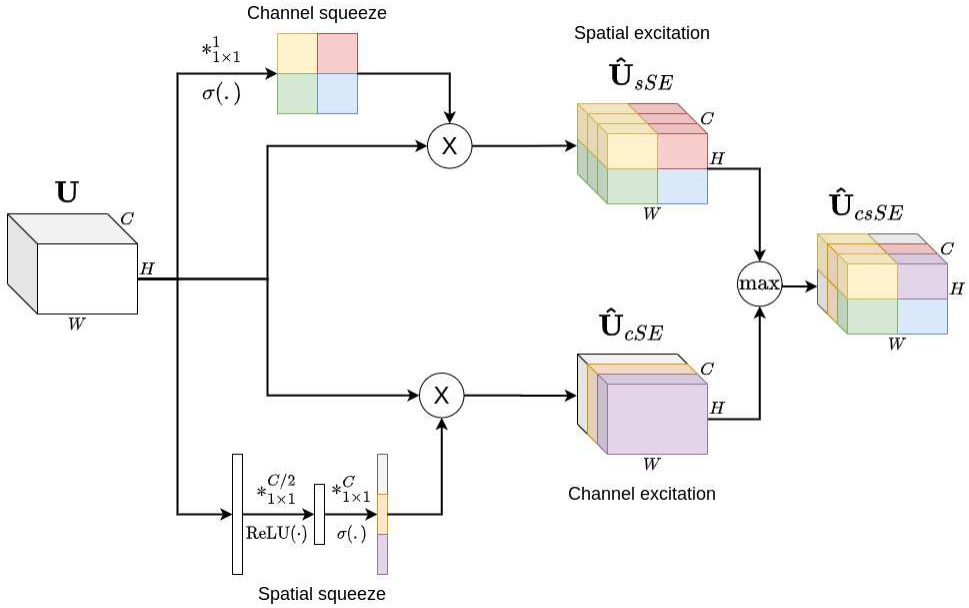



Figure 3.4: The *csSE* schema. The input feature map **U** is transformed through two paths, the 'Channel squueze' and the 'Spatial squeeze' as discussed in the text. The results are composed with a element-wise max operation which produces the end result $\hat{\mathbf{U}}_{csSE}$.

The connection descriptions used in the diagram of Figure 3.4 have the same meaning as shown in Table 3.1. The *csSE* block takes as input a feature map with $C$ channels, **U**, and it transforms it through two paths:

- the first path, 'Channel squeeze', applies a $1 \times 1$ convolution with stride 1 and 1 output channel and a sigma function to obtain a 'channel squeezed' representation of **U**. Spatial regions of **U** with more active channel features translate into more active regions in the one dimensional representation;

- the second path, 'Spatial squeeze', applies two fully convolutional layers, the first one outputs $C/2$ channels, and the second one $C$ channels. The output vector representation has higher activation values associated with channels with higher average activation in the original **U** map.

The two representations are used as weights that multiply the input feature map **U** to obtain two new intermediate feature maps: $\hat{\mathbf{U}}_{sSE}$ and $\hat{\mathbf{U}}_{cSE}$ as shown in the diagram of Figure 3.4. An element-wise max operation between these two intermediate maps provide the final output map $\hat{\mathbf{U}}_{csSE}$.

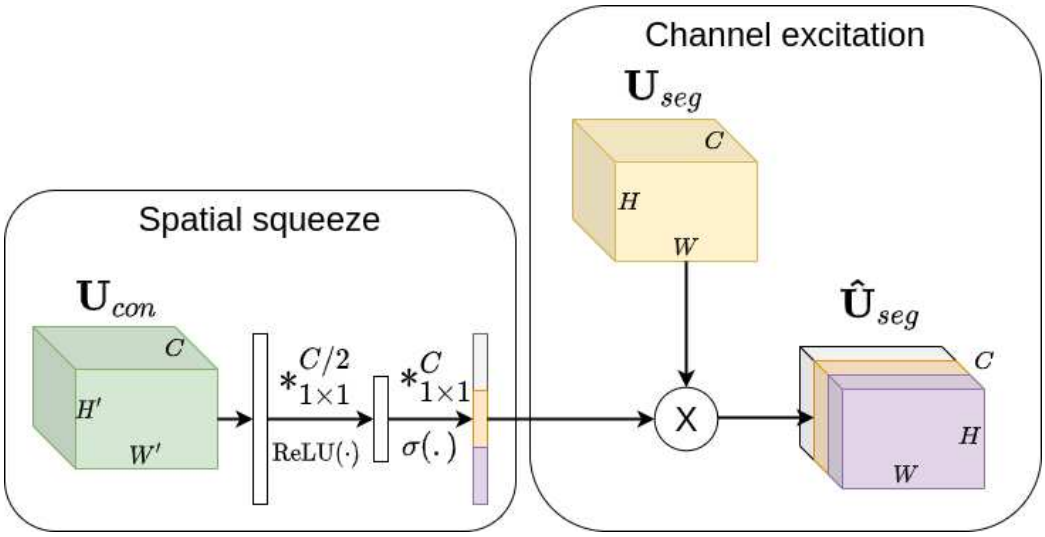



Figure 3.5: Spatial squeeze and channel excitation used in the *MX* blocks to connect the conditioning and segmentation branch of the FSS.

Another SE block used by my network, in particular by the two *MX* blocks reviewed in Subsection 3.1.1, is the 'channel SE' *cSE* block shown in Figure 3.5. Logically the *cSE* block performs two actions:

- in the 'spatial squeeze' section the input feature map $\mathbf{U}_{con}$ with $C$ channels, which in the FSS comes from the Conditional branch, is squeezed spatially through two fully connected layers that output a vector with $C$ features. This is conceptually similar to the 'Spatial squeeze' path of Figure 3.5;

- the vector output from the 'spatial squeeze' is used as weights that multiply, channel wise, the input feature map $\mathbf{U}_{seg}$, which in the FSS comes from the an Encoder or Decoder block of the segmentation branch, to produce a weighted output feature map $\hat{\mathbf{U}}_{seg}$.

## 3.2 Dataset and experimental setup

For my experiments, I rely on the CAMELYON17 dataset (Litjens et al., 2018). This dataset contains lesion-level annotations for 10 slides from 5 different medical centers (50 slides in total). Some slides are associated with the same patient. Table 3.3 maps each medical center ID to the actual medical center (center for short) and scanner model used for data collection.

The full name of each center and further details on the dataset composition is available in Litjens et al. (2018).

Table 3.3: The association between medical center ID, medical center acronym and digital scanner used for the CAMELYON17 dataset.

| **ID** | **Center** | **Scanner** |
|---|---|---|
| 0 | CWZ | 3DHistech Pannoramic Flash II 250 |
| 1 | RST | 3DHistech Pannoramic Flash II 250 |
| 2 | UMCU | Hamamatsu NanoZoomer-XR C12000-0 |
| 3 | RUMC | 3DHistech Pannoramic Flash II 250 |
| 4 | LPON | PhilipsUltrafast Scanner |

In order to mimic a possible clinical scenario, where the network is trained on a dataset of WSIs acquired by medical centers different from the ones that would use the network, I defined the following rules to distribute the CAMELYON17 WSIs among the training and test sets:

- WSIs in the training and test sets had to come from different medical centers;
- a portion of WSIs in the test set had to be acquired with a different digital scanner from that used in the training set;
- for control, the remaining WSIs in the test set had to be acquired with a scanner also used for the acquisition of a portion of WSIs in the training set.

The introduction of such rules allows to quantify the presence and impact of the issue of domain shift in the dataset, i.e. the degradation of deep learning algorithm performance trained on a distribution of WSIs different from the one used during inference (see Section 2.2 for details), and if the conditional FCN could cope and alleviate the issue.

As a result I trained on WSIs from medical centers 0, 1 and 2 of the dataset. I used the rest of WSIs from center 3 and center 4 as test set.

Another preliminary choice necessary for few-shot learning is how to distribute the WSIs of each medical center among the support and query sets. Because we have a limited number of WSIs, each of them with significant

differences in the extent and number of lesions, the distribution must be executed carefully to avoid having marked unbalances between the data characteristics of the slides assigned to the support and query sets. Patients in the CAMELYON17 dataset, in fact, are classified by their pN-stage based on the number and type of metastases they have:

- macro-metastases: metastases greater than 2.0 mm;
- micro-metastases: metastases greater than 0.2 mm or more than 200 cells, but smaller than 2.0 mm;
- Isolated Tumour Cells (ITCs), strictly not a metastasis, but defined as: single tumour cells or a cluster of tumour cells smaller than 0.2 mm or less than 200 cells.

The challenge for a screening technique is to detect micro-metastases and ITCs, because they are very difficult to spot, but they are also critical for early diagnosis.

The pN-stage classification used in CAMELYON17 is the following:

- pN0 - no micro-metastases or macro-metastases or ITCs found;
- pN0(i+) - only ITCs found;
- pN1mi - micro-metastases found, but no macro-metastases found;
- pN1 - metastases found in 1-3 lymph nodes, of which at least one is a macro-metastasis;
- pN2 - metastases found in 4-9 lymph nodes, of which at least one is a macro-metastasis.

as described in `https://camelyon17.grand-challenge.org/Evaluation/`.

To maintain balance among the query and support sets in terms of patients pN-stage, I manually distributed, for each center, the patients among the two sets with the final distribution shown in Table 3.4. The first number of each pair in the Table 3.4 is the number of WSIs assigned to the query set and the second number is the number of WSIs assigned to the support set.

In CAMELYON17, each patient is associated with one or more slides, and for each slide the presence of ITCs, micro-metastases, macro-metastases or the absence thereof is recorded. Therefore another rule I applied to the distribution among query and support sets was also to balance the distribution of WSI by the type of lesions they contained. I show in Table 3.5 the number

Table 3.4: Each row of the table shows, per medical center, the number of patients classified at a particular stage present in the query set (first number in the pair) and in the support set (second number in the pair).

| **Center ID** | **pN0** | **pN0(i+)** | **pN1** | **pN1mi** | **pN2** |
|---|---|---|---|---|---|
| 0 | 0, 0 | 1, 0 | 3, 1 | 2, 0 | 0, 0 |
| 1 | 0, 0 | 1, 0 | 2, 1 | 1, 2 | 1, 0 |
| 2 | 0, 0 | 2, 0 | 2, 0 | 1, 1 | 2, 1 |
| 3 | 0, 0 | 0, 2 | 2, 2 | 1, 2 | 0, 1 |
| 4 | 0, 1 | 1, 1 | 1, 1 | 1, 1 | 0, 2 |

Table 3.5: Lymph-nodes of different centers are stratified by the presence of ITCs, micro-metastases (micro), macro-metastases (macro) or the absence of either (negative). Each cell in the table represents how the stratified slides for each center are distributed among the query (first number in the pair) and support sets (second number in the pair).

| **Center ID** | **ITC** | **macro** | **micro** | **negative** |
|---|---|---|---|---|
| 0 | 2, 1 | 3, 1 | 2, 1 | 0, 0 |
| 1 | 2, 0 | 1, 1 | 3, 1 | 1, 1 |
| 2 | 2, 0 | 3, 1 | 3, 1 | 0, 0 |
| 3 | 1, 3 | 1, 2 | 1, 2 | 0, 0 |
| 4 | 1, 3 | 1, 2 | 1, 0 | 0, 2 |

of WSIs assigned to the query and support set of each center, grouped by the lesion types they contain (pair of numbers are ordered as in Table 3.4).

For processing I follow the common approach of splitting the WSIs in a grid of patches, each 128 x 128 pixels wide. I use slides at 20x magnification ($0.5\mu$m pixel$^{-1}$). To reduce the computing time I also discard patches containing no tissue, i.e patches whose maximum intensity value of the saturation channel, after having applied a Gaussian blur, is below 10% of the full saturation. I retain instead all patches that have at least half of the pixels in the 64x64 central region classified as lesion[1]. Of the remaining patches, due to heavy class imbalance, I drop 85% of them in the support sets and 95% in the query sets. In tables 3.6 and 3.7 I show, for each center, the total number of patches assigned to the support and to the query sets along with their classification as either lesion or non-lesion patches. Because the WSIs

Table 3.6: The number of patches, either classified as lesion or non-lesion, extracted from the WSIs of each medical center and assigned to the support set. The non-lesion patches are decimated by 85% due to heavy class imbalance. All patches that have at least 50% of pixels classified as lesion in the central 64x64 region are retained.

| **Center ID** | **non-lesions** | **lesions (ratio)** | **Total** |
|---|---|---|---|
| 0 | 8577 | 2551 (22.9%) | 11128 |
| 1 | 10402 | 1264 (12.2%) | 11666 |
| 2 | 17623 | 3372 (16.1%) | 20995 |
| 3 | 67032 | 24547 (26.8%) | 91579 |
| 4 | 72898 | 62533 (45.5%) | 135431 |

from centers 3 and 4 are used for test, I have assigned the majority of WSIs of these two centers to their support sets, hence the higher number of total patches for these centers with respect to the centers 0, 1 and 2.

## 3.3 Selection of the support set

The FSS architecture assumes that for each query patch presented to the segmentation branch one or more support patches, i.e. one or more *shots*, are presented to the conditioning branch to guide the segmentation.

[1]Other authors classify as lesion patches all the patches that have at least one pixel classified as lesion in the central region. This choice, however, increases the misclassification of patches falling at the borders of lesions, especially for macro-metastases where the annotations in the CAMELYON17 dataset are not pixel level accurate.

Table 3.7: The number of patches, either classified as lesion or non-lesion, extracted from the slides of each medical center and assigned to the query sets. The non-lesion patches are decimated by 95% due to heavy class imbalance. All patches that have at least 50% of lesion pixels in the central 64x64 region are retained. Only centers 0, 1 and 2 are listed as centers 3 and 4 are used for validation and test only, as such the inferences are run on the full WSIs of these two centers that are not in their support sets.

| **Center ID** | **non-lesions** | **lesions (ratio)** | **Total** |
|---|---|---|---|
| 0 | 13748 | 5380 (28.1%) | 19128 |
| 1 | 28351 | 3559 (11.2%) | 31910 |
| 2 | 41898 | 16483 (28.2%) | 58381 |
| tot | 83997 | 25372 (23.2%) | 109369 |

The choice of how to associate each query patch to its corresponding support set is a major architectural decision. In previous work (see Gerard and Piastra, 2019) we used a simple random approach where the only constraint was that at least half of the shots had to be classified as lesion patches and the rest as non-lesion. Follow up experiments highlighted that this simple strategy caused great variance in the output segmentation at inference time.

Therefore I tested and refined an original process, which partially relies on unsupervised methods to extract a latent representation of the patches and on clustering methods to facilitate the query to support patches association, which is summarized below and for which I provide further details in the following sections.

At an high level, the support patches associated to each query patch are chosen with the following protocol:

- for each medical center, a separate convolutional autoencoder is trained on a portion of the support set patches to learn a latent representation; the latent representation is averaged spatially to produce a vector with 8 components for each patch, and the number of components is reduced to 3 components via a Principal Component Analysis;

- the support set patches, in their reduced three dimensional encoding, are clustered with a Gaussian Mixture Model clustering algorithm (GMM);

- I run a second level of separate clustering among all the support patches belonging to each GMM cluster in order to only retain, for each GMM cluster, only the most representative, 'prototype', patches. For this task

I use a *k-means* clustering algorithm with Euclidean distance and with a dynamic number of clusters computed by dividing the total number of patches belonging to each GMM cluster by a fixed quantity;

- once k-means sub-clusters have been identified for each GMM cluster, the support patches that are closest to the center of each sub-cluster are retained as prototypes;

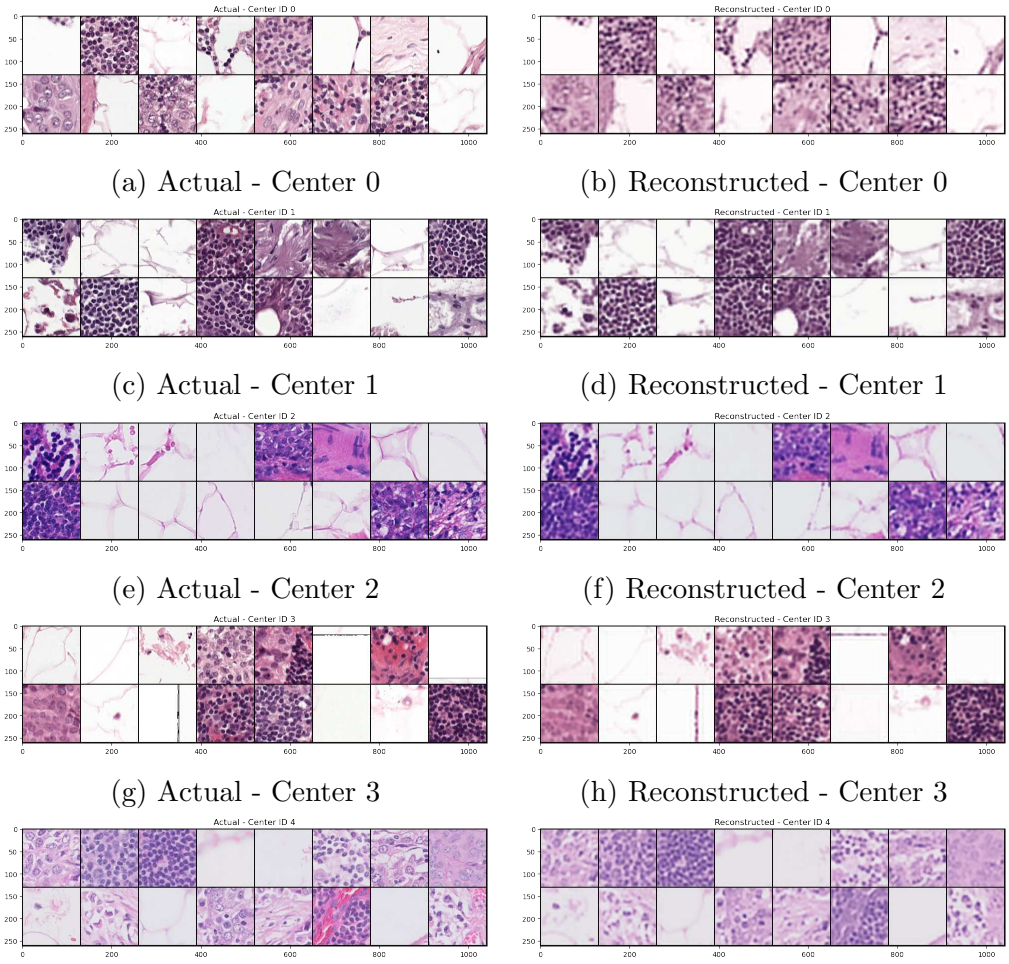
- for each query patch, $k$ prototypes, extracted from the same GMM cluster of the query patch, are chosen according to the policy described in Section 3.4.

The number of clusters to use must be provided as input to the GMM clustering. For the training phase of the FSS I chose the same number of clusters, 6, for all medical centers based on a preliminary data exploration. Further details on the above protocol are provided in the next subsections.

### 3.3.1 Latent representation of the patches with a convolutional autoencoder

80% of the support patches of each center are used to train a convolutional autoencoder, one for each medical center, which encode each patch into a latent representation with 8 channels and spatial shape $16 \times 16$. An example of the reconstruction of multiple patches by such autoencoders is shown in Figure 3.6. The reconstruction by the autoencoders is accurate although the output is slightly blurred with the finer details lost and with some of the most unusual hues replaced with the average hue associated with the patches of their medical center. Hue variations among the slides of different medical centers is also apparent from the random sampling of patches shown in Figure 3.6.

The convolutional autoencoder, whose architecture is shown in Figure 3.7a, has a contraction path with 3 encoder blocks. Each encoder block (see Figure 3.7b) is a convolution with kernel size $3 \times 3$, stride 1 and padding 1, followed by a ReLU activation unit and a maxpooling layer with stride 2. Each encoder halves the input dimensions. The first encoder outputs 16 channels, the second and the third 8 each. Symmetrically, the expansion path has a sequence of 3 decoder blocks and a final convolution. Each decoder block (see Figure 3.7c) is a sequence of a convolution with kernel size $3 \times 3$, stride 1 and padding 1, a ReLU activation and an upsampling layer that uses a nearest neighbor algorithm to double the input dimensions. The first two decoders output feature maps with 8 channels and the last encoder outputs




(a) Actual - Center 0

(b) Reconstructed - Center 0

(c) Actual - Center 1

(d) Reconstructed - Center 1

(e) Actual - Center 2

(f) Reconstructed - Center 2

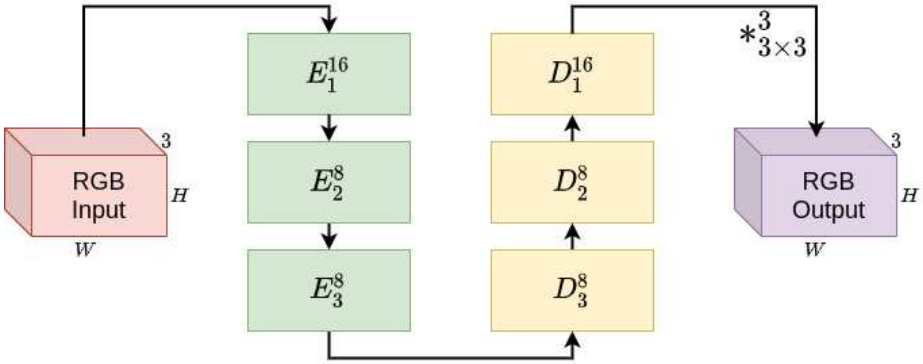
(g) Actual - Center 3

(h) Reconstructed - Center 3

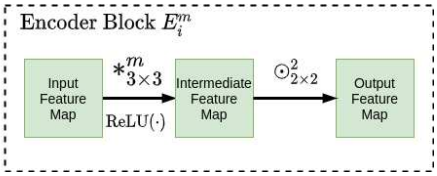
(i) Actual - Center 4

(j) Reconstructed - Center 4

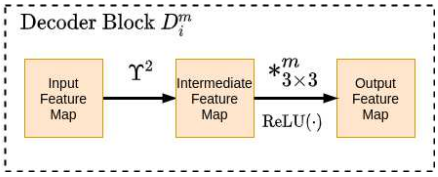
Figure 3.6: For each medical center, I show 16 random patch examples and their reconstruction obtained as output of the convolutional autoencoders trained separately on each center.




(a) Convolutional autoencoder architecture




(b) Encoder Block $E_i^m$




(c) Decoder Block $D_j^m$

Figure 3.7: The convolutional autoencoder used for unsupervised learning with its building blocks is shown in Figure 3.7a. Each encode/decode block, $E_i^m$ and $D_j^m$, has a convolution with padding and a $3\times3$ kernel with $m$ output channels, followed by a ReLU activation layer. The encoding block, which is shown in Figure 3.7b, has a maxpooling layer at the end while instead the decoder block, which is shown in Figure 3.7c, has an upsampling layers, which uses a nearest neighbor algorithm, as its first layer.

a feature map with 16 channels. The final convolution has a kernel size $3\times3$, 3 output channels and no padding.

Each autoencoder is trained with an Adam optimzer and an initial learning rate of 0.004. Early stopping is used to avoid overfitting. The initial weights of each autoencoder are set based on an identical autoencoder trained on a sample of WSIs from the CAMELYON16 dataset.

### 3.3.2 Unsupervised clustering of the support patches

The autoencoder is used to extract a latent representation of each support patch. The latent representation obtained by the encoder is spatially averaged to produce a feature vector with 8 elements for each patch, the number of components is further reduced to 3 by using a Principal Component Analysis. The resulting 3 dimensional vectors associated to a sample of the support

patches of each medical center are clustered by using a GMM clustering algorithm in the implementation of the Scikit-learn library (Pedregosa et al., 2011). The GMM of each medical center is trained independently. I chose the initial number of clusters (number of components), 6, by searching for a common number that would minimize the Bayesian information criterion (BIC) for all centers. Each cluster component has its own general covariance matrix.

A t-SNE representation of the clustering obtained for all medical centers is shown in Figures 3.8 - 3.12 along with the lesion/non-lesion classification of each patch. Each point in the t-SNE graph represents a patch: the GMM clusters labels are shown in the left graphs and the lesion/non-lesion classification appears on the right graphs. A patch is classified as a lesion patch if more than half of the pixels in its central 64x64 region are lesion pixels.

A visual inspection of the graphs in all Figures 3.8 through 3.12 shows a strong correspondence between at least one cluster of each medical center and the regions with the highest density of lesion patches. This strong correspondence is quantified in Table 3.8.

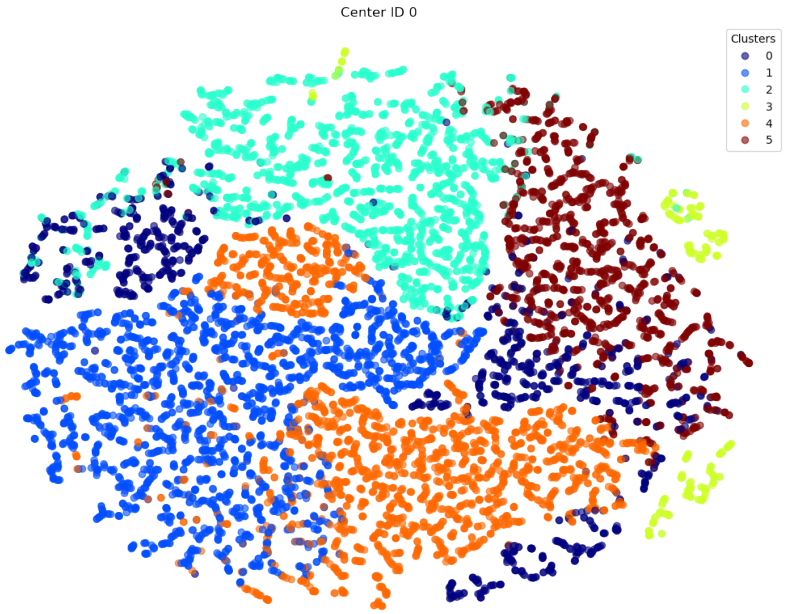
In Table 3.8 I show the percentage of lesion and non-lesion patches assigned to each cluster out of the total number of lesion and non-lesion patches of each medical center. These values are computed on a random sampling containing 20% of the patches in the support set of each medical center.

The last column of Table 3.8 contains the following value

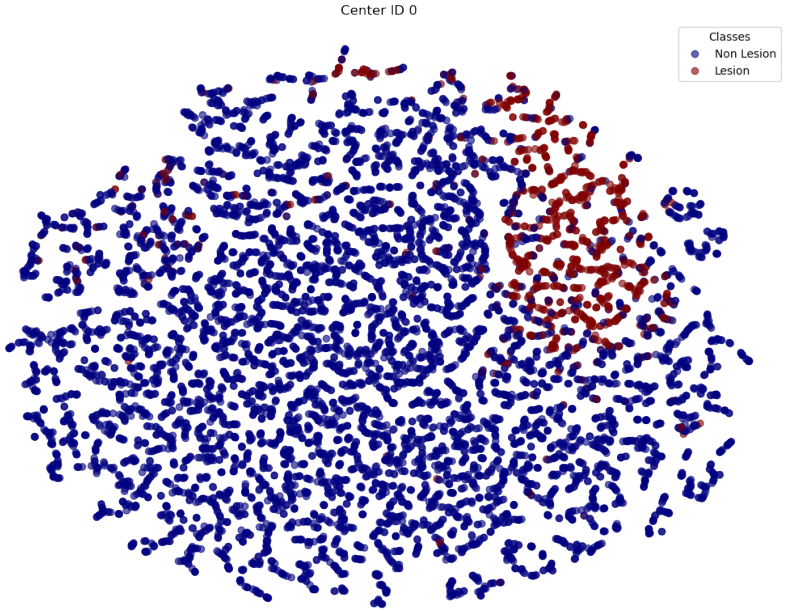
$$\pi_l(c, g) := \frac{r_{pos}(c, g)}{r_{pos}(c, g) + r_{neg}(c, g)} \tag{3.1}$$

where $r_{pos}(c, g)$ is the ratio of lesion patches of medical center $c$ associated with cluster $g$, and $r_{neg}(c, g)$ is the same ratio for non-lesion patches. The value $\pi_l(c, g)$ is the estimated probability of lesion patches inside each cluster: values close to 1 signals clusters where lesion patches are prevalent, values close to zero signals instead clusters containing mostly non-lesion patches. I use this lesion probability estimate to choose the support patches to fed to the conditioning branch for each query patch as described in Section 3.4.

The results in Table 3.8 show that some clusters are highly correlated with the presence of lesion patches. In fact, if we look at the clusters that contain the largest portion of lesion patches of each center (underlined cluster numbers in Table 3.8) we see that they always have a far lower proportion of non-lesion patches. In a fully unsupervised way we have therefore extracted and learned relevant physiological differences among the patches somewhat confirming the results of Yamamoto et al. (2019), who used a fully unsupervised approach to extract "explainable features" from histopathology images by using a combination of autoencoders and clustering techniques.

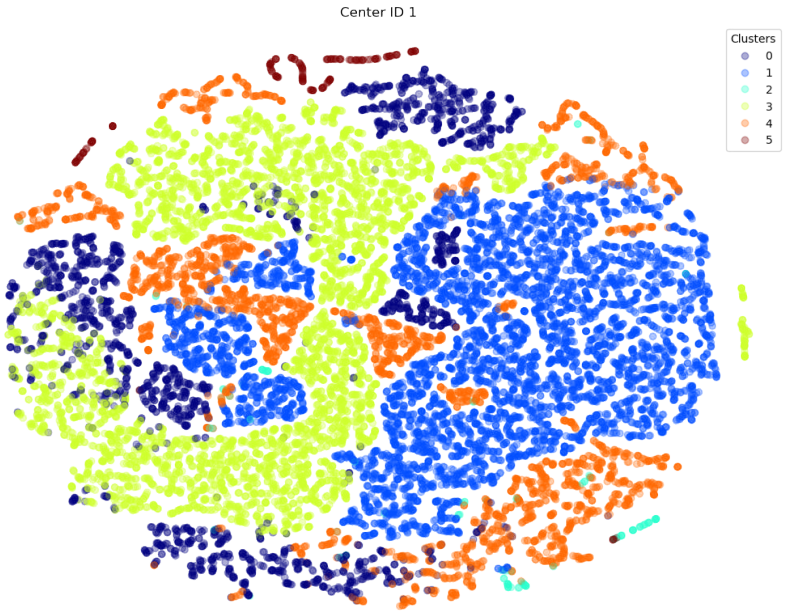



(a) GMM Clusters - Center 0

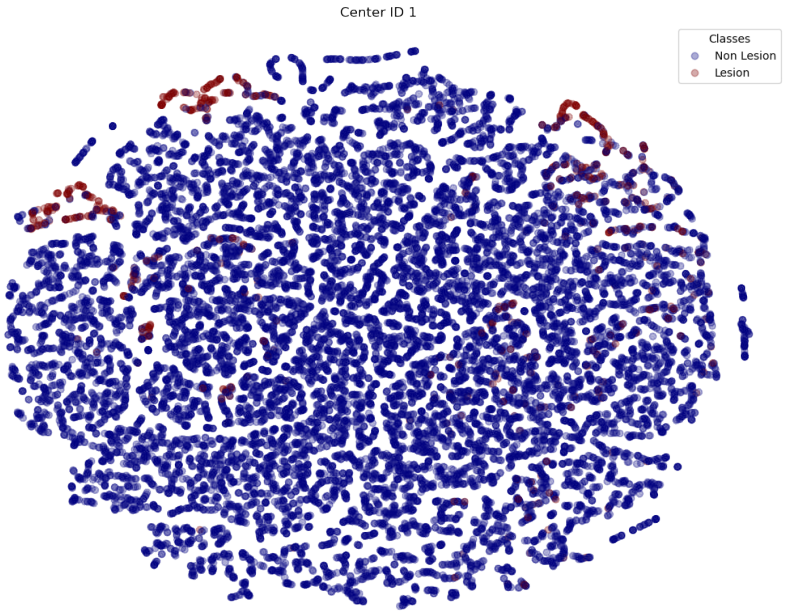



(b) Lesion/Non-Lesion - Center 0

Figure 3.8: t-SNE representation of the GMM clusters and the corresponding lesion/non-lesion classification on a sample of the support patches of medical center 0. Figure 3.8a shows the different clusters identified by the GMM algorithm. Figure 3.8b shows in red the patches classified as lesions and in blue the patches classified as non-lesion.

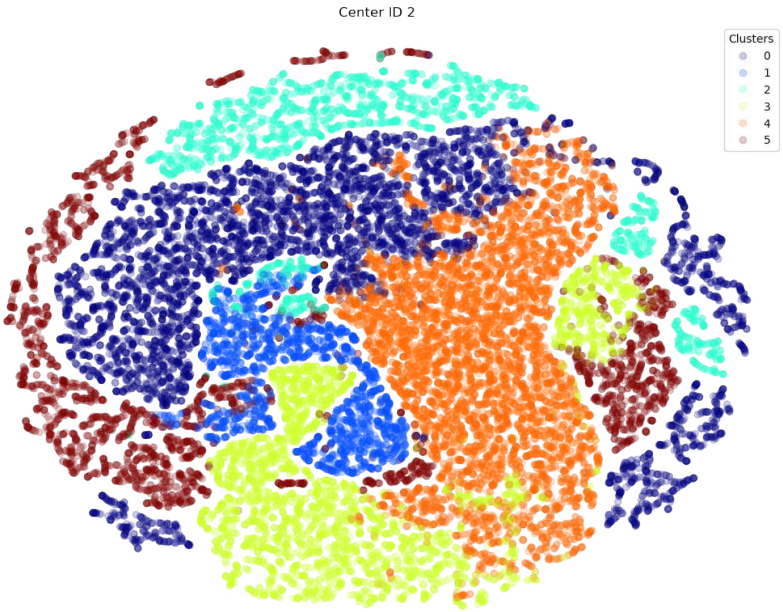



(a) GMM Clusters - Center 1

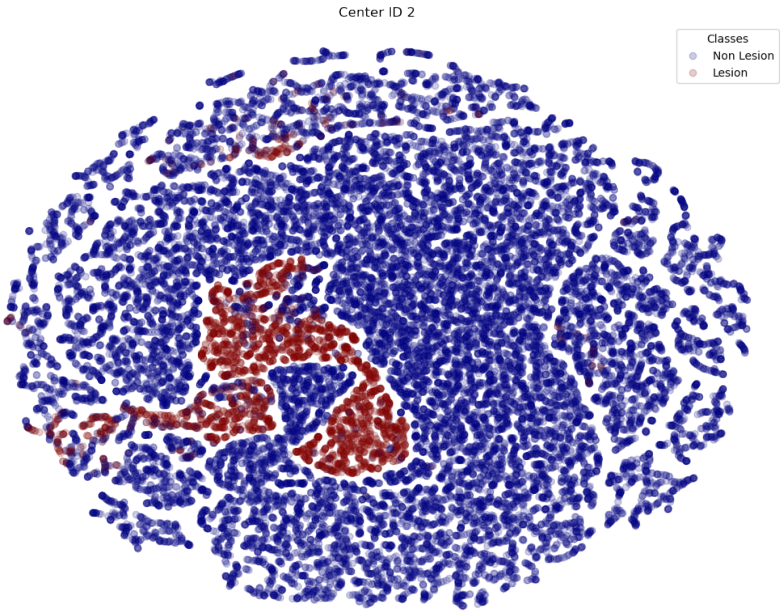



(b) Lesion/Non-Lesion - Center 1

Figure 3.9: t-SNE representation of the GMM clusters and the corresponding lesion/non-lesion classification on a sample of the support patches of medical center 1. Figure 3.9a shows the different clusters identified by the GMM algorithm. Figure 3.9b shows in red the patches classified as lesions and in blue the patches classified as non-lesion.

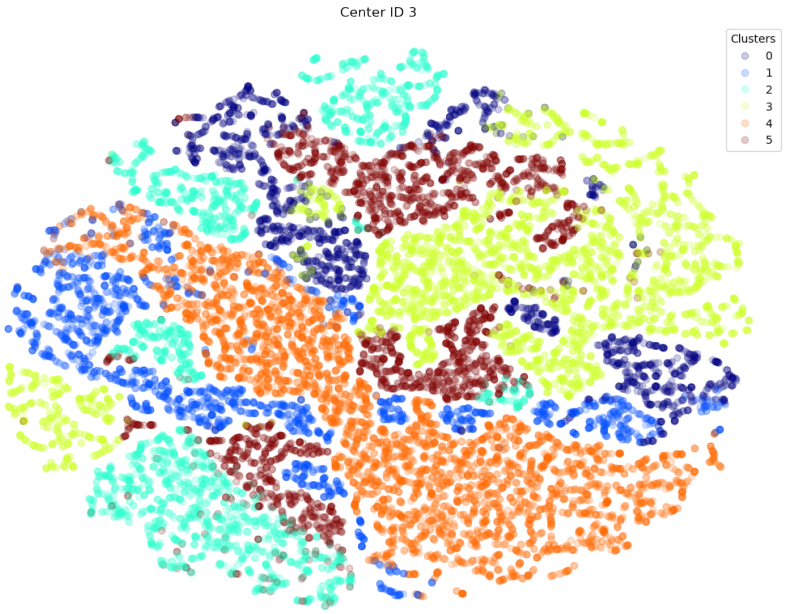



(a) GMM Clusters - Center 2

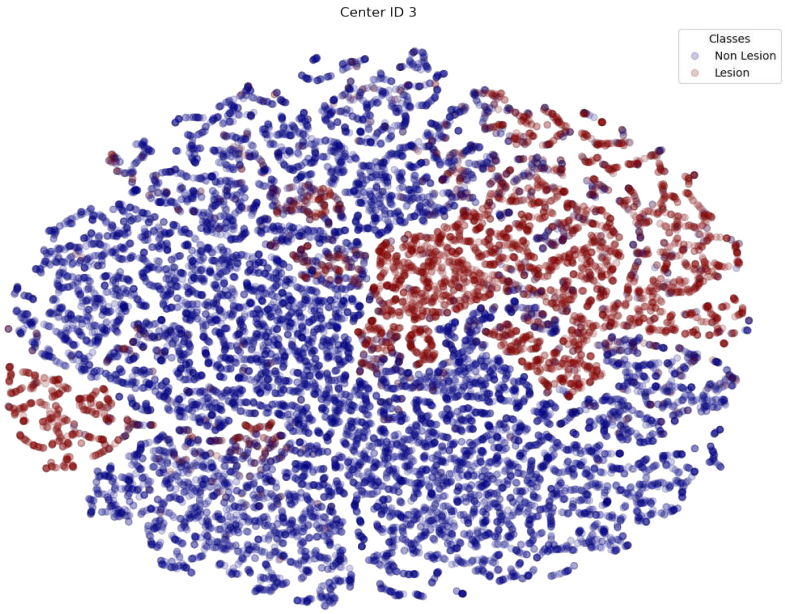



(b) Lesion/Non-Lesion - Center 2

Figure 3.10: t-SNE representation of the GMM clusters and the corresponding lesion/non-lesion classification on a sample of the support patches of medical center 2. Figure 3.10a shows the different clusters identified by the GMM algorithm. Figure 3.10b shows in red the patches classified as lesions and in blue the patches classified as non-lesion.

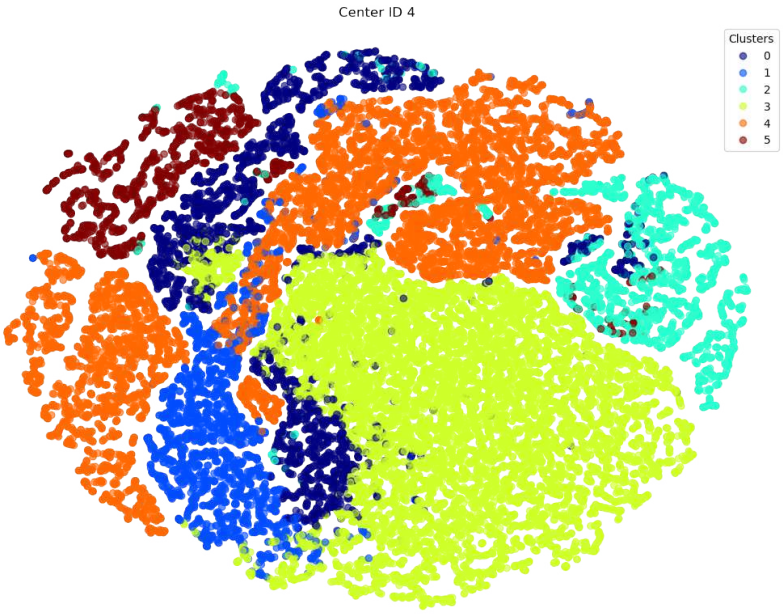



(a) GMM Clusters - Center 3

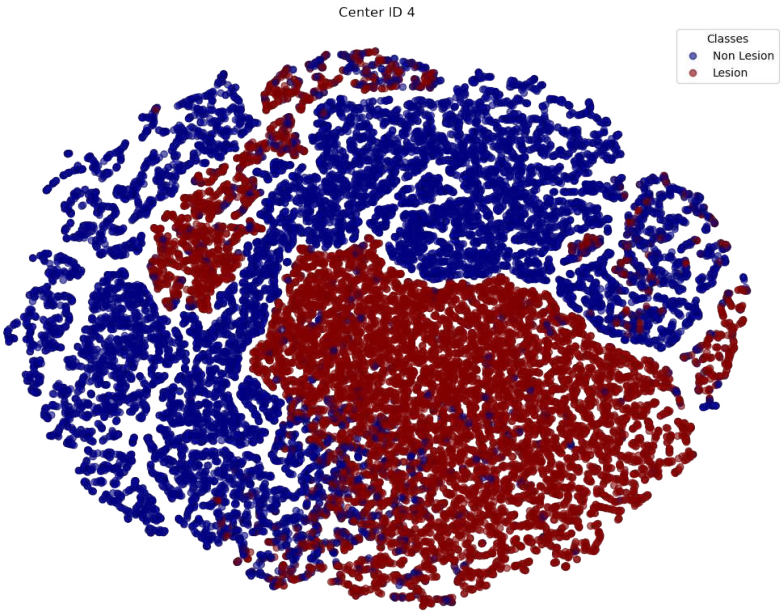



(b) Lesion/Non-Lesion - Center 3

Figure 3.11: t-SNE representation of the GMM clusters and the corresponding lesion/non-lesion classification on a sample of the support patches of medical center 3. Figure 3.11a shows the different clusters identified by the GMM algorithm. Figure 3.11b shows in red the patches classified as lesions and in blue the patches classified as non-lesion.




(a) GMM Clusters - Center 4




(b) Lesion/Non-Lesion - Center 4

Figure 3.12: t-SNE representation of the GMM clusters and the corresponding lesion/non-lesion classification on a sample of the support patches of medical center 4. Figure 3.12a shows the different clusters identified by the GMM algorithm. Figure 3.12b shows in red the patches classified as lesions and in blue the patches classified as non-lesion.

Table 3.8: GMM clusters of the support patches per each medical center. The percentage of lesion and non-lesion patches assigned to each cluster out of the total number of lesion and non-lesion patches is shown in columns $r_{pos}$ and $r_{neg}$ respectively. For each medical center I underlined the clusters that contain the majority of lesion patches. The last column is the estimated probability of lesion patches present in each cluster as computed with equation 3.1.

| **Center ID ($c$)** | **Cluster ID ($g$)** | **Lesion % ($r_{pos}$)** | **Non-Lesion % ($r_{neg}$)** | **$\pi_l(c,g)$ %** |
|---|---|---|---|---|
| 0 | 0 | 17.3 | 14.6 | 51.7 |
| | 1 | 0.1 | 31.3 | 0.3 |
| | 2 | 7.1 | 16.5 | 28.2 |
| | 3 | 0.0 | 2.6 | 0.0 |
| | 4 | 0.9 | 24.8 | 3.2 |
| | 5 | 74.6 | 10.3 | 86.3 |
| 1 | 0 | 0.8 | 14.3 | 4.8 |
| | 1 | 23.8 | 36.8 | 38.3 |
| | 2 | 0.5 | 0.8 | 16.7 |
| | 3 | 0.6 | 31.8 | 1.9 |
| | 4 | 74.2 | 15.1 | 81.6 |
| | 5 | 0.0 | 1.1 | 0.0 |
| 2 | 0 | 0.0 | 59.0 | 0.0 |
| | 1 | 79.8 | 0.8 | 97.1 |
| | 2 | 13.0 | 8.3 | 56.8 |
| | 3 | 4.4 | 9.8 | 28.1 |
| | 4 | 0.2 | 19.6 | 0.8 |
| | 5 | 2.7 | 2.5 | 39.7 |
| 3 | 0 | 0.7 | 10.4 | 35.2 |
| | 1 | 0.8 | 16.6 | 4.3 |
| | 2 | 2.9 | 18.5 | 12.6 |
| | 3 | 78.1 | 2.3 | 95.3 |
| | 4 | 0.2 | 35.8 | 0.7 |
| | 5 | 11.4 | 16.3 | 38.8 |
| 4 | 0 | 15.5 | 6.5 | 65.7 |
| | 1 | 0.1 | 16.2 | 2.8 |
| | 2 | 0.3 | 12.1 | 19.1 |
| | 3 | 80.1 | 3.5 | 94.1 |
| | 4 | 0.0 | 51.4 | 0.0 |
| | 5 | 0.1 | 10.2 | 0.0 |

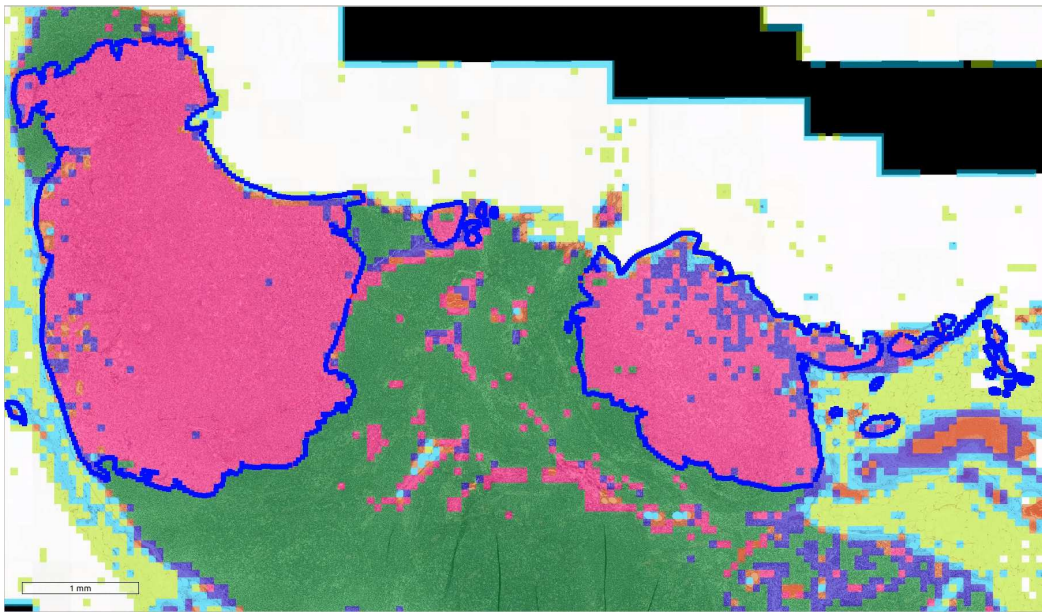



Figure 3.13: Visual example of the strong agreement between GMM clusters and lesions. This example shows a region of an annotated slide (node 4) of CAMELYON17's patient 75 belonging to medical center 3. The ground truth annotation delimiting the macro-metastases is the blue line. Each patch is marked with a different color based on the GMM cluster it is associated to. We can see that the macro-metastases are almost exclusively associated with a single GMM cluster.

This strong association is also confirmed by looking at single WSIs, the one shown in Figure 3.13, for example, is a region with two macro-metastases of a WSI (patient 75) of medical center 3.

### 3.3.3 Extraction of prototypes support patches

Once GMM clusters have been identified, for each medical center prototype patches are extracted for each cluster. These prototype patches will be associated to each query patch as their support. I extract prototypes for computational reasons, because it would be impractical to scan through all possible support patches of each cluster to identify the support patches closest, in the latent space, to each query patch.

Prototypes are extracted for each GMM cluster by running a k-means clustering on all the support patches assigned to each GMM cluster. The number of clusters for the k-means algorithm is the number of support patches assigned to a GMM cluster divided by a constant number, called

*microcluster dimension*, set as a hyperparameter during training. With this approach, I aim at having each k-means cluster associated with roughly the same number of elements[2]. I extract separately prototypes of patches classified as lesion and non-lesion, as such for each GMM clusters two pools of prototypes will be available:

- the pool of lesion prototypes; it might eventually be empty if the GMM cluster does not contain lesion patches;
- the pool of non-lesion prototypes; it could be empty if the GMM cluster contains lesion patches only.

This separation enables the query patch to support prototypes association performed during training as described in Section 3.4.

## 3.4 Support shots selection and FSS training

During training, the segmentation branch takes as input mini-batch of pair of query patches and their corresponding annotation masks $(I_q^c, L_q^c)$, where $I_q^c$ is a generic query patch from center $c$ and $L_q^c$ is its corresponding annotation mask ground truth necessary for computing the loss. In my case $c \in \{0, 1, 2\}$. The conditioning branch, instead, receives as input, for each query set patch, $k$ patches, a.k.a. *k-shots*, chosen among the support prototypes of the same medical center the query patch belongs to. The prototypes are the ones described in Subsection 3.3.3 and are selected with the following policy:

- each query patch is associated with its corresponding GMM cluster $g$;
- all Euclidean distances in the latent PCA space between the query patch and all the support prototypes belonging to the same center and to the same cluster are computed;
- the lesion probability value of the GMM cluster, $\pi_l(c, g)$ (last column of Table 3.8) sets the type and ordering of prototype support patches associated with the query and fed to the conditioning branch, in particular

[2]This assumption holds true exactly if the support patches of a GMM cluster have, in the latent space, a uniform distribution. If the distribution is not uniform, denser areas will be over-represented with more prototypes and sparser areas will be under-represented.

  - the lesion probability $\pi_l(c, g)$ (last column of Table 3.8) is multiplied by $2^k$ and the result is binarized with $k$ binary digits. From each binary digit of the representation I extract the class the support prototype must belong to. If the digit is 0 the prototype is extracted from the pool of non-lesion prototypes, vice versa, if the digit is 1 the prototype is extracted the from the pool of lesion prototypes[3]
  - once the pool of prototypes to extract the prototype from has been chosen the support prototype which is closest, in the PCA latent space, to the query patch is chosen as support shot;
  - the process continues until we have completed all support shots; at each iteration, if the prototype is selected from a pool which had already been used in a previous iteration, the next closest prototype of that pool is used as support shot;

- the chosen support prototypes are concatenated together;

- the result of the concatenation is fed to the conditioning branch, such as the resulting tensor received as input by the conditioning branch has shape $3k \times 128 \times 128$.

In summary the input to the conditioning branch are, for each $I_q^c$ belonging to a cluster $g$, $k$ patches $I_s^c(g, l)$ belonging to the same center $c$ and cluster $g$ and being either lesion patches, $l = 1$, or non-lesion patches, $l = 0$. The choice of $l$ is associated with the value of $\pi_l(c, g)$ for the cluster $g$ of center $c$ as described with the policy above. The estimated probability of lesion patches for cluster $g$ of medical center $c$ is an additional input the conditioning branch receives for each query patch $I_q^c$ belonging to cluster $g$. This value is used to compute an *auxiliary* loss for "pretext tasks" as explained in Subsection 3.4.1.

In the theoretical view of Section 2.3, each support set identified with the policy above and all the query patches that have identical support set form a task $\mathcal{T}$ from the distribution of possible tasks $p(\mathcal{T})$. The conditioning branch is represented by the parametric function $\boldsymbol{g}_\omega$ of equation 2.2, while the segmentation branch is parameterized by $\theta$.

[3]For example if $\pi_l(c, g)$ is 0.8, and $k$ is 2, '11' is the binary representation of the integer part of $0.8 * 4 = 3.2$, therefore both support shots are chosen among the pool of lesion prototypes. As another examples if $p_l(c, g)$ is 0.49, the associated binary with 2 shots and 2 digits of the integer part of $0.48 * 4 = 1.92$ is '01'. In this case the first support shot is extracted from non-lesion pool of prototypes and the second from the lesion pool.

### 3.4.1 Training loss and regularizer

For training I use an Adam Optimizer with the initial learning rate set to 0.001. The loss is a weighted binary cross-entropy loss summed with a auxiliary pretext loss. Formally, given a query pair $(I_q, L_q)$, we can select $k$ support shots $\{I_s^0, \ldots, I_s^{k-1}\}$ and one estimated probability value $\pi_l(I_q)$[4]. The associated training loss is then defined as

$$\mathcal{L}_{train}(I_q) := \mathcal{L}_{\texttt{wBCE}}\left(M_s(I_q), L_q\right) + w \cdot \mathcal{L}_{\texttt{BCE}}\left(\left(\sigma(k^{-1}\sum_{j=0}^{k-1} M_c(I_s^j)\right), \pi_l(I_q)\right) \quad (3.2)$$

The first term of the sum is the ordinary Binary Cross-Entropy (BCE) loss between the predicted masks $M_s(I_q)$ for the query patch $I_q$ and its corresponding ground truth masks $L_q$. The second term is instead a auxiliary pretext loss which is again a BCE loss but this time for the pretext task of estimating the probability of lesion patches $\pi_l(I_q)$ from the output of sigma function $\sigma$, applied to the average activation of the sum of the masks $M_c(I_s^j)$ predicted by the conditioning branch for all $k$ shots. The weight $w$ is a weighting hyperparameter.

Through initial experiments the auxiliary pretext loss proved useful to regularize the training of the network. It is a valid alternative to supplying the segmentation masks of the support patches as an input to the network, a solution that would conflict with the use of skip connections in the segmentation branch as already witnessed and explained by Guha Roy et al. (2020) and as I have also verified through initial experiments conducted to prune the FSS possible architectures.

The initial weight for the lesion class is 4.0 as the lesion/non-lesion patch ratio in the query set is 16.4%, i.e. we have around 1 lesion patch every 4 normal patches (see Table 3.7). Early stopping is used to avoid overfitting: I stop the training if the validation loss does not improve for 3 consecutive epochs. I trained on slides from medical centers 0, 1 and 2 and I used WSI of patient 75 of medical center 3 to choose the best microcluster dimension hyperparameter. The code is implemented in PyTorch 1.6 (Paszke et al., 2019) with the use of the PyTorch Lightning 0.9 framework (Falcon, 2019). All trainings and experiments were conducted on a workstation equipped with a NVIDIA RTX™ 2080 Ti GPU.

[4]For every $I_q$ we have a uniquely identified center $c$ and cluster $g$, therefore we can write $\pi_l(I_q)$ as a shorthand for $\pi_l(\mathbf{c}(I_q), \mathbf{g}(I_q))$ where $\mathbf{c}(\dot{})$ and $\mathbf{g}(\dot{})$ would be maps from $I_q$ to its corresponding center $c$ and cluster $g$.

During inference, the same policy is applied to associate the patches of any input WSI to their corresponding support patches. The support patches identified with the combination of the autoencoder and the two levels of clustering contain information help the segmentation branch complete its task as I show with the experiments described in Chapter 4.

Training is done in mini-batches, it is not episodic in the sense that I do not sample tasks first, I sample query patches and for each query patch its associated support set. Although this approach might appear to go against the established paradigm of meta-learning to use episodic training, because the task involved the classification of a single class, episodic learning might have caused an early collapse of the network toward a local minima. This view is supported by recent research conducted by Laenen and Bertinetto (2020) who found evidence that episodic training could be detrimental to the performance of widely used FSL architectures and, under specific training conditions, demonstrated the advantage of regular mini-batch training.

## 3.5 Summary

In this chapter, I discussed:

- the FSS architecture with details on the function of the blocks used in the network;
- the differences between the FSS used for this research and the reference architecture used by Guha Roy et al. (2020);
- the composition of the training dataset and the splitting of the available WSIs among query and support sets for few-shot learning;
- the unsupervised methods used for the selection of the support set patches;
- the policy to associate the query set patches with their corresponding support patches at training and inference time.

Chapter 4

# Results

In this chapter I review the results of the experiments conducted with the FSS network described in Chapter 3. To understand if and for which lesion types and datasets the conditioning branch improves the output of the segmentation branch, I compare the FSS results against a baseline U-Net architecture. The U-Net is architecturally identical to the segmentation branch of the FSS network, it is therefore equivalent to the FSS with the conditioning branch and all its connections removed. The experiments show that the conditioning extracted from the support set is useful to improve the performances of the U-Net on the WSIs of medical center 4 that uses a slide digital scanner not used to collect the WSIs in the training set (see Section 3.2).

## 4.1 Baseline comparison

I compared the FSS network against the baseline U-Net like architecture of the segmentation branch only, U-Net for short in the following. The U-Net has been trained with the same Adam optimizer, binary cross-entropy loss, and relevant hyperparameters, such as the initial learning rate, of the FSS. The dataset used for training of the U-Net is the union of the support and query sets for centers 0, 1 and 2. Adding the support set of these three centers to the training set of the U-Net increases the dataset size by approximately 40%. I evaluated two variants of the U-Net: with and without 'spatial and channel SE' *csSE* blocks (see Subsection 3.1.2).

### 4.1.1 Evaluation Metrics and test set

A constraint of the chosen method, which relies on the latent representation of the autoencoder described in Subsection 3.3.1 is that at inference time the WSI needs to be split in separate patches, each $128 \times 128$ pixels wide. As a consequence the evaluation metrics are also patch level rather than pixel level. We discovered at the beginning of the research that, for the task at hand, such patch level metrics were analogous, and more efficient to compute, than pixel level metrics, such as for example the Intersection over Union. Patch level metrics also have the advantage that are less sensitive to errors and discrepancies present in the segmentation boundaries of the CAMELYON16 and CAMELYON17 datasets.

For evaluation purposes I classify a patch as a lesion patch if at least one pixel is marked as lesion in its central area, $64 \times 64$ pixels wide. Analogously a patch is classified as a predicted lesion patch if at least one pixel in its central $64 \times 64$ pixels wide region has a probability higher than a certain threshold to be a lesion pixel: in practice, for each patch, the probability that the patch has of being a lesion patch (lesion probability for short) equals the minimum probability any of its pixels in its central region have of being classified as lesion pixels.

Because of its wide adoption in the clinical practice of evaluating the performance of diagnostic tests for binary predictors (Mandrekar, 2010), I chose to use the Receiver Operating Characteristic (ROC) curve and the corresponding Area Under the Curve (AUC) metric to assess and compare the performance of the FSS against the U-Net. Rephrasing DeLong et al. (1988), the AUC represents the probability that, when a randomly selected lesion patch and a randomly selected non-lesion patch are selected, their predicted lesion probabilities are in the correct order, i.e. the predicted lesion probability of a true lesion patch is higher than the predicted lesion probability of a true non-lesion patch. The comparison of the actual class of each patch, assigned starting from the pathologists annotations, and the predicted class, assigned from the predicted pixels semantic classification, allows to compute the ROC curve of the inference on each WSI. To evaluate the performance of different networks and network configurations on each separate WSI, I summarize the ROC with the AUC metric; all AUC metrics and ROC curves have been computed and displayed with the support of the `pROC` R package (Robin et al., 2011).

For evaluation purposes I relied on 6 WSIs from medical centers 3 and 4 of the CAMELYON17 dataset, 3 WSIs for each medical center (see Section 3.2 for details). The 6 WSIs were selected not to be part of the support set of centers 3 and 4 and such that the annotated slides contained all three

different types of metastases present in the dataset: ITCs, micro-metastases and macro-metastases. In the following subsections I identify each WSI with the notation 'patient ID/node ID', where the 'patient ID' is a unique patient identifier in the CAMELYON17 dataset, and the 'node ID' is a unique identifier of the slides collected for each patient.

### 4.1.2 Segmentation branch inference results with no conditioning

Tables 4.1 and 4.2 report the inference results on the WSIs of medical centers 3 and 4 by the 'classically' trained U-Net. Along with the AUC mean values I also include their 95% confidence interval (CI). For computing the CI value I use the `ci.auc` function of the `pROC` package: the AUC variance is computed with the method described in DeLong et al. (1988) and implemented by Sun and Xu (2014), the CI is then computed with the standard R quantile function `qnorm`.

Higher than 0.97 AUC scores are obtained on all WSIs except for the slide with micro-metastases of patient 67 (center 3) and a slide containing ITCs of patient 89 (center 4). There is no significant difference in performance between the two network configurations as most mean AUCs of a network configuration are well within the range of possible values of the other network configuration. The only two exceptions are two WSIs belonging to center 4, a slide of patient 88 and a slide of patient 99. For patient 99 the U-Net with no *csSE* blocks achieves a marginally better AUC score, for patient 88 the reverse is true with the configuration with *csSE* blocks having a better AUC.

Table 4.1: AUC with 95% CI of U-Net with and without *csSE* blocks for center 3 WSIs.

| | **AUC** | | |
|---|---|---|---|
| **csSE** | **72/0** | **67/4** | **75/4** |
| **block** | **ITC** | **micro** | **macro** |
| No | $0.981 \pm 0.011$ | $0.533 \pm 0.077$ | $0.982 \pm 0.003$ |
| Yes | $0.977 \pm 0.014$ | $0.604 \pm 0.087$ | $0.982 \pm 0.003$ |

In summary there is no clear advantage of one network configuration over the other, and for the FSS comparison both configurations, with and without *csSE* blocks, is tested in the following sections.

Table 4.2: AUC with 95% CI of U-Net with and without *csSE* blocks for center 4 WSIs.

| | AUC | | |
|---|---|---|---|
| **csSE** | **89/3** | **88/1** | **99/4** |
| **block** | **ITC** | **micro** | **macro** |
| No | $0.577 \pm 0.068$ | $0.943 \pm 0.008$ | $0.989 \pm 0.002$ |
| Yes | $0.589 \pm 0.070$ | $0.974 \pm 0.007$ | $0.983 \pm 0.002$ |

### 4.1.3 Branches connection via channel weights

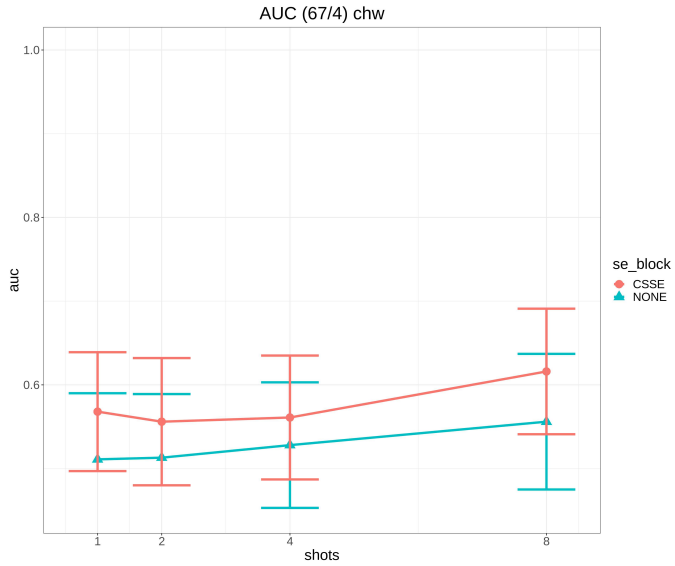
In this section, I review the AUC scores for the WSIs of center 3 and 4 when the two branches are connected via 'channel weights', i.e. with the *MX* blocks in the configuration shown in Figure 3.3b. All trainings and experiments were conducted with the same hyperparameters:

- initial learning rate 0.001;
- positive class weight 4.0, i.e. the loss of the positive/lesion class is weighted 4 times more than the loss of the negative/non-lesion class;
- pretext task loss weight 0.2, i.e. the loss associated with the conditioning branch that compares the predicted and the actual probability estimate of lesion patches (see Subsection 3.4.1) is weighted before being added to the loss of the segmentation branch;
- microcluster dimension 20 (see Subsection 3.3.3);
- 75% of the patches in the query set are used as training and the rest are left for validation;
- early stopping, the training stops if there is no improvement in the validation loss for 3 consecutive epochs.

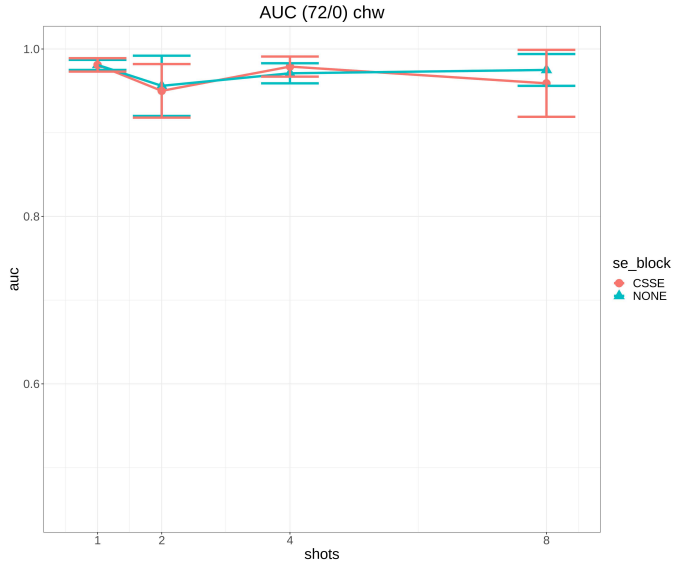
Analogously to what I did for the U-Net, I tested with and without *csSE* blocks in the encoder and decoder blocks of both branches. For each network configuration I tested at different number of support shots: 1, 2, 4 and 8 shots.

Figures 4.1 and 4.2 summarize the results per each WSI of the two medical centers. The numeric AUC scores with their CI at 95% are available in Section A.1 of appendix A.

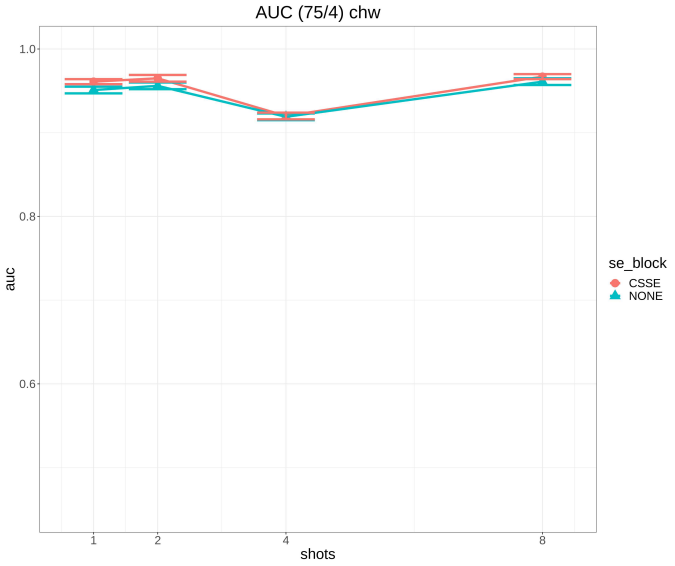
For medical center 3, we see from Figure 4.1 that for the slides with ITCs (patient 72, Figure 4.1b) and micro-metastases (patient 67, Figure 4.1a) the




(a) Patient 67, node 4 (micro-metastases)




(b) Patient 72, node 0 (ITCs)




(c) Patient 75, node 4 (macro-metastases)

Figure 4.1: AUCs of two branch network with channel weights connections on Center 3 WSIs. 'CSSE' labeled points use *csSE* blocks, the others do not.

two network configurations are equivalent at any number of shots (the mean AUC scores of both network configurations are well withing the range of possible AUC values of the other network configuration). For the WSI of patient 75, containing macro-metastatses, the configuration with *csSE* blocks has a slight advantage at 1 and 2 shots (see Figure 4.1c).

For medical center 4, we see from Figure 4.2 that the configuration with no *csSE* blocks has a slight advantage for the WSI with micro-metastates (patient 88, Figure 4.2a) at 2 and 4 shots, while instead for the WSI with macro-metastases (patient 99, Figure 4.2b) the configuration with *csSE* blocks has a consistent better performance at any shots, confirming the behaviour already seen for the WSI with macro-metastases in center 3 (patient 75, Figure 4.1c).

**FSS with 'channel weights' vs U-Net**

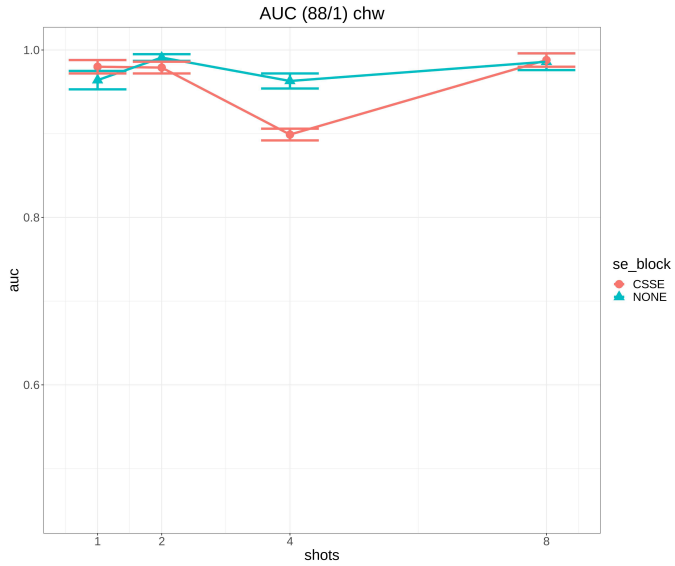
In Tables 4.3 and 4.4 I show the percentage difference between the AUCs of the FSS configurations without and with the *csSE* blocks and the equivalent, i.e. without and with *csSE* blocks, configurations of the baseline U-Net.

In Tables 4.3 and 4.4 I also compare the AUCs of the two correlated ROC curves, one from the prediction of the FSS and the other from the prediction of the corresponding U-Net, with the `roc.test` function of the `pROC` R package. This function implements the test described in DeLong et al. (1988) to assess whether the AUCs have a statistically significant difference or not: for each two AUCs comparison, the test outputs a *p-value* which I record in the tables next to the percentage difference using the following significance codes (analogous to the significance codes used in regression output of some other R packages):

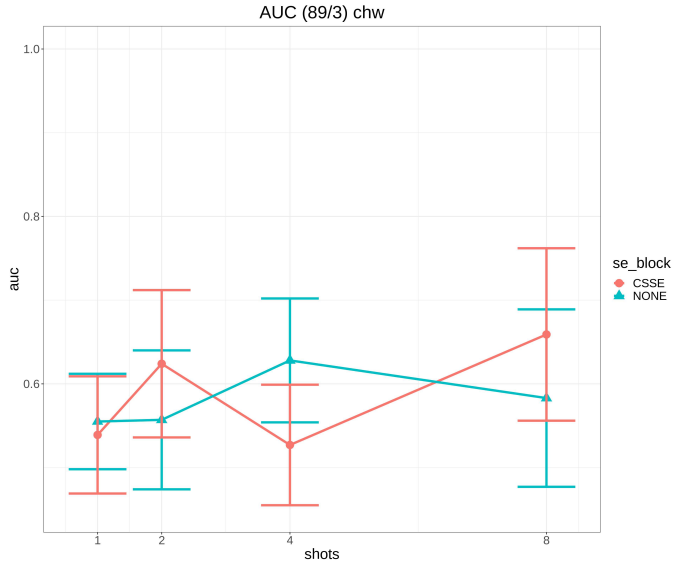
- *** - $0 \leq$ *p-value* $< 0.001$
- ** - $0.001 \leq$ *p-value* $< 0.01$
- * - $0.01 \leq$ *p-value* $< 0.05$
- . - $0.05 \leq$ *p-value* $< 0.1$

For *p-value*s larger than 0.1 I do not include any code next to the percentage difference of the AUC values.

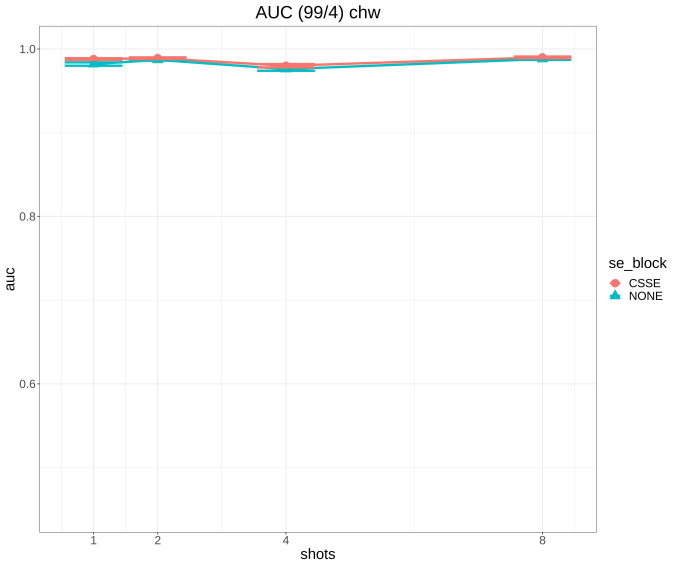
Table 4.3 shows that with no *csSE* blocks, a statistically significant improvement in the AUC score is obtained, for any number of shots, for the WSI with micro-metastases of center 4 (patient 88). On all other WSIs the conditioning does not significantly improve the AUC scores and for both WSIs with macro-metastases, patient 75 of medical center 3 and patient 99




(a) Patient 88, node 1 (micro-metastases)




(b) Patient 89, node 3 (ITCs)




(c) Patient 99, node 4 (macro-metastases)

Figure 4.2: AUCs of two branch network with channel weights connections on Center 4 WSIs. 'CSSE' labeled points use *csSE* blocks, the others do not.

Table 4.3: Percentage difference of AUC of FSS with no *csSE* blocks and with channel weights connections between branches w.r.t. AUC of baseline U-Net for Center 3 and 4 WSIs.

| | AUC variation | | | | | |
|---|---|---|---|---|---|---|
| | **72/0** | **67/4** | **75/4** | **89/3** | **88/1** | **99/4** |
| **Shots** | **ITC** | **micro** | **macro** | **ITC** | **micro** | **macro** |
| 1 | 0.0% | -4.1% | -3.2% | -3.8% | 2.2% ** | -0.7% *** |
| 2 | -2.5% | -3.8% | -2.6% *** | -3.5% | 5.1% *** | -0.2% . |
| 4 | -1.0% | -0.9% | -6.4% *** | 8.8% | 2.1% ** | -1.3% *** |
| 8 | -0.6% | 4.3% | -2.1% *** | 1.0% | 4.6% *** | -0.1% |

of medical center 4, it marginally degrades the AUC performance. Overall the best performances with no *csSE* blocks is achieved at 8 shots.

In Table 4.4 I show instead the comparison between the U-Net and the FSS with *csSE* blocks. In this case a statistically significant drop of AUC score is present for the WSI with macro-metastases of medical center 3 (patient 75, node 4), but we have instead a statistically significant improvement for the the WSI with macro-metastases of medical center 4 (patient 99, node 4) at all shots except 4. The only other statistically significant difference happens for shots 4 and 8 on the WSI with ITCs of medical center 4 (patient 88, node 1). Again the best overall performance is achieved at 8 shots.

Table 4.4: Percentage difference of mean AUC of FSS with *csSE* blocks and with channel weights connections between branches w.r.t. baseline U-Net for Center 3 and 4 WSIs.

| | AUC variation | | | | | |
|---|---|---|---|---|---|---|
| | **72/0** | **67/4** | **75/4** | **89/3** | **88/1** | **99/4** |
| **Shots** | **ITC** | **micro** | **macro** | **ITC** | **micro** | **macro** |
| 1 | 0.4% | -6.0% | -2.1% *** | -8.5% | 0.6% | 0.5% *** |
| 2 | -2.8% | -7.9% | -1.7% *** | 5.9% | 0.5% | 0.6% *** |
| 4 | 0.2% | -7.1% | -6.3% *** | -10.5% | -7.7% *** | -0.3% ** |
| 8 | -1.8% | 2.0% | -1.5% *** | 11.9% | 1.4% ** | 0.7% *** |

In summary the 'channel weights' FSS configuration shows a minor advantage over the plain U-Net architecture on the WSIs of medical center 4, with the most prominent improvement seen at 8 shots and with the use of *csSE* blocks.

### 4.1.4 Branches connection via features concatenation

In this section, I review the AUCs for the WSIs of center 3 and 4 when the two branches are connected via 'features concatenation', i.e. with the *MX* blocks in the configuration shown in Figure 3.3a. All trainings and experiments are conducted with the same hyperparameters (initial learning rate 0.001, positive class weight 4.0, pretext task loss weight 0.2, microcluster dimension 20, 75%/25% training/validation split, early stopping) used for the configuration with 'channel weights'. I tested with and without *csSE* blocks in the encoder and decoder blocks of both branches. For each network configuration I tested at different number of shots: 1, 2, 4 and 8 shots.

Figures 4.3 and 4.4 summarize the results per each WSI of the two medical centers with and without *csSE* blocks. The AUC numerical scores with their CI at 95% are available in Section A.2 of appendix A.

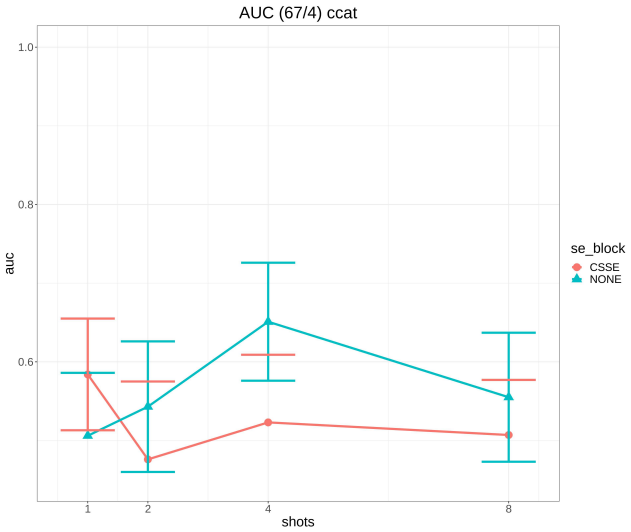
For the WSIs of center 3, a review of the graphs in Figure 4.3 shows that there is no clear advantage of choosing a network configuration with or without *csSE* blocks. For center 4 a slight advantage can be seen using the *csSE* blocks at 2 and 4 shots for the WSI with macro-metastases (patient 99, Figure 4.4c) while for the other two WSIs, with ITCs and micro-metastases, a slight preference for the no *csSE* blocks configuration is seen at 4 and 8 shots (see Figures 4.4b and 4.4a).

#### FSS with 'features concatenation' vs U-Net

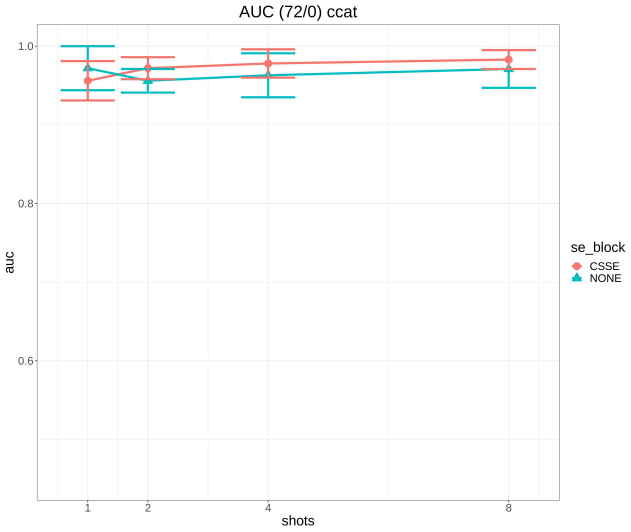
Similarly to what I have already discussed in Subsection 4.1.3, Tables 4.5 and 4.6 show the percentage difference of the AUCs of two different configurations of the FSS, without and with *csSE* blocks, with the AUCs of their equivalent baseline U-Net configurations. The significance codes are as in Subsection 4.1.3.

In Table 4.5 I show that with no *csSE* blocks, a statistically significant improvement in the AUC score is obtained, for all shots greater than 1, for the WSI with micro-metastases of center 4 (patient 88). This is similar to the 'channel weights' configuration. In addition a marginal improvement can also be seen on the WSI with ITCs (patient 89, Figure 4.4b). On all other WSIs the conditioning does not significantly improve the AUC scores and for both WSIs with macro-metastases, patient 75 of medical center 3 and patient 99 of medical center 4, it marginally degrades the AUC performance. This is again similar to what I have obtained for the 'channel weights' configuration. Overall the best performances with no *csSE* blocks is achieved at 4 shots.

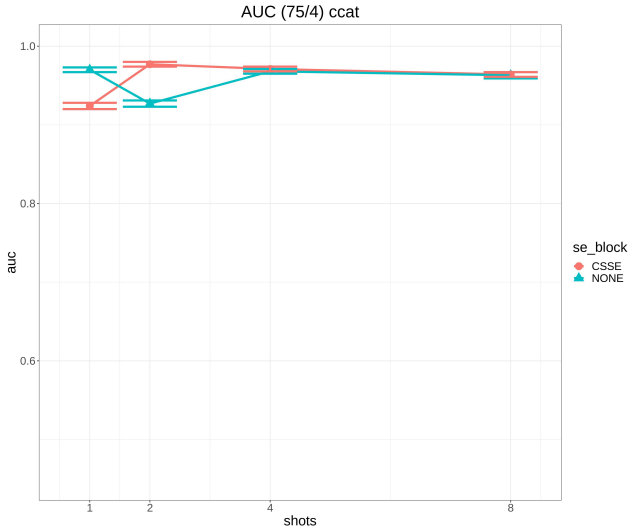
In Table 4.6 I show the comparison between the U-Net and the FSS with *csSE* blocks. Similarly to the 'channel weights' configuration, a statistically




(a) Patient 67, node 4 (micro-metastases)

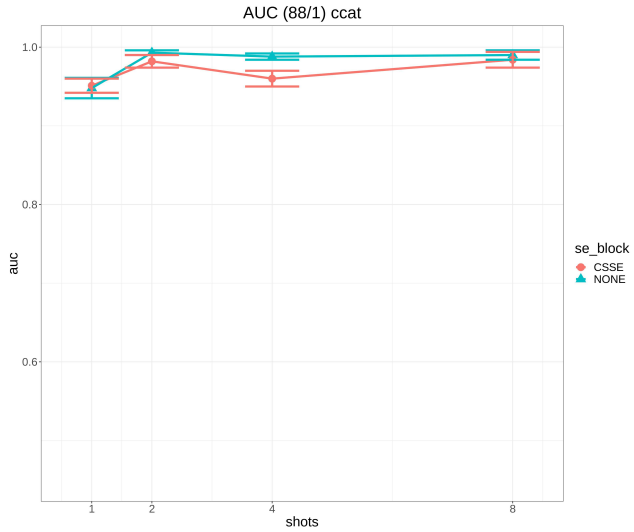



(b) Patient 72, node 0 (ITCs)

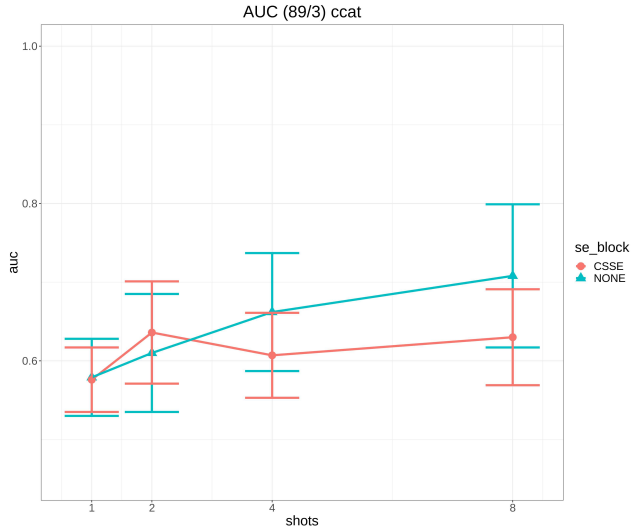



(c) Patient 75, node 4 (macro-metastases)

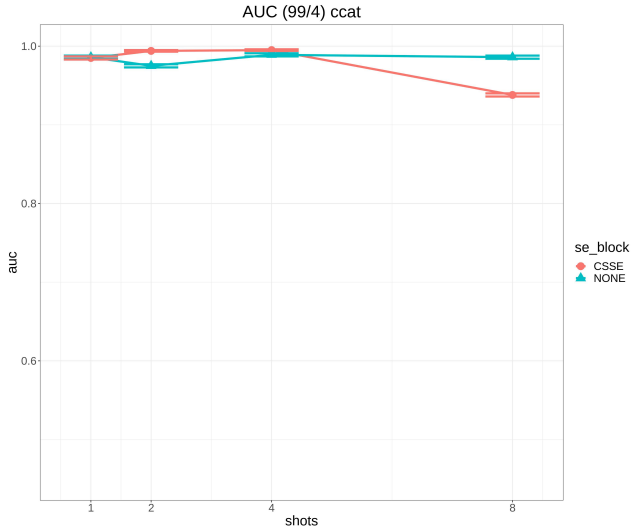
Figure 4.3: AUCs of two branch network with channel weights connections on Center 4 WSIs. 'CSSE' labeled points use *csSE* blocks, the others do not.




(a) Patient 88, node 1 (micro-metastases)




(b) Patient 89, node 3 (ITCs)




(c) Patient 99, node 4 (macro-metastases)

Figure 4.4: AUCs of two branch network with channel weights connections on Center 4 WSIs. 'CSSE' labeled points use *csSE* blocks, the others do not.

Table 4.5: Percentage difference of mean AUC of FSS with no *csSE* blocks and concatenated connections between branches w.r.t. baseline U-Net for Center 3 and 4 WSIs.

| | AUC variation | | | | | |
|---|---|---|---|---|---|---|
| | **72/0** | **67/4** | **75/4** | **89/3** | **88/1** | **99/4** |
| **Shots** | **ITC** | **micro** | **macro** | **ITC** | **micro** | **macro** |
| 1 | -0.9% | -5.1% | -1.2% *** | 0.3% | 0.5% | -0.3% * |
| 2 | -2.5% ** | 1.9% | -5.6% *** | 5.7% | 5.3% *** | -1.4% *** |
| 4 | -1.8% | 22.1% . | -1.4% *** | 14.7% . | 4.8% *** | 0.0% |
| 8 | -1.0% | 4.1% | -1.9% *** | 22.7% * | 5.0% *** | -0.3% * |

Table 4.6: Percentage difference of mean AUC of FSS with *csSE* blocks and concatenated connections between branches w.r.t. baseline U-Net for Center 3 and 4 WSIs.

| | AUC variation | | | | | |
|---|---|---|---|---|---|---|
| | **72/0** | **67/4** | **75/4** | **89/3** | **88/1** | **99/4** |
| **Shots** | **ITC** | **micro** | **macro** | **ITC** | **micro** | **macro** |
| 1 | -2.1% | -3.3% | -5.9% *** | -2.2% | -2.4% *** | 0.2% |
| 2 | -0.5% | -21.0% . | -0.5% * | 8.0% | 0.8% . | 1.1% *** |
| 4 | 0.1% | -13.4% | -1.1% *** | 3.1% | -1.4% * | 1.2% *** |
| 8 | 0.6% | -16.1% . | -1.8% *** | 7.0% | 1.0% . | -4.6% *** |

significant drop of AUC score is present for the WSI with macro-metastaes of medical center 3 (patient 75, node 4), but we have instead a statistically significant improvement for the the WSI with macro-metastases of medical center 4 (patient 99, node 4) at 2 and 4 shots. At 2 and 8 shots a marginal improvement is shown by the FSS again on the other two WSIs of medical center 4, while, instead, the performance for the WSIs of center 3 is generally worse compared to the baseline U-Net.

In summary a pattern emerge, especially for the configuration with no *csSE* blocks, where minor improvements can be seen with respect to the U-Net on the WSIs of medical center 4 and instead the performance drops or remains unchanged on the WSIs of center 3.

An example of the predictions on macro-metastases of the FSS with no *csSE* blocks and 'features concatenation' is shown in Figures 4.5 and 4.6a where the prediction heat-maps are superimposed to the original WSIs. For all figures the heat-map scale is transparent for prediction probabilities below 0.75 and it is colored with hues from green to red for probabilities ranging from 0.75 to 1.0. The predictions by the U-Net for the macro-metastases of medical center 3 are similar, but for the macro-metastasis of medical center 4 the U-Net is less certain about the size and structure of the macro-metastases as shown in Figure 4.6b.

The improvement in performance by the FSS is largely driven by the GMM clusters as can be seen in Figure 4.7 where I superimposed a representation of the patch labels, shown as shade of grey, with the prediction heat-maps of the FSS (see Figure 4.7a) and of the U-Net (see Figure 4.7b). For the FSS the high probability lesion regions almost completely match with one single cluster whose label appears as white and which is an information the FSS receives through the support set and the training policy described in Section 3.4. The U-Net has no such information and so many regions of the 'white' GMM cluster have lower lesion probability predictions.

Figures 4.8 and 4.9 show the heat map of predictions superimposed on the WSIs containing ITCs and micro-metastases of the patients of medical center 3. The same predictions from the U-Net are not shown as the visual performance is similar but with less false positive regions. The case of WSIs with ITCs and micro-metastases of medical center 4 are shown instead in Subsection 4.2.2.

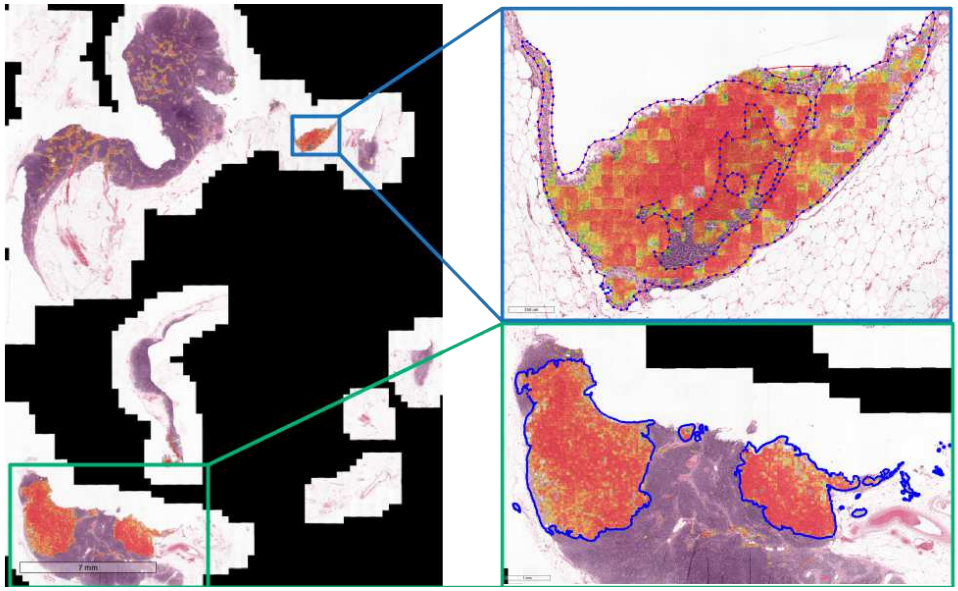
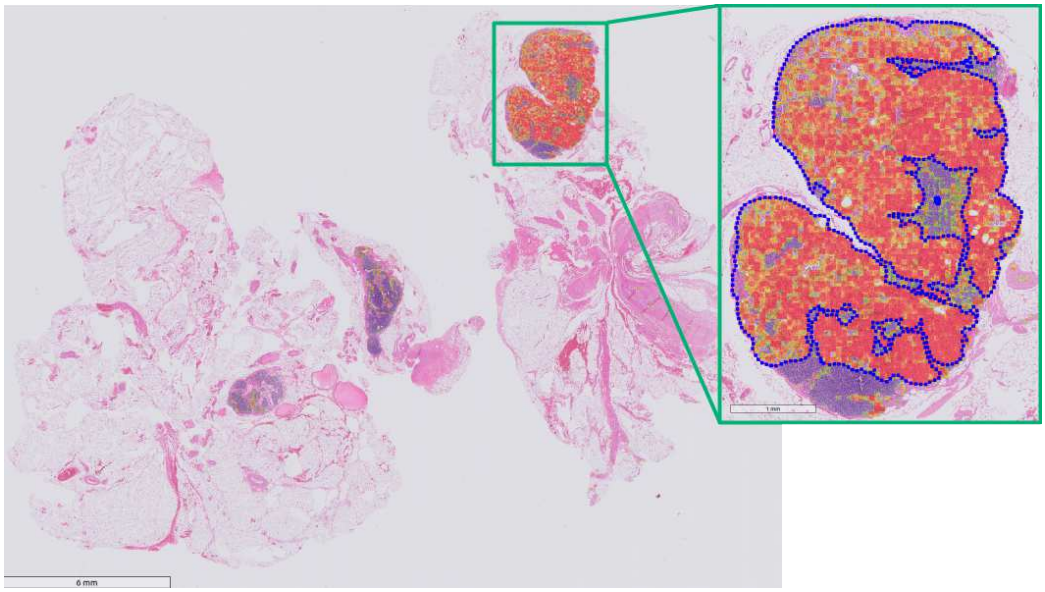


Figure 4.5: Predictions by the FSS with no *csSE* blocks and 'features concatenation' on the entire WSI with three macro-metastases of patient 75 node 4 of medical center 3. Predictions below 0.75 are transparent, the other probabilities have hues from green (0.75) to red (1.0).

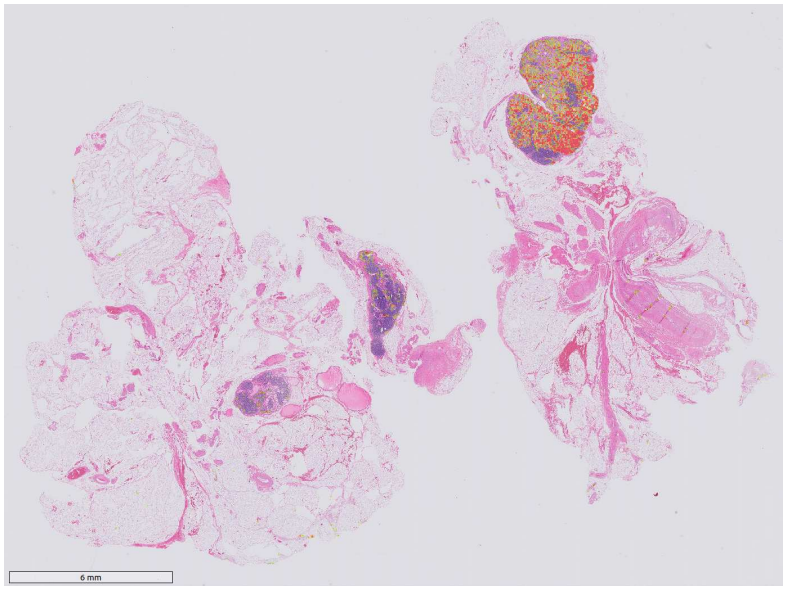
### 4.1.5 Graphical comparison of different two-branch connections vs baseline U-Net

Another way to look at the FSS AUC scores is to add to the same graph, for each *csSE* blocks configuration, the AUC scores of the two two branch connection configurations, either 'channel weights' or 'features concatenation', together with AUC scores of the baseline U-Net architecture with the same *csSE* blocks configuration. This comparison highlights which network configuration improves over the equivalent baseline U-Net architecture and for which WSIs. The results are shown in Figures 4.10, 4.11, 4.12 and 4.13.

From Figure 4.11a we see a significant improvement, at any shots greater than 1, of the FSS, both with 'features concatenation' and 'channel weights', over the regular U-Net with no *csSE* blocks. The U-Net with *csSE* blocks performs better on this same WSI, but the FSS still manages to marginally beat the U-Net at 8 shots with the 'channel weight' configuration as shown in Figure 4.13a. From Figure 4.13c, we see 5 branch connection/shots combinations that easily beat the performance of the U-Net on the WSI with macro-metastases of patient 99 belonging to center 4. Again a pattern emerges that the FSS appears to bring an improvement, under certain architectural combinations, on the WSIs of medical center 4.

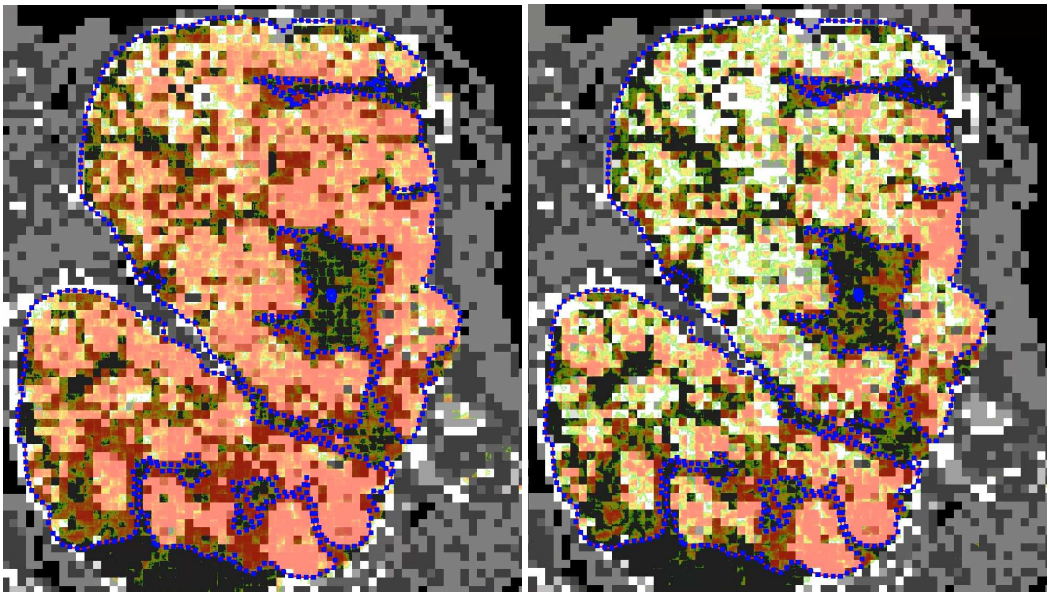


(a) FSS



(b) U-Net

Figure 4.6: A micro-metastasis in patient 99 of medical center 4 as detected by the FSS with no *csSE* blocks and 'features concatenation' and by the U-Net with no *csSE* blocks. Predictions below 0.75 are transparent, the other probabilities have hues from green (0.75) to red (1.0).



(a) FSS (b) U-Net

Figure 4.7: Heat-map of the predictions by the FSS with no *csSE* blocks and 'features concatenation' and by the U-Net superimposed to the GMM cluster labels (shown as shade of grey) for the macro-metastases of patient 99 node 4 in medical center 4. For the FSS, the high probability lesion regions in red almost completely match the cluster with the white background label. The red line with blue dots is the pathologist original segmentation.

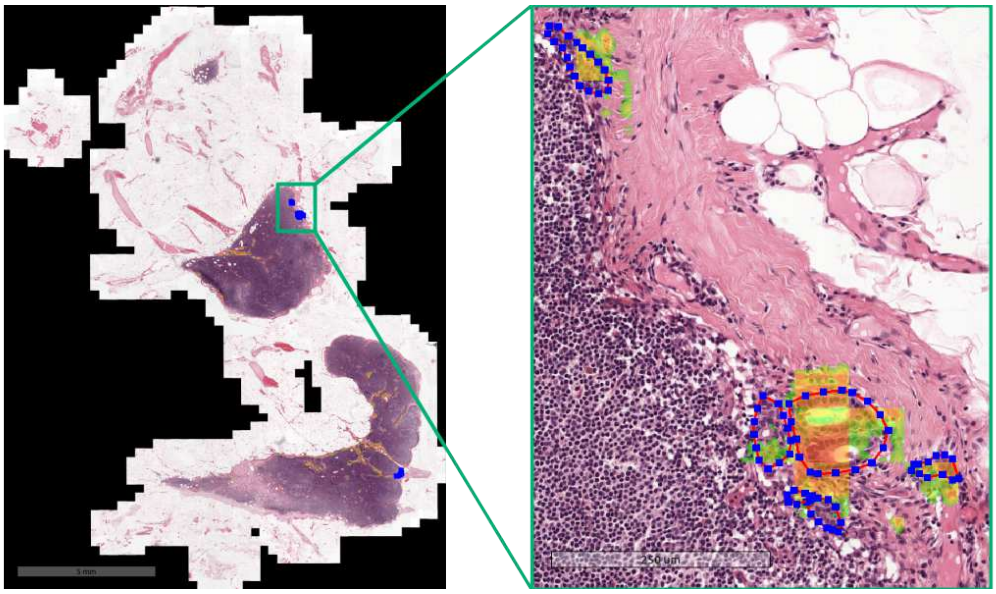


Figure 4.8: Predictions by the FSS with no *csSE* blocks and 'features concatenation' on the entire WSI with ITCs of patient 72 node 0 of medical center 3.

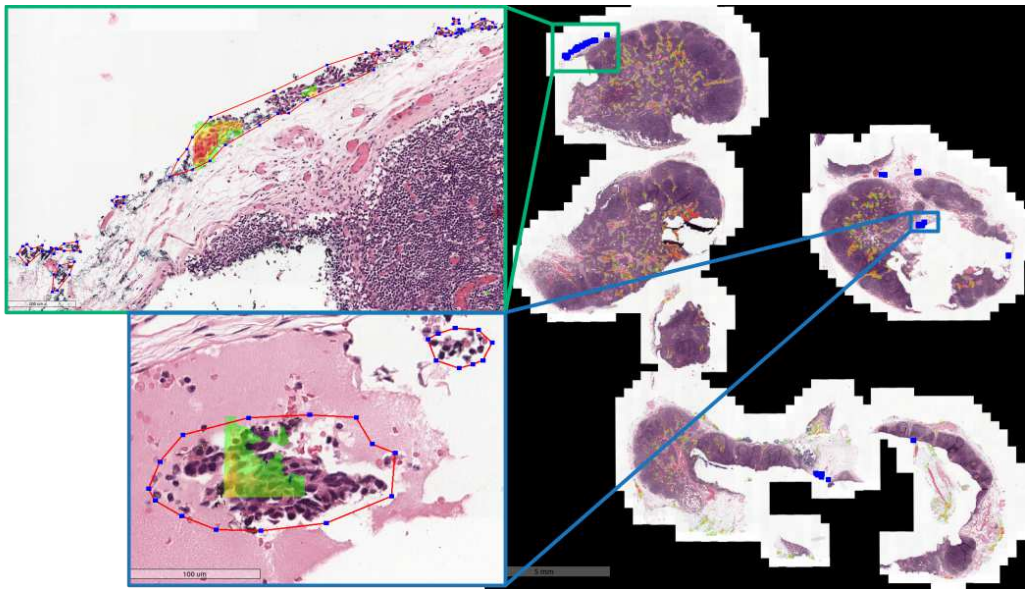
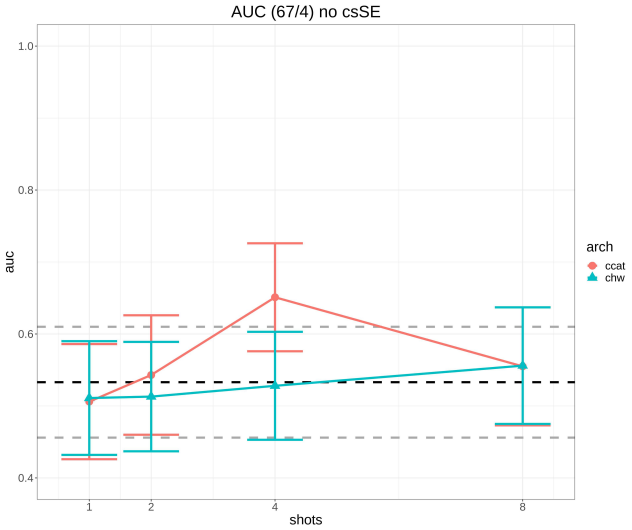
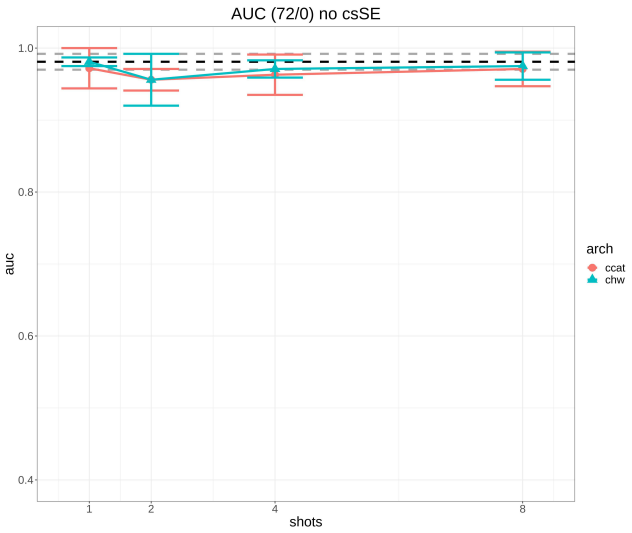


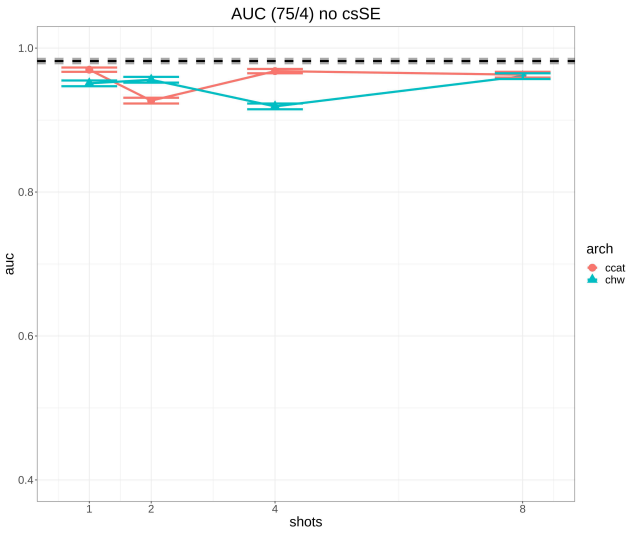
Figure 4.9: Predictions by the FSS with no *csSE* blocks and 'features concatenation' on the entire WSI with micro-metastases of patient 67 node 4 of medical center 3.




(a) Patient 67, node 4 (micro-metastases)

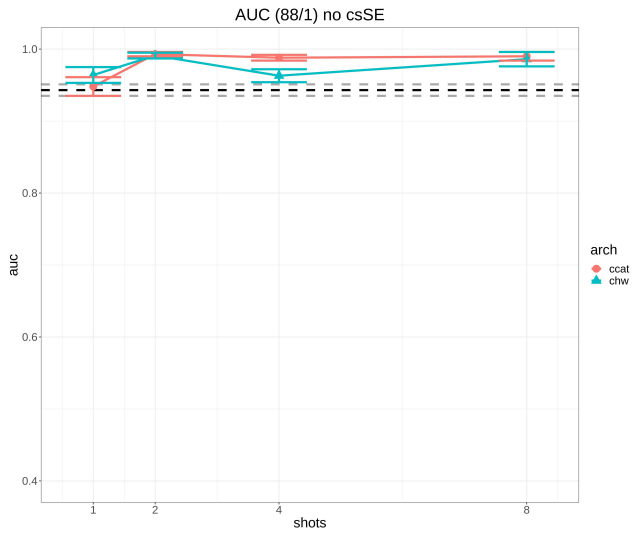



(b) Patient 72, node 0 (ITCs)

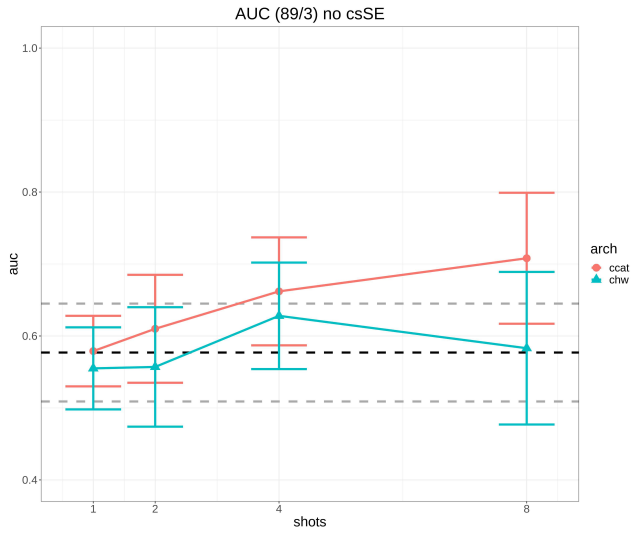



(c) Patient 75, node 4 (macro-metastases)

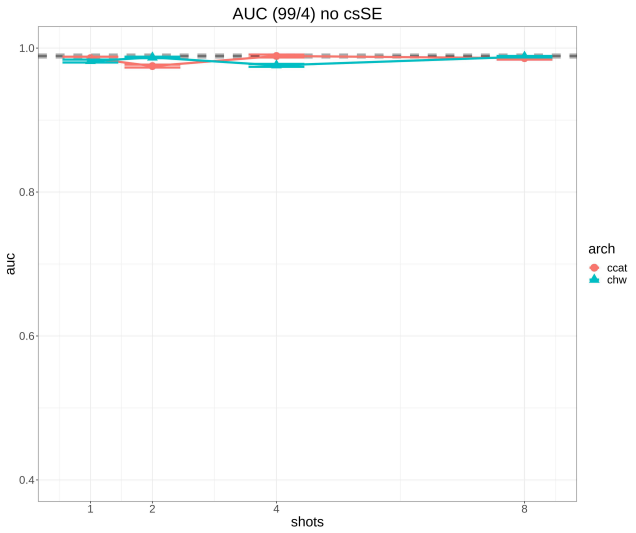
Figure 4.10: AUCs of two branch network without *csSE* blocks on Center 3 WSIs for two network configurations: 'ccat' points use 'concatenated features' connections, the others 'channel weights'.




(a) Patient 88, node 1 (micro-metastases)

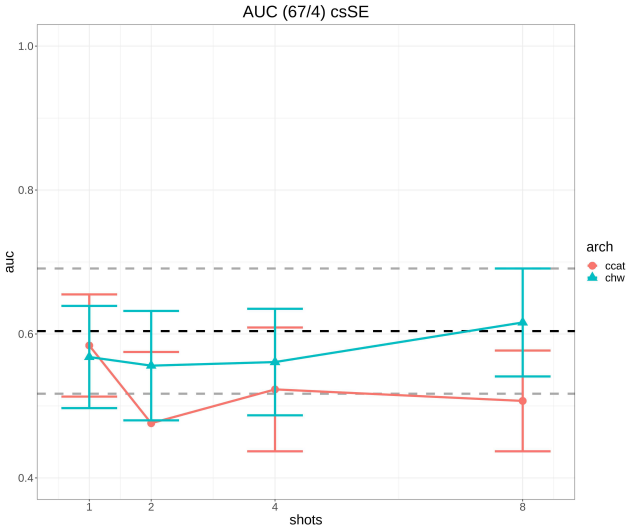



(b) Patient 89, node 3 (ITCs)

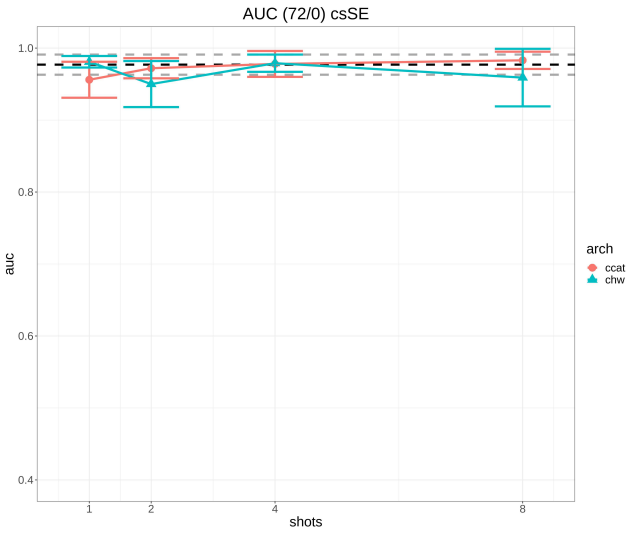



(c) Patient 99, node 4 (macro-metastases)

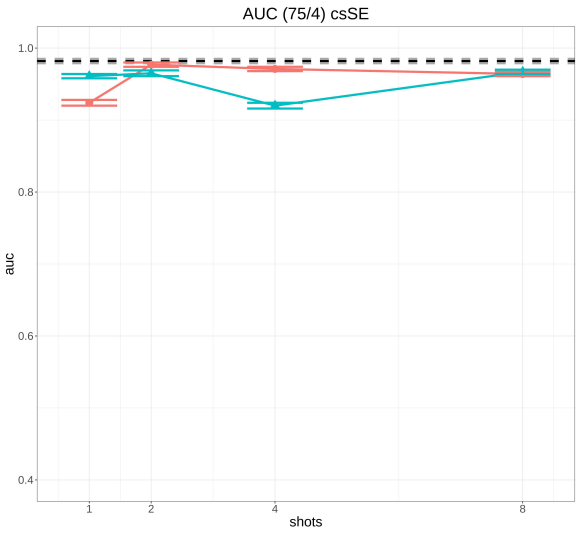
Figure 4.11: AUCs of two branch network without *csSE* blocks on Center 4 WSIs for two network configurations: 'ccat' points use 'concatenated features' connections, the others 'channel weights'.




(a) Patient 67, node 4 (micro-metastases)

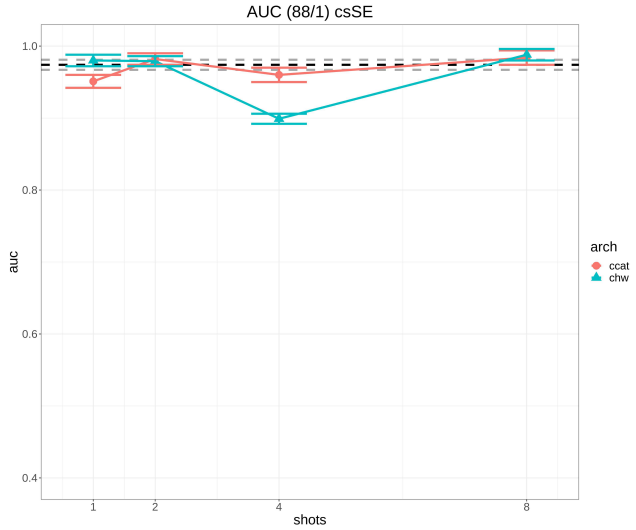



(b) Patient 72, node 0 (ITCs)

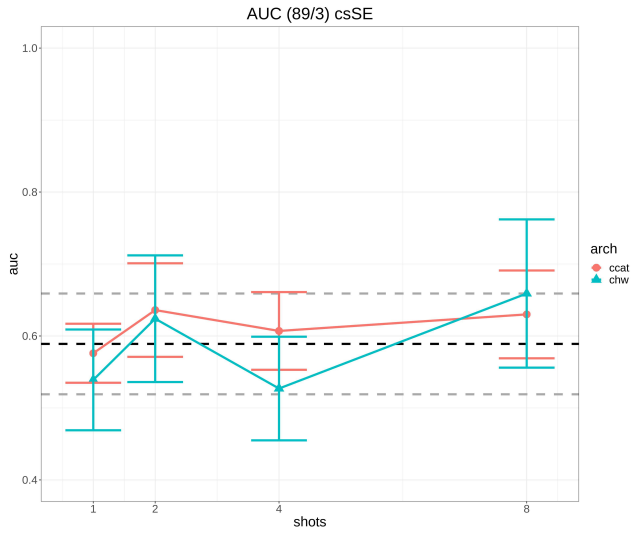



(c) Patient 75, node 4 (macro-metastases)

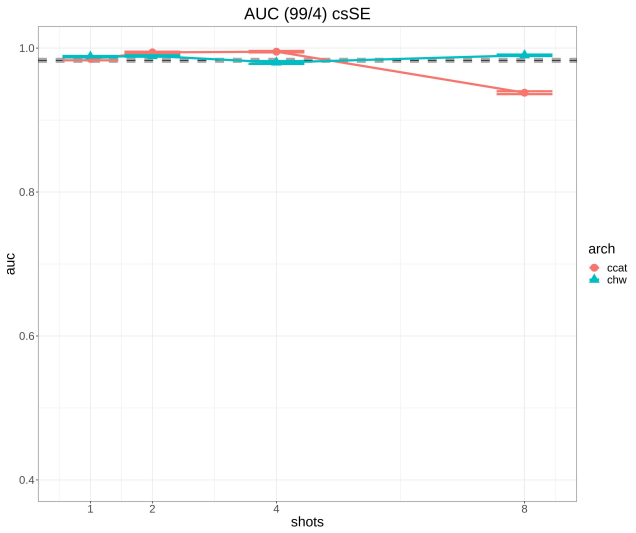
Figure 4.12: AUCs of two branch network with *csSE* blocks on Center 3 WSIs for two network configurations: 'ccat' points use 'concatenated features' connections, the others 'channel weights'.




(a) Patient 88, node 1 (micro-metastases)




(b) Patient 89, node 3 (ITCs)




(c) Patient 99, node 4 (macro-metastases)

Figure 4.13: AUCs of two branch network with *csSE* blocks on Center 4 WSIs for two network configurations: 'ccat' points use 'concatenated features' connections, the others 'channel weights'.

In addition, we can notice that both connections have similar performance although the 'features concatenation' choice seems to provide marginally better performances in all FSS network configurations (with and without *csSE* blocks used in the encoders and decoders). At 4 shot for all 12 tests but two (see Figures 4.10b and 4.12a) the 'features concatenation' connection performs better than the 'channel weights' variant. At two shots there are only two more cases where the 'features concatenation' is significantly worse than 'channel weights' (the WSIs with macro-metastases when no *csSE* blocks are used, see Figures 4.10c and 4.11c). At 8 shots there is only one instance where the 'features concatenation' is significantly worse (again a macro-metastases but this time with *csSE* blocks, see Figure 4.13c) and two where it is just marginally worse. In summary at 2, 4 and 8 shots, in all instances tested, the 'features concatenation' connections achieves better performances than the 'channel weights' connection.

## 4.2 Influence of the support set on inference results

Aside from variation in the network configuration, another important way inference results can change is by modifying the support set and the query patches to support patches association. The main ways these can change are through the tuning of the following:

- change of the microcluster dimension hyperparameter (see Subsection 3.3.3);
- change of the GMM clustering (see Subsection 3.3.2);
- change in the autoencoder.

In the following I explored the impact of the first two possible changes leaving an evaluation of changes in the autoencoder, e.g. the adoption of a variational autoencoder, for future work.

### 4.2.1 AUC scores with different microcluster dimensions

I conducted a further test to check the influence of the average microcluster dimension on the AUC scores. I tested the FSS network in the 'features concatenation' setup with no *csSE* blocks and at 4 shots. I summarize the

Table 4.7: AUC comparison vs baseline with no *csSE* blocks, concatenated features as connection between branches and 4 shots for different microcluster dimensions.

| | AUC variation | | | | | |
|---|---|---|---|---|---|---|
| **Microcluster** | **72/0** | **67/4** | **75/4** | **89/3** | **88/1** | **99/4** |
| **dimension** | **ITC** | **micro** | **macro** | **ITC** | **micro** | **macro** |
| 5 | -0.51% | 9.57% | -1.43% | 7.97% | -0.85% | 0.40% |
| 10 | 0.71% | 12.01% | -2.34% | 3.12% | 2.65% | 0.40% |
| 20 | -1.83% | 22.14% | -1.43% | 14.73% | 4.77% | 0.00% |
| 40 | -0.92% | -0.56% | -2.34% | 18.20% | 4.56% | -3.34% |

results, as a variation with respect to the U-Net baseline with no *csSE* blocks, in Table 4.7.

The tests confirm that setting the microcluster dimension of 20 gives balanced results. In particular on medical center 4 it is a good choice to improve the performances of the baseline U-Net on the most challenging WSIs, the ones containing ITCs and micro-metastases. This choice also achieves the best results for the WSI with micro-metastases of medical center 3 (patient 67, node 4).

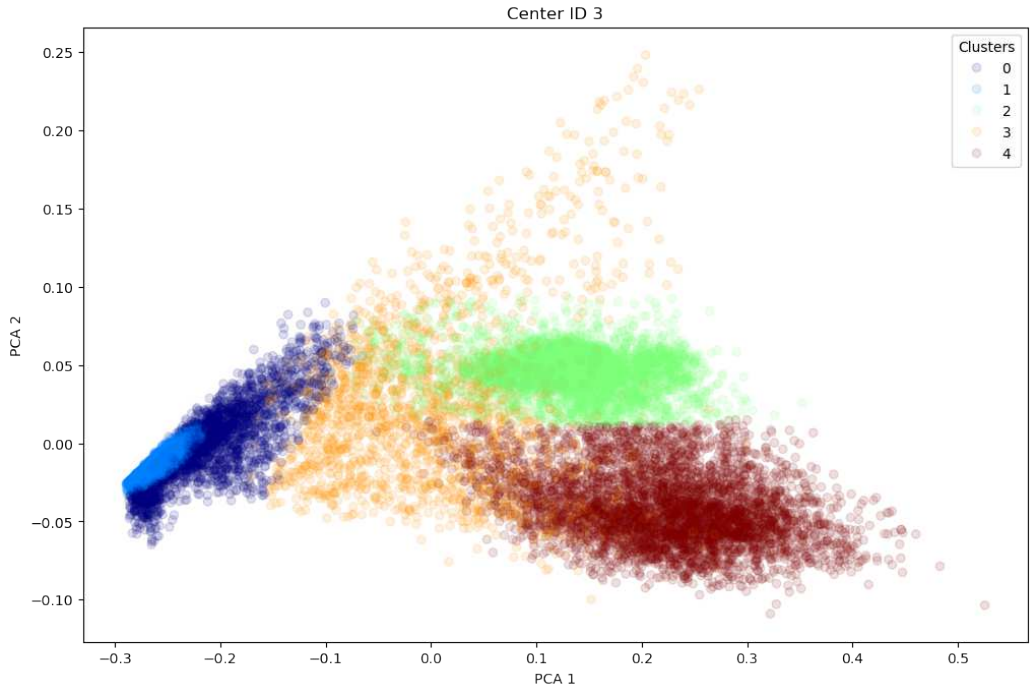
### 4.2.2 Test with manually optimized GMM clusters

Optimizing the GMM clustering for center 3 and center 4 modifies the results as shown below.

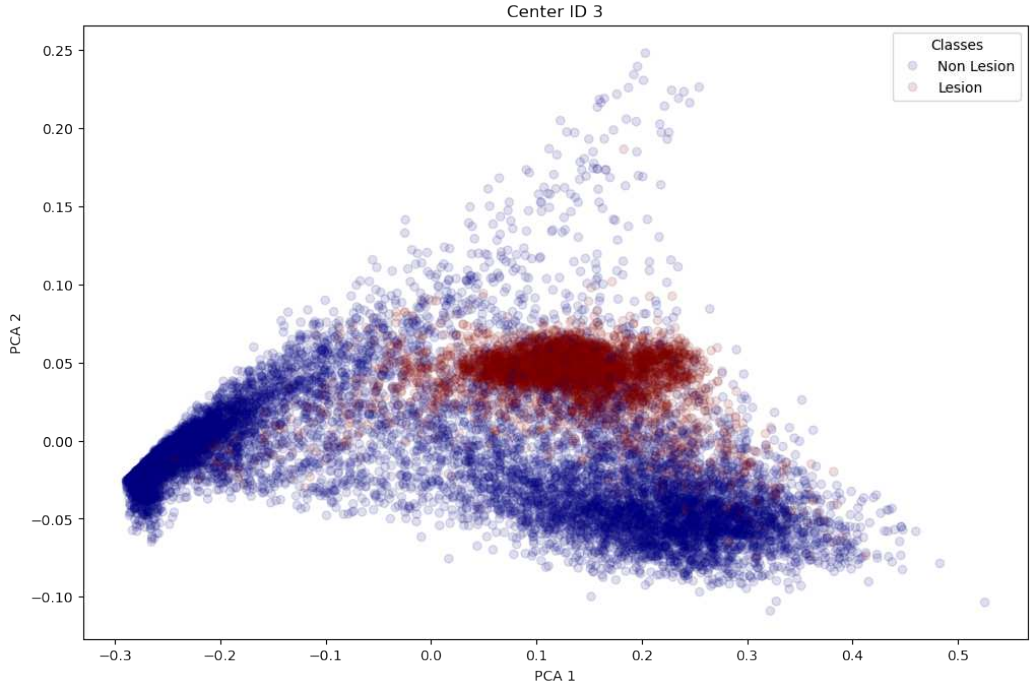
For medical center 3 I manually optimized by using Bayesian GMM and 5 components (clusters). Figure 4.14 shows, in the space of the first two principal components, the Bayesian GMM clusters distribution as well as the distribution of lesion (red points)/non-lesion (blue points) patches. A visual comparison of the two highlights the high correlation of at least one cluster with lesion patches.

For medical center 4 I manually optimized the clustering using Bayesian GMM and 8 components (clusters). Figure 4.15 shows, in the space of the first two principal components, the Bayesian GMM clusters distribution as well as the distribution of lesion (red points)/non-lesion (blue points) patches. Again, a visual comparison shows a strong correlation among some of the clusters and the lesion patches.

By using the new clustering and the newly computed probability estimate of lesion patches for each cluster (see Section 3.3.2) I re-run the inference. The outcome of these new experiments is summarized in tables 4.8 and 4.9 where I compare the AUC scores of the U-Net and of the same FSS network

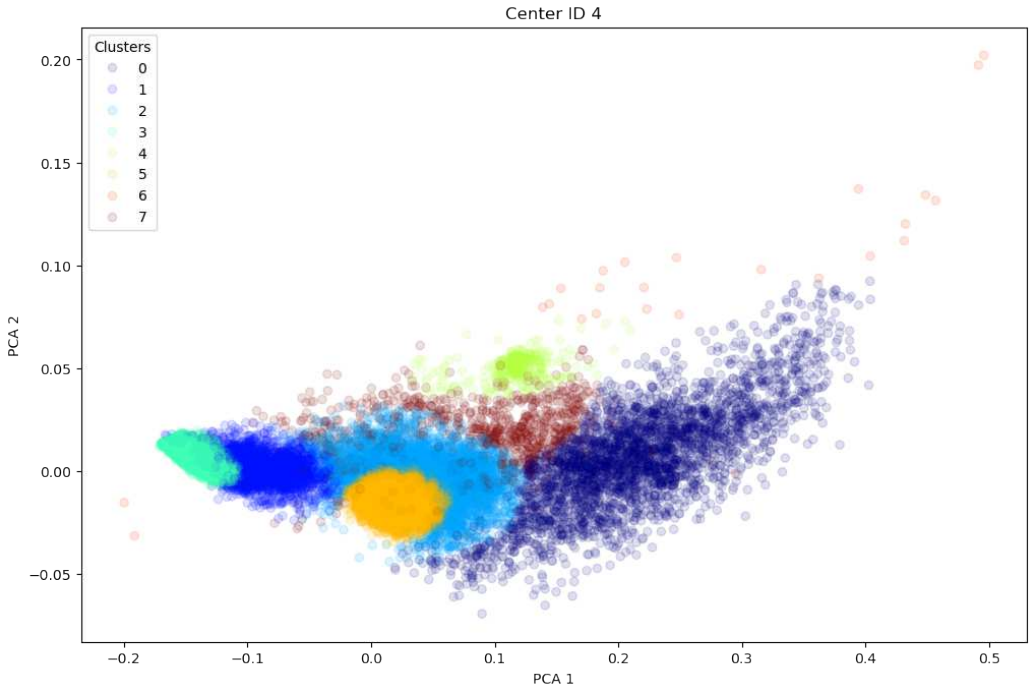



(a) Clusters Center 3

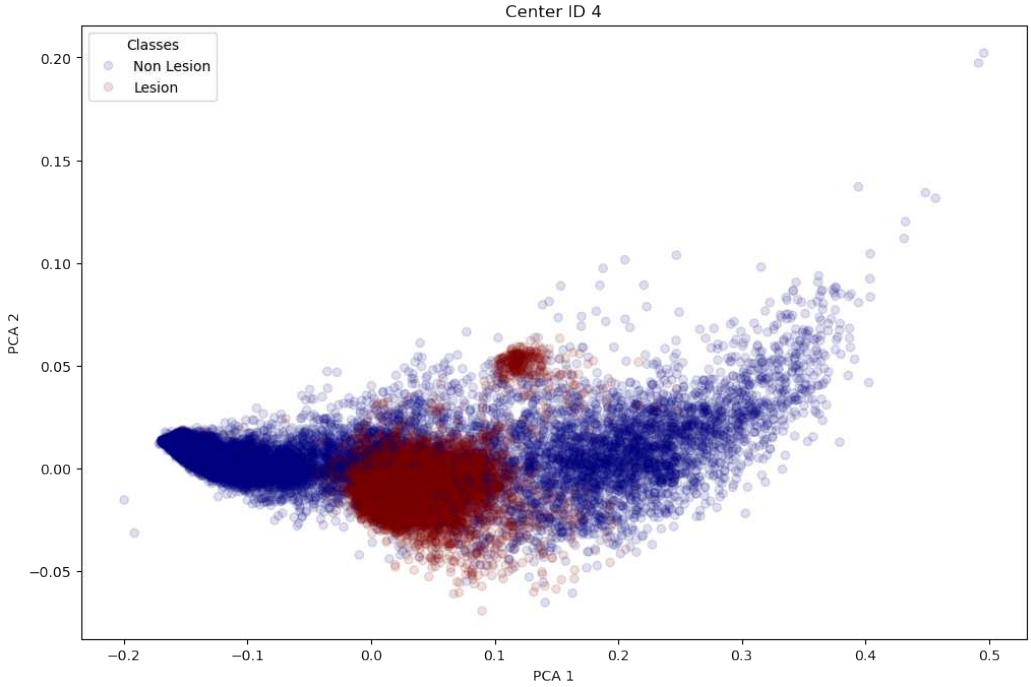



(b) Classes Center 3

Figure 4.14: Clusters and Classes for Medical Center 3




(a) Clusters Center 4




(b) Classes Center 4

Figure 4.15: Clusters and Classes for Medical Center 4

at 4 and 8 shots where different support sets are used:

- 'FSS GMM 6 clusters': the support set generated by the standard 6 clusters GMM described in Subsection 3.3.2;
- 'FSS Bayesian GMM 5 clusters': the support set generated by the Bayesian GMM clustering with 5 clusters for the WSIs of medical center 3;
- 'FSS Bayesian GMM 8 clusters' the support set generated by the Bayesian GMM clustering with 8 clusters for the WSIs of medical center 4.

Table 4.8: Comparison of FSS with supports generated by different clustering for medical center 3 WSIs.

| | | **AUC** | | |
|---|---|---|---|---|
| | | **72/0** | **67/4** | **75/4** |
| **Architecture** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $\mathbf{0.981 \pm 0.011}$ | $0.533 \pm 0.077$ | $\mathbf{0.982 \pm 0.003}$ |
| FSS GMM 6 clusters | 4 | $0.963 \pm 0.028$ | $0.651 \pm 0.075$ | $0.968 \pm 0.003$ |
| FSS GMM 6 clusters | 8 | $0.971 \pm 0.024$ | $0.555 \pm 0.082$ | $0.963 \pm 0.004$ |
| FSS Bayesian GMM 5 clusters | 4 | $0.970 \pm 0.026$ | $\mathbf{0.668 \pm 0.070}$ | $0.967 \pm 0.003$ |
| FSS Bayesian GMM 5 clusters | 8 | $0.977 \pm 0.014$ | $0.530 \pm 0.077$ | $0.964 \pm 0.004$ |

Table 4.9: Comparison of FSS with supports generated by different clustering for medical center 4 WSIs.

| | | **AUC** | | |
|---|---|---|---|---|
| | | **89/3** | **88/1** | **99/4** |
| **Architecture** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.577 \pm 0.068$ | $0.943 \pm 0.008$ | $\mathbf{0.989 \pm 0.002}$ |
| FSS GMM 6 clusters | 4 | $0.662 \pm 0.075$ | $0.988 \pm 0.004$ | $\mathbf{0.989 \pm 0.002}$ |
| FSS GMM 6 clusters | 8 | $\mathbf{0.708 \pm 0.091}$ | $0.990 \pm 0.006$ | $0.986 \pm 0.002$ |
| FSS Bayesian GMM 8 clusters | 4 | $0.676 \pm 0.081$ | $0.988 \pm 0.005$ | $0.988 \pm 0.002$ |
| FSS Bayesian GMM 8 clusters | 8 | $0.687 \pm 0.106$ | $\mathbf{0.992 \pm 0.006}$ | $0.986 \pm 0.002$ |

For medical center 3 the best FSS mean AUC score increases by 0.7% for the WSI with ITCs (patient 72, node 0) and by 2.6% for the WSI with micro-metastases (patient 67, node 4). For medical center 4 the best FSS AUC scores decreases by 3% for the WSI with ITCs (patient 89, node 3) and it increases by just 0.2% for the WSI with micro-metastases (patient 88, node 1). On both WSIs with macro-metastases the mean AUC scores remain substantially unchanged.

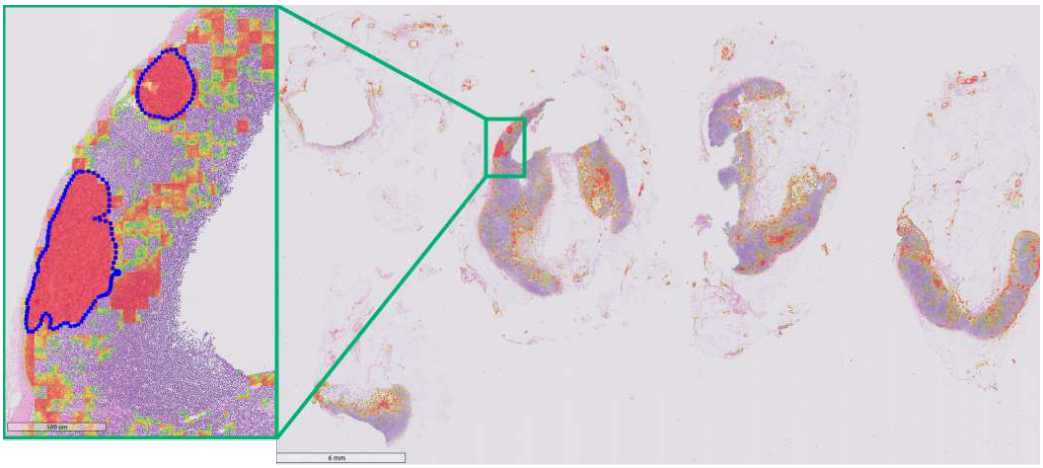
On the WSI with micro-metastases of medical center 4 the best FSS performance is obtained at 8 shots with new Bayesian GMM clustering. The results can best be seen in Figure 4.16a. The result is even more interesting if compared with the regular U-Net as seen in Figure 4.16b where effectively the U-Net classifies the entire region with tissue as potential lesion. The comparison of the two ROC curves in Figure 4.17a confirms the better performance of the FSS which is most prominent in the *partial AUC* (pAUC), a measure which summarizes a portion of the ROC curve with its AUC over a range of interest, either a specificity or sensitivity range, instead of the full curve: in the range of specificity 90%-100% the pAUC for the FSS is 96.8% as shown in Figure 4.17b higher than the pAUC of 72.0 % for the U-Net at the same specificity range (see Figure 4.17c). Similarly the pAUC of the FSS in the sensitivity range 90%-100% is 96.5%, almost 4% points higher than the pAUC in the same range of the U-Net (see Figures 4.17b and 4.17c).

The same behavior can be seen on the WSI with ITCs from medical center 4 (patient 89, node 3). Again a comparison of the two ROC curves is shown in Figure 4.18a. The comparison of the two AUCs with DeLong et al. (1988) gives a rather high *p-value* of 0.08, but the behaviour of the two networks is rather different. The U-Net shows very poor specificity; its pAUC, in fact, in the specificity range 90%-100% cannot even be evaluated because the U-Net specificity is always lower than the 90% threshold. The FSS instead shows a more regular behaviour with its ROC curve above the diagonal at every specificity and with an higher AUC overall as shown in Table 4.9. The different behaviour is even more evident in Figure 4.19 where the FSS heatmap (Figure 4.19a) is compared against the U-Net heatmap (Figure 4.19b). The FSS detects all except one of the ITCs, but what is more evident is that the U-Net classifies all the regions with tissue as potential lesion, confirming the very low specificity seen in the ROC curve, and it would add no information in screening the WSI. The FSS is still very conservative, but it does a better job in discriminating areas where ITCs could be present from the rest.

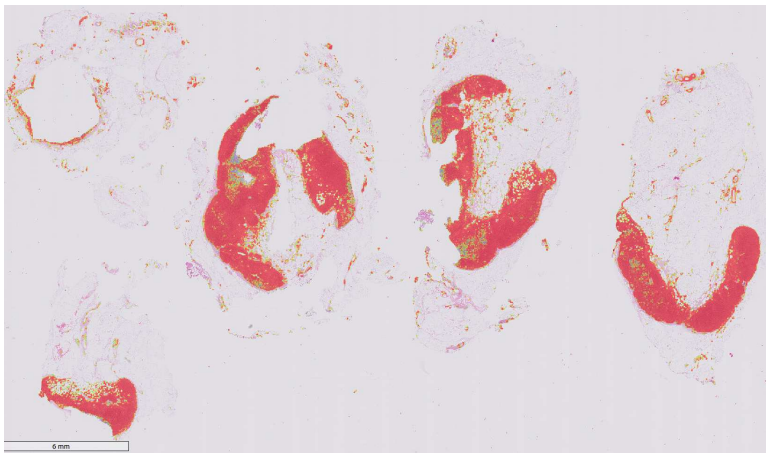
## 4.3 Summary

In this chapter, I have shown that:

- the conditioning branch provide better performances with respect to a plain U-Net like architecture on the WSIs of the CAMELYON17 medical center 4, which uses a scanner not used by any of the medical centers present in the training set;

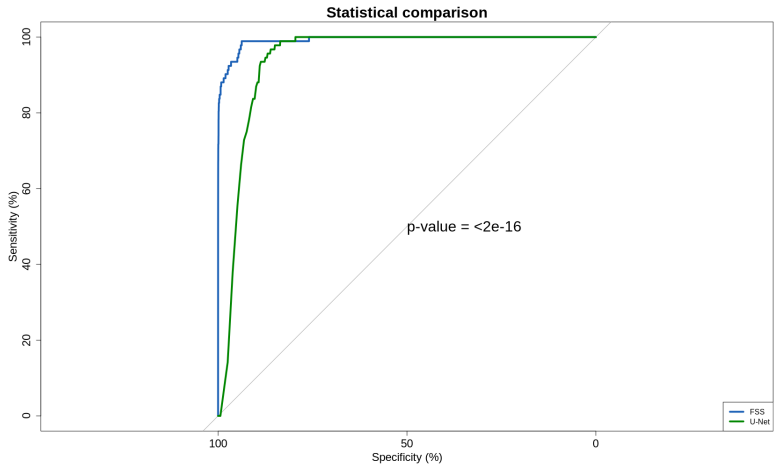


(a) FSS @ 8-shots

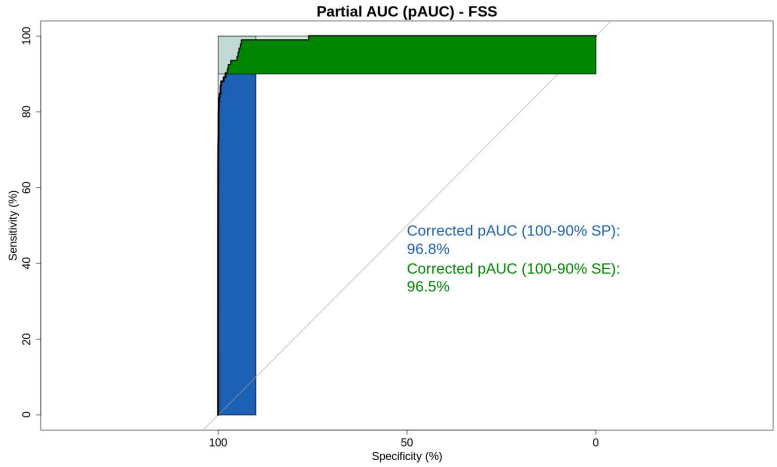


(b) U-Net

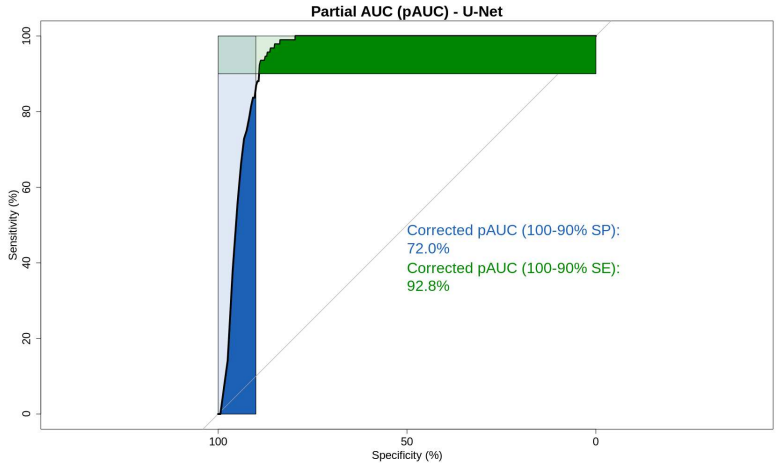
Figure 4.16: Micro-metastases in patient 88 of medical center 4 as detected by the FSS with no *csSE* blocks and 'features concatenation' (the new Bayesian GMM clustering was used to create the support set) and by the U-Net with no *csSE* blocks. Prediction probabilities below 0.75 are transparent, between 0.75 and 1.0 are shown with hues changing from green to red.




(a) ROC FSS and U-Net

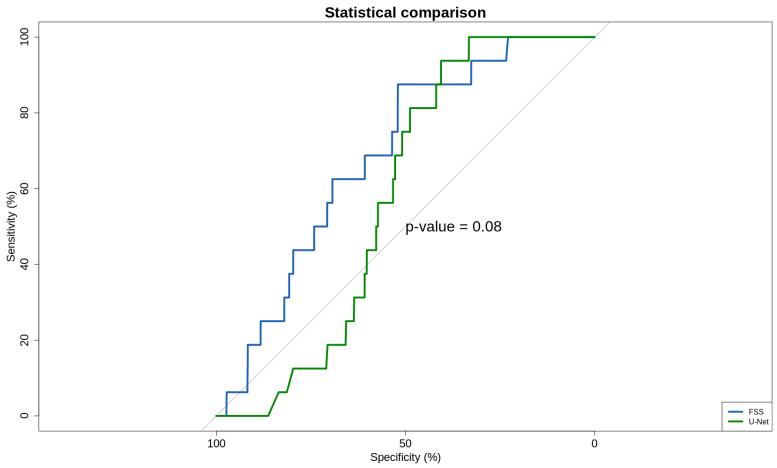



(b) pAUC FSS

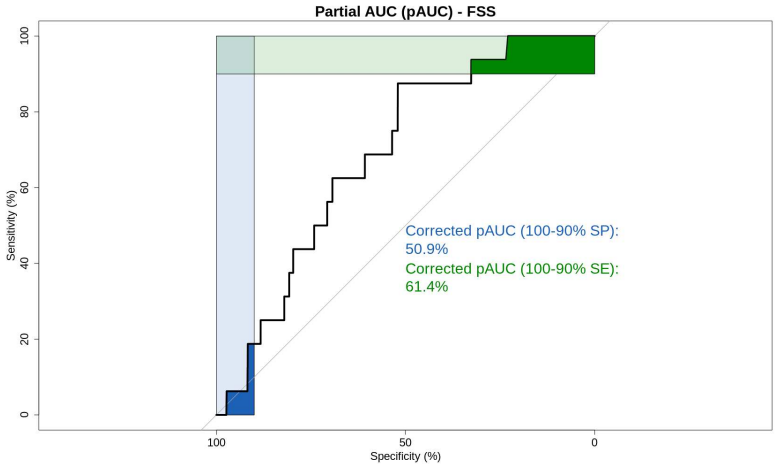



(c) pAUC U-Net

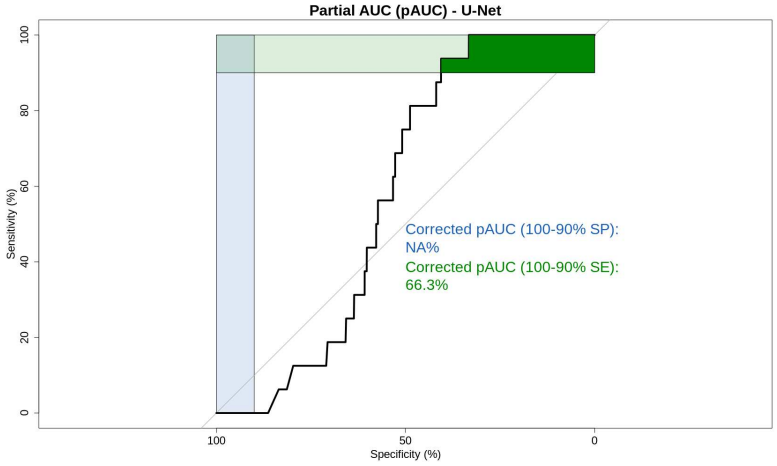
Figure 4.17: ROC comparison of FSS vs U-Net for patient 88 with micrometastases of medical center 4.




(a) ROC FSS and U-Net

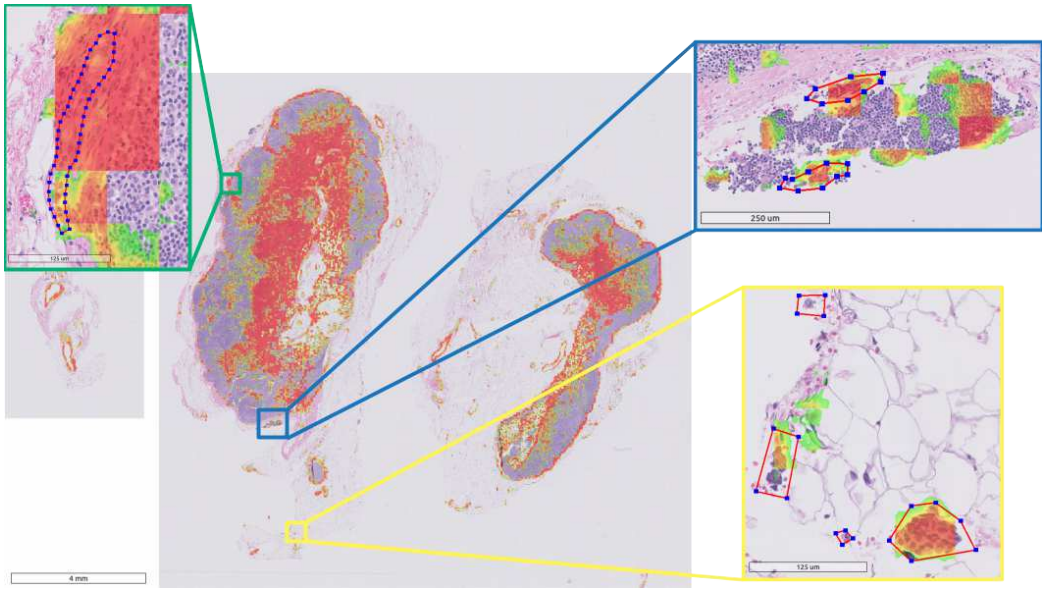



(b) pAUC FSS

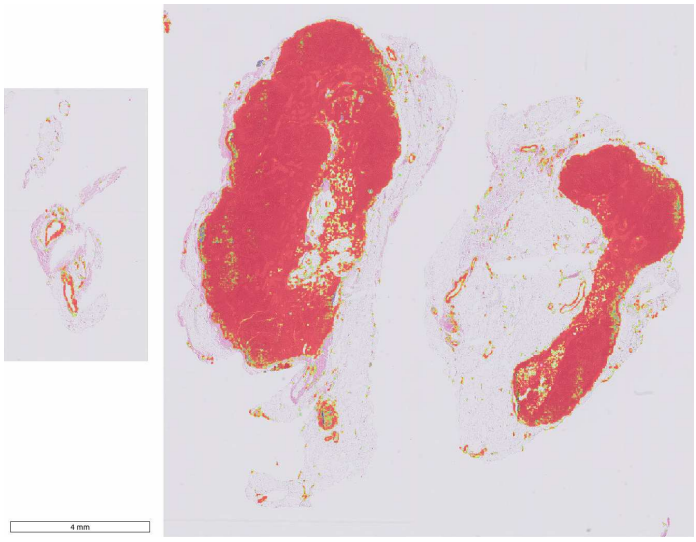



(c) pAUC U-Net

Figure 4.18: ROC comparison of FSS vs U-Net for patient 89 with ITCs of medical center 4.




(a) FSS @ 8-shots




(b) U-Net

Figure 4.19: ITCs in patient 89 of medical center 4 as detected by the FSS with no *csSE* blocks and 'features concatenation' (new Bayesian GMM clustering was used to create the support set) and by the U-Net with no *csSE* blocks. Prediction probabilities below 0.75 are transparent, the others are shown with hues from green (0.75) to red (1.0).

- the experiments confirm that the conditioning is able to extract information from the support set and influence the behavior of the network at inference time;
- the selection of the support set is important as shown in Subsection 4.2.2 where the tuning of the clustering algorithm, if we exclude the macro-metastases where it made no sensible difference, slightly improved the FSS AUC scores for 3 out of 4 WSIs.

Chapter 5

# Discussion

In this chapter, I summarize the main contributions and possible further developments of this research.

## 5.1 Contributions

In this study, I discussed, to the best of my knowledge, the first application of a conditional FCN to digital pathology. This network, originated and evolved in the machine learning subdomain of few-shot learning, has initially been developed to learn to classify and then segment new classes of objects in natural images starting from few examples. I have explored the changes that have to be introduced in the architecture, already successfully used in radiology as shown by Guha Roy et al. (2020), to be applicable to digital pathology.

### 5.1.1 Architectural changes and motivations

I have introduced and explored the following changes to the network architecture:

1. I reintroduced skip connections in the segmentation branch with respect to the original architecture of Guha Roy et al. (2020), as detailed in Subsection 3.1.1; this modification was possible due to the related change in the composition of the support set as explained below;

2. I introduced and tested different ways for the conditioning and segmentation branch to be connected together.

About point 1, recalling that all reference architectures, including Rakelly et al. (2018) and Guha Roy et al. (2020), included in the support set both the WSI patches and the corresponding annotations, an important change in my architecture is that I use image patches only in the support set, i.e. the support set does not contain image patches and their corresponding annotation masks. This allowed for the skip connections to be reintroduced. In fact, if we keep the annotation masks in the support set and the skip connections in the segmentation branch, the network, especially with limited number of shots, learns to copy the annotation masks, found in the support set, as the final output of the segmentation.

Not having the annotation masks not only allows for the skip connections to be retained, an architectural choice which improves the resolution of the FSS output, it also allows to define a pretext loss associated with the output of the conditioning branch and which is used to regularize and improve the training (see Subsection 3.4.1). Because, in fact, the support patches are extracted from lesion and non-lesion regions based on the lesion probability estimate of each cluster (according to the policy described in Section 3.4), if the conditioning branch would receive as input, together with the support patches, also the corresponding segmentation masks, the branch could learn to regress the lesion probability estimate $\pi_l(I_q)$ (see Subsection 3.4.1) from the segmentation masks only[1]. This would impact adversely the effectiveness of the pretext task, which is instead able to learn an association between the support patch images (without their annotation masks) and the lesion probability estimate of the GMM cluster they belong to.

About point 2, another change that was introduced in the reference architecture of Guha Roy et al. (2020) is the way the conditioning and segmentation branches are connected. The original architecture was tasked with segmenting organs in CT scans, as such the spatial and local information the

[1]The following example can help clarify what are the potential effects of adding the segmentation masks to the support set. Supposing that a query image belongs to a cluster with an high density of lesion patches, for example 0.9, and supposing that the network uses a 4 shots support set, according to the policy described in Section 3.4, each shot of the support set except the last one would have to be a lesion patch because the binary representation of the integer part of $0.9 * 2^4 = 14.4$ is 1110. The first three support patches would then be lesion patches with binary segmentation masks pixels being mostly ones, and the last support patch would have instead a binary segmentation mask composed mostly of zeros. Given this input, the conditioning branch could learn to spatially average each of the 4 segmentation masks to approximate the binary representation 1110 and then regress this representation to the estimate, 0.9, of the lesion probability. If no segmentation masks are provided as input, the conditioning branch still has to learn to predict the same lesion probability, but starting from the support patches only, without the additional information provided by the segmentation masks.

support set can pass to the segmentation branch is relevant. Obviously in digital pathology patches organs are not visible as this information is absent. This is probably the most likely reason why the original reference architecture, as confirmed by early tests, when applied to the digital pathology did not work satisfactorily. As such I tested and introduced two variations to the reference architecture with the use of the *MX* blocks discussed in Subsection 3.1.1:

- 'channel weights' connections;
- 'features concatenation' connections.

The first connection type uses the SE blocks introduced by Roy et al. (2019) for FCNs. The second type of connection is instead inspired by the work of Rakelly et al. (2018) and leaves the freedom to the network to learn a distance metric between the feature maps of the query patches and the feature maps of the $k$ support shots. As discussed in Subsection 4.1.5 the 'features concatenation' connection appears to provide marginally better performances for histopathology slides than the 'channel weights' connection choice. It appears therefore that the sensible choice, especially at not too large shots, is to allow the network to learn a distance metric between the query feature maps and the support feature maps.

### 5.1.2 Support set selection method

From all the experiments conducted with the FSS it is apparent that the choice of a relevant support set is critically important for the conditioning to work properly. I conducted various tests and I evaluated various policies before settling on the one discussed in Section 3.4. In particular my initial choice was to pass as support a random selection of support patches extracted from a pool of lesion and non-lesion patches, one shot each, similarly to the policy adopted, for natural images, by Rakelly et al. (2018). This approach aimed at training the network to identify similarities and differences between the query patches and the $k$ support shots. However when the network is presented a random stream of (query, support) pairs, the network did not learn to extract the information from the support necessary to effectively condition the segmentation of the query patches. This is most likely due to the significant variation of characteristics and features among the patches extracted, even from a single WSI.

I have therefore introduced a strategy for the selection of the support set which was correlated to the input query patches. The techniques, which I have described in Chapter 3 and which relies on a combination of methods

such as autoencoders and GMM clustering has also been inspired by the successful unsupervised learning described in Yamamoto et al. (2019). An advantage of my method is that, in order to compute the lesion prevalence value $\pi_l$ for each cluster (see Subsection 3.3.2), only patch level classification is necessary. That opens up the opportunity to rely on sparse annotations only to create the support set, with a considerable reduction in the effort of pathologists to assemble the support set. This is, of course, similar to the opportunity already explored by Rakelly et al. (2018) in their conditional FCN architecture.

### 5.1.3 Domain adaptation

The U-Net applied to WSIs from medical center 4, which uses a digital slide scanner not used in any other centers, shows many more false positives compared to the FSS trained on the same set of WSIs but also conditioned on a support set coming from center 4 as well. This is especially true for the more difficult WSIs containing ITCs and micro-metastases:

- the best AUC for the WSI of patient 89 (with ITCs) obtained by the U-Net is $0.589 \pm 0.070$ with *csSE* blocks (see Table 4.2), a result improved by 20% by the FSS with 'features concatenation' and no *csSE* blocks at 8 shots: $0.708 \pm 0.091$, see Table 4.9;

- the best AUC for the WSI of patient 88 (with micro-metastases) obtained by the U-Net $0.974 \pm 0.007$ is improved by 2% by the FSS with 'features concatenation' and no *csSE* blocks at 8 shots: $0.992 \pm 0.006$ (see Table 4.9); the difference, due to the high number of false positives raised by the U-Net, is more striking with the pAUC in the specificity range 90%-100%, in this case the U-Net tops at 0.720 pAUC a result improved by 34.4% by the FSS which achieves a pAUC of 0.968 (see Figures 4.17c and 4.17b).

These improvements, which are mostly noticeable in the reduction of false positive regions by the U-Net, highlights the potential of the FSS to be used to contain the domain shift issue and enables the network to be effectively used for screening of WSIs especially for the most challenging cases of ITCs and micro-metastases.

## 5.2 Developments

This research opens up many future possible investigation streams. In the following I mention some of them.

### 5.2.1 Network architecture and training

Further tests would be necessary to better understand the influence and the interaction between the 'features concatenation' connection and the number of shots in the support set. There is, in fact, a likely interaction between the width of the conditioning branch (see Section 3.1), the number of shots, and the width of the feature maps passed to the segmentation branch: in order to confidently process and extract information from a large number of shots the conditioning branch needs to be wide enough, however, if a wide encoder/decoder block passes a wide feature maps to the segmentation branch, the segmentation branch might discard the information of the query patches and just rely on the support set to output the predicted segmentation mask. Therefore it could be beneficial to shrink the connections between a wide conditioning branch and a segmentation branch with further blocks (such as a fully connected layer interposed between all connections).

The optimal number of encoders/decoders levels, as well as the most effective width to be used at each level, should be assessed to identify an improved configuration of the architecture. In a similar fashion, the choice proposed of using only 'channel weights' or 'features concatenation' connections was due to time constraints. A mix of the two connections in which, for example, the external layers could use one type of connection and the internal ones another, has not been explored.

Another area that might be further explored is the impact of *episodic* training to the performance of the FSS: mini-batches could be grouped by medical center, or by medical center and lesion type (ITCs, micro or macro-metastasis) or even by WSI. Doing so would better align the training and inference phases and might provide an improvement of the FSS network altogether. In fact, early work in few-shot learning by Vinyals et al. (2016) suggested that episodic training could improve the performance although this approach has recently been questioned work by Laenen and Bertinetto (2020).

### 5.2.2 Support set selection

We have already seen that the proper selection of the support set is a crucial step for the successful training and inference of the FSS. This is therefore an area worth further investigation.

The first possible development would be to use more sophisticated autoencoder architectures than the one described in Subsection 3.3.1, such as a variational autoencoder. Another approach could be to use two autoencoders operating at different resolutions, similar to what described by Yamamoto

et al. (2019).

Similarly, the clustering algorithm has the potential to be further explored and improved. Early experiments were conducted with Hierarchical DBSCAN clustering (McInnes et al., 2017). This choice had the advantage of not requiring a pre-selection of the number of clusters, however the results proved unstable, and often failing to separate patches which appeared to belong to different populations in the PCA latent space. Many different clustering algorithms are available and have not been explored to identify an optimal choice. Analogously a simple k-means clustering algorithm was used to extract the support prototypes (see Subsection 3.3.3). This is also an area which could be further investigated as it might lead to better and more sophisticated approaches.

Very recent advances in few-shot learning such as the novel Transformer discussed in Doersch et al. (2020) could also be applied to find a better correspondence between the query and the support patches. This new architecture, in fact, has been tuned to find spatially-corresponding features between query images and a small number of labeled images, as such it could replace the method, that I described in Section 3.4, and which is based on a simple Euclidean distance metric in the PCA latent space, to associate each query patch to the most similar patches in the support set.

Finally, it is conceivable that the pathologist, instead of pre-annotating the support WSIs (even with sparse, patch level annotations) could first let the unsupervised algorithms process and cluster the WSI patches and then post-classify the clusters as a whole. The pathologist could in fact sample patches belonging to each cluster and classify just a few samples in order to provide a reasonable estimate of the probability of lesion patches (ref. Equation 3.1) for each cluster. This approach would have the advantage of speeding up the assembling of a large, representative and diversified support set.

### 5.2.3 Domain adaptation

One of the most interesting potential of the FSS architecture is to tackle the domain shift issue present with histopathology slides. Further tests should be conducted to better understand the full potential of the approach, in particular tests could be conducted on other datasets to diagnose other diseases, such as prostate and lung cancers.

## 5.3 Summary

In this section, I have discussed my main contributions:

- the application of a few-shot segmentor to digital pathology and the architectural changes introduced to the architecture to make it effective to the screening of WSIs;
- the introduction of a novel support set selection process which could open up the opportunity to assemble the support set with just sparse annotations
- the exploration of the impact of this architecture to address the domain shift problem which exists in the histopathology domain

I have also summarized possible future development streams to further improve on the contributions just mentioned.

Appendix A

# Appendix A

Details of the AUC scores obtained with the two hyper-network configurations discussed in chapter 4 are provided here.

## A.1 Chanel weights

The AUC scores with Confidence Interval (CI) at 95% are provided in the following tables:

- for the configuration without *csSE* blocks tables A.1 and A.2;
- for the configuration with *csSE* blocks, tables A.3 and A.4.

The CI is computed with the `ci.auc` command of the `pROC` R package as discussed in Subsection 4.1.3. Because this function implements the method derived from DeLong et al. (1988), in the tables the reference '(DeLong)' is added for clarity.

## A.2 Features concatenation

The AUC scores with CI at 95% are provided in the following tables:

- for the configuration without *csSE* blocks tables A.5 and A.6;
- for the configuration with *csSE* blocks, tables A.7 and A.8.

Table A.1: AUC with 95% CI (DeLong) of FSS with no *csSE* block and with channel weights connections between branches compared with baseline U-Net for Center 3 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | | **72/0** | **67/4** | **75/4** |
| **Arch.** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.981 \pm 0.011$ | $0.533 \pm 0.077$ | $0.982 \pm 0.003$ |
| FSS | 1 | $0.981 \pm 0.006$ | $0.511 \pm 0.079$ | $0.951 \pm 0.004$ |
| FSS | 2 | $0.956 \pm 0.036$ | $0.513 \pm 0.076$ | $0.956 \pm 0.004$ |
| FSS | 4 | $0.971 \pm 0.012$ | $0.528 \pm 0.075$ | $0.919 \pm 0.004$ |
| FSS | 8 | $0.975 \pm 0.019$ | $0.556 \pm 0.081$ | $0.961 \pm 0.004$ |

Table A.2: AUC with 95% CI (DeLong) of FSS with no *csSE* block and with channel weights connections between branches compared with baseline U-Net for Center 4 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | | **89/3** | **88/1** | **99/4** |
| **Arch.** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.577 \pm 0.068$ | $0.943 \pm 0.008$ | $0.989 \pm 0.002$ |
| FSS | 1 | $0.555 \pm 0.074$ | $0.964 \pm 0.011$ | $0.982 \pm 0.002$ |
| FSS | 2 | $0.557 \pm 0.057$ | $0.991 \pm 0.004$ | $0.987 \pm 0.001$ |
| FSS | 4 | $0.628 \pm 0.074$ | $0.963 \pm 0.009$ | $0.976 \pm 0.002$ |
| FSS | 8 | $0.583 \pm 0.106$ | $0.986 \pm 0.010$ | $0.988 \pm 0.001$ |

Table A.3: AUC with 95% CI (DeLong) of FSS with *csSE* block and with channel weights connections between branches compared with baseline U-Net for Center 3 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | | **72/0** | **67/4** | **75/4** |
| **Arch.** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.977 \pm 0.014$ | $0.604 \pm 0.087$ | $0.982 \pm 0.003$ |
| FSS | 1 | $0.981 \pm 0.008$ | $0.568 \pm 0.071$ | $0.961 \pm 0.003$ |
| FSS | 2 | $0.950 \pm 0.032$ | $0.556 \pm 0.076$ | $0.965 \pm 0.004$ |
| FSS | 4 | $0.979 \pm 0.012$ | $0.561 \pm 0.074$ | $0.920 \pm 0.004$ |
| FSS | 8 | $0.959 \pm 0.040$ | $0.616 \pm 0.075$ | $0.967 \pm 0.003$ |

Table A.4: AUC with 95% CI (DeLong) of FSS with *csSE* block and with channel weights connections between branches compared with baseline U-Net for Center 4 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | | **89/3** | **88/1** | **99/4** |
| **Arch.** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.589 \pm 0.070$ | $0.974 \pm 0.007$ | $0.983 \pm 0.002$ |
| FSS | 1 | $0.539 \pm 0.070$ | $0.980 \pm 0.008$ | $0.988 \pm 0.001$ |
| FSS | 2 | $0.624 \pm 0.088$ | $0.979 \pm 0.007$ | $0.989 \pm 0.001$ |
| FSS | 4 | $0.527 \pm 0.072$ | $0.899 \pm 0.007$ | $0.980 \pm 0.002$ |
| FSS | 8 | $0.659 \pm 0.103$ | $0.988 \pm 0.008$ | $0.990 \pm 0.001$ |

Table A.5: AUC with 95% CI (DeLong) of FSS without *csSE* block and concatenated connections between branches compared with baseline U-Net for Center 3 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | | **72/0** | **67/4** | **75/4** |
| **Architecture** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.981 \pm 0.011$ | $0.533 \pm 0.077$ | $0.982 \pm 0.003$ |
| FSS | 1 | $0.972 \pm 0.028$ | $0.506 \pm 0.08$ | $0.97 \pm 0.003$ |
| FSS | 2 | $0.956 \pm 0.015$ | $0.543 \pm 0.083$ | $0.927 \pm 0.004$ |
| FSS | 4 | $0.963 \pm 0.028$ | $0.651 \pm 0.075$* | $0.968 \pm 0.003$ |
| FSS | 8 | $0.971 \pm 0.024$ | $0.555 \pm 0.082$ | $0.963 \pm 0.004$ |

Table A.6: AUC with 95% CI (DeLong) of FSS without *csSE* block and concatenated connections between branches compared with baseline U-Net for Center 4 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | | **89/3** | **88/1** | **99/4** |
| **Architecture** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.577 \pm 0.068$ | $0.943 \pm 0.008$ | $0.989 \pm 0.002$ |
| FSS | 1 | $0.579 \pm 0.049$ | $0.948 \pm 0.013$ | $0.986 \pm 0.002$ |
| FSS | 2 | $0.610 \pm 0.075$ | $0.993 \pm 0.003$ | $0.975 \pm 0.002$ |
| FSS | 4 | $0.662 \pm 0.075$ | $0.988 \pm 0.004$ | $0.989 \pm 0.002$ |
| FSS | 8 | $0.708 \pm 0.091$ | $0.990 \pm 0.006$ | $0.986 \pm 0.002$ |

Table A.7: AUC with 95% CI (DeLong) of FSS with *csSE* block and concatenated connections between branches compared with baseline U-Net for Center 3 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | **72/0** | | **67/4** | **75/4** |
| **Architecture** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.977 \pm 0.014$ | $0.604 \pm 0.087$ | $0.982 \pm 0.003$ |
| FSS | 1 | $0.956 \pm 0.025$ | $0.584 \pm 0.071$ | $0.924 \pm 0.004$ |
| FSS | 2 | $0.972 \pm 0.014$ | $0.476 \pm 0.099$ | $0.977 \pm 0.003$ |
| FSS | 4 | $0.978 \pm 0.018$ | $0.523 \pm 0.086$ | $0.971 \pm 0.003$ |
| FSS | 8 | $0.983 \pm 0.012$ | $0.507 \pm 0.070$ | $0.964 \pm 0.003$ |

Table A.8: AUC with 95% CI (DeLong) of FSS with *csSE* block and concatenated connections between branches compared with baseline U-Net for Center 4 WSIs.

| | | AUC | | |
|---|---|---|---|---|
| | | **89/3** | **88/1** | **99/4** |
| **Architecture** | **Shots** | **ITC** | **micro** | **macro** |
| U-Net | - | $0.589 \pm 0.070$ | $0.974 \pm 0.007$ | $0.983 \pm 0.002$ |
| FSS | 1 | $0.576 \pm 0.041$ | $0.951 \pm 0.009$ | $0.985 \pm 0.002$ |
| FSS | 2 | $0.636 \pm 0.065$ | $0.982 \pm 0.008$ | $0.994 \pm 0.001$ |
| FSS | 4 | $0.607 \pm 0.054$ | $0.960 \pm 0.010$ | $0.995 \pm 0.001$ |
| FSS | 8 | $0.630 \pm 0.061$ | $0.984 \pm 0.010$ | $0.938 \pm 0.002$ |

# Bibliography

G. Campanella, M. G. Hanna, L. Geneslaw, A. Miraflor, V. Werneck Krauss Silva, K. J. Busam, E. Brogi, V. E. Reuter, D. S. Klimstra, and T. J. Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature Medicine*, 25(8):1301–1309, aug 2019. ISSN 1546170X. doi: 10.1038/s41591-019-0508-1. URL `https://pubmed.ncbi.nlm.nih.gov/31308507/`.

P. H. C. Chen, K. Gadepalli, R. MacDonald, Y. Liu, S. Kadowaki, K. Nagpal, T. Kohlberger, J. Dean, G. S. Corrado, J. D. Hipp, C. H. Mermel, and M. C. Stumpe. An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis. *Nature Medicine*, 25(9):1453–1457, sep 2019. ISSN 1546170X. doi: 10.1038/s41591-019-0539-7.

T. de Bel, M. Hermsen, J. van der Laak, G. J. S. Litjens, B. Smeets, and L. Hilbrands. Automatic segmentation of histopathological slides of renal tissue using deep learning. 2018. ISBN 9781510616516. doi: 10.1117/12.2293717.

E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson. Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach. *Biometrics*, 1988. ISSN 0006341X. doi: 10.2307/2531595.

N. Dimitriou, O. Arandjelović, and P. D. Caie. Deep Learning for Whole Slide Image Analysis: An Overview. *Frontiers in Medicine*, 6, nov 2019. ISSN 2296858X. doi: 10.3389/fmed.2019.00264.

C. Doersch, A. Gupta, and A. Zisserman. CrossTransformers: Spatially-aware few-shot transfer, 2020. ISSN 23318422. URL `http://arxiv.org/abs/2007.11498`.

W. Falcon. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

A. R. Feyjie, R. Azad, M. Pedersoli, C. Kauffman, I. B. Ayed, and J. Dolz. Semi-supervised few-shot learning for medical image segmentation. 2020. URL `http://arxiv.org/abs/2003.08462`.

G. Gerard and M. Piastra. Slide Screening of Metastases in Lymph Nodes via Conditional, Fully Convolutional Segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. ISBN 9783030307530. doi: 10.1007/978-3-030-30754-7_22.

A. Guha Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger. 'Squeeze & excite' guided few-shot segmentation of volumetric images. *Medical Image Analysis*, 59:101587, jan 2020. ISSN 13618423. doi: 10.1016/j.media.2019.101587.

K. He, X. Zhang, S. Ren, and J. Sun. ResNet. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. ISSN 10636919.

T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-Learning in Neural Networks: A Survey, apr 2020. ISSN 23318422. URL `http://arxiv.org/abs/2004.05439`.

J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7132–7141. IEEE, jun 2018. ISBN 9781538664209. doi: 10.1109/CVPR.2018.00745.

S. Laenen and L. Bertinetto. On Episodes, Prototypical Networks, and Few-shot Learning. 2020. URL `http://arxiv.org/abs/2012.09831`.

G. Litjens, C. I. Sánchez, N. Timofeeva, M. Hermsen, I. Nagtegaal, I. Kovacs, C. Hulsbergen-Van De Kaa, P. Bult, B. Van Ginneken, and J. Van Der Laak. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Scientific Reports*, 2016. ISSN 20452322. doi: 10.1038/srep26286.

G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis, 2017. ISSN 13618423.

G. Litjens, P. Bandi, B. E. Bejnordi, O. Geessink, M. Balkenhol, P. Bult, A. Halilovic, M. Hermsen, R. van de Loo, R. Vogels, Q. F. Manson, N. Stathonikos, A. Baidoshvili, P. van Diest, C. Wauters, M. van Dijk, and J. van der Laak. 1399 H&E-stained sentinel lymph node sections of breast cancer patients: The CAMELYON dataset, jun 2018. ISSN 2047217X. URL `http://orcid.org/0000-0003-1554-1291`.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 431–440. IEEE, jun 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298965.

M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood. Data efficient and weakly supervised computational pathology on whole slide images, 2020. ISSN 23318422.

J. N. Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 2010. ISSN 15561380. doi: 10.1097/JTO.0b013e3181ec173d.

L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017. ISSN 2475-9066. doi: 10.21105/joss.00205.

A. Medela, A. Picon, C. L. Saratxaga, O. Belar, V. Cabezon, R. Cicchi, R. Bilbao, and B. Glover. Few shot learning in histopathological images: Reducing the need of labeled data on biological datasets. In *Proceedings - International Symposium on Biomedical Imaging*, volume 2019-April, pages 1860–1864. IEEE Computer Society, apr 2019. ISBN 9781538636411. doi: 10.1109/ISBI.2019.8759182.

T. M. Mitchell. *Machine Learning. Annual Review Of Computer Science.* 1997. ISBN 0070428077.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011. ISSN 15324435.

S. Petushi, F. U. Garcia, M. M. Haber, C. Katsinis, and A. Tozeren. Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. *BMC Medical Imaging*, 2006. ISSN 14712342. doi: 10.1186/1471-2342-6-14.

M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. Transfusion: Understanding Transfer Learning for Medical Imaging. feb 2019. URL `http://arxiv.org/abs/1902.07208`.

K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine. Conditional networks for few-shot semantic segmentation. In *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 2018.

J. Ren, I. Hacihaliloglu, E. A. Singer, D. J. Foran, and X. Qi. Unsupervised Domain Adaptation for Classification of Histopathology Whole-Slide Images. *Frontiers in Bioengineering and Biotechnology*, 7, may 2019. ISSN 2296-4185. doi: 10.3389/fbioe.2019.00102.

X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J. C. Sanchez, and M. Müller. pROC: An open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 2011. ISSN 14712105. doi: 10.1186/1471-2105-12-77.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

A. G. Roy, S. Conjeti, D. Sheet, A. Katouzian, N. Navab, and C. Wachinger. Error corrective boosting for learning fully convolutional networks with limited data. In M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne, editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*, pages 231–239, Cham, 2017. Springer International Publishing. ISBN 978-3-319-66179-7.

A. G. Roy, N. Navab, and C. Wachinger. Recalibrating Fully Convolutional Networks With Spatial and Channel 'Squeeze and Excitation' Blocks. *IEEE Transactions on Medical Imaging*, 38(2):540–549, feb 2019. ISSN 1558254X. doi: 10.1109/TMI.2018.2867261.

H. Seo, M. Badiei Khuzani, V. Vasudevan, C. Huang, H. Ren, R. Xiao, X. Jia, and L. Xing. Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications. In *Medical Physics*, 2020. doi: 10.1002/mp.13649.

A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots. One-shot learning for semantic segmentation. In *British Machine Vision Conference 2017, BMVC 2017*, 2017. ISBN 190172560X. doi: 10.5244/c.31.167.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

C. L. Srinidhi, O. Ciga, and A. L. Martel. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67, jan 2021. ISSN 13618423. doi: 10.1016/j.media.2020.101813. URL `https://www.sciencedirect.com/science/article/pii/S1361841520301778`.

K. Stacke, G. Eilertsen, J. Unger, and C. Lundström. A Closer Look at Domain Shift for Deep Learning in Histopathology. sep 2019. URL `http://arxiv.org/abs/1909.11575`.

X. Sun and W. Xu. Fast implementation of delong's algorithm for comparing the areas under correlated receiver operating characteristic curves. *IEEE Signal Processing Letters*, 21(11):1389–1393, 2014. doi: 10.1109/LSP.2014.2337313.

C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594.

E. van der Spoel, M. P. Rozing, J. J. Houwing-Duistermaat, P. Eline Slagboom, M. Beekman, A. J. M. de Craen, R. G. J. Westendorp, and D. van Heemst. Siamese Neural Networks for One-Shot Image Recognition. *ICML - Deep Learning Workshop*, 7(11):956–963, 2015. ISSN 19454589.

O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

D. Wang, M. Li, N. Ben-Shlomo, C. E. Corrales, Y. Cheng, T. Zhang, and J. Jayender. Mixed-Supervised Dual-Network for Medical Image Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11765 LNCS:192–200, jul 2019a. ISSN 16113349. doi: 10.1007/978-3-030-32245-8_22. URL http://arxiv.org/abs/1907.10209.

S. Wang, D. M. Yang, R. Rong, X. Zhan, and G. Xiao. Pathology image analysis using segmentation deep learning algorithms. *The American Journal of Pathology*, 189(9):1686 – 1698, 2019b. ISSN 0002-9440. doi: https://doi.org/10.1016/j.ajpath.2019.05.007. URL http://www.sciencedirect.com/science/article/pii/S0002944018311210.

Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), June 2020. ISSN 0360-0300. doi: 10.1145/3386252. URL https://doi.org/10.1145/3386252.

Y. Yamamoto, T. Tsuzuki, J. Akatsuka, M. Ueki, H. Morikawa, Y. Numata, T. Takahara, T. Tsuyuki, K. Tsutsumi, R. Nakazawa, A. Shimizu, I. Maeda, S. Tsuchiya, H. Kanno, Y. Kondo, M. Fukumoto, G. Tamiya, N. Ueda, and G. Kimura. Automated acquisition of explainable knowledge from unannotated histopathology images. *Nature Communications*, 10(1): 1–9, dec 2019. ISSN 20411723. doi: 10.1038/s41467-019-13647-8.

A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 8535–8545, 2019. ISBN 9781728132938. doi: 10.1109/CVPR.2019.00874.

J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. ISBN 9781538610329. doi: 10.1109/ICCV.2017.244.

# Acknowledgements

First of all, I want to thank my family for their loving support and patience.

A very special gratitude goes to my advisors who have provided precious guidance and expert advise all along the way.

A special mention to Marco Pozzi, the general manager of Sorint.Tek, who first believed in this project against all odds. He provided constant support through this long but extremely gratifying experience.

I would also like to express my sincere gratitude to all the members of the Computer Vision laboratory of the University of Pavia for their advises, their support as well as the exquisite time and the many coffees (most, if not all, expertly brewed by Prof. Cantoni to fuel the research) we had together.

I thank also Martina Stella and Francesco Maestri whom I had the honour of tutoring for their Master Thesis, and that both greatly contributed to preliminary investigation of conditional FCNs applied to histopathology.