Bachelor's thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Automatic Detection of Metastases in Whole-slide Lymph Node Images Using Deep Neural Networks

Pavlína Koutecká

Supervisor: prof. Dr. Ing. Jan Kybic
Field of study: Cybernetics and Robotics
August 2020

CTU
CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Koutecká Pavlína** Personal ID number: **474383**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Automatic Detection of Metastases in Whole-Slide Lymph Node Images Using Deep Neural Networks

Bachelor’s thesis title in Czech:

Automatická detekce metastáz v histologických obrázcích lymfatických uzlin pomocí hlubokých neuronových sítí

Guidelines:

Develop a method based on deep convolutional neural networks for solving the task of the detection of metastases in whole-slide lymph node images using deep neural networks, as defined in the Kaggle Histopathological Cancer Detection, CAMELYON16 and CAMELYON17 challenges. Get familiar with related work from the literature and develop a baseline solution for patch classification using the ResNet architecture and test it on the data from the Kaggle Histopathological Cancer Detection challenge. Improve and extend these techniques for the full slide segmentation as required by the CAMELYON16 challenge. Implement a slide-level aggregation. Evaluate the performance using the CAMELYON16 criteria. Consider possible improvements using for example attention networks or machine-learning based aggregation.
Time-permitting, consider implementing the patient-level aggregation as defined by the CAMELYON17 challenge and end-to-end learning based on the patient-level weakly labeled data.
Evaluate your results experimentally on the provided datasets and submit your solution to the above mentioned online challenges to compare the performance of your method with state of the art.

Bibliography / sources:

[1] Kaggle. Histopathologic Cancer Detection. https://www.kaggle.com/c/histopathologic-cancer-detection.
[2] The CAMELYON17 challenge. https://camelyon17.grand-challenge.org/.
[3] The CAMELYON16 challenge. https://camelyon16.grand-challenge.org/.
[4] Ehteshami Bejnordi B, Veta M, Johannes van Diest P, van Ginneken B, Karssemeijer N, Litjens G, van der Laak JAWM, and the CAMELYON16 Consortium. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. JAMA. 2017;318(22):2199–2210. doi:10.1001/jama.2017.14585
[5] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, Andrew H. Beck. Deep Learning for Identifying Metastatic Breast Cancer. http://arxiv.org/abs/1606.05718

Name and workplace of bachelor’s thesis supervisor:

prof. Dr. Ing. Jan Kybic, Biomedical imaging algorithms, FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **09.01.2020** Deadline for bachelor thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

prof. Dr. Ing. Jan Kybic Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

© ČVUT v Praze, Design: ČVUT v Praze, VIC

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Acknowledgements

Foremost, I would like to express my sincere appreciation to my supervisor, prof. Dr. Ing. Jan Kybic, for a tremendous amount of patience and great guidance throughout the whole process of making this thesis. He always motivated me to give my best and was great support whenever I ran into trouble.

I wish to show my gratitude to the people whose assistance also helped me with the completion of this thesis. Dr. ret. nat. Jan Hering for his technical advice which helped me to manage the thesis, and doc. MUDr. Tomáš Kučera, Ph.D. for his medical guidance and valuable insights.

Besides these people, I would like to thank Center for Machine Perception for the opportunity to be part of it and use its technical and hardware resources.

Finally, I wish to express my deepest gratitude to my family and friends for providing me with unfailing love and continuous encouragement throughout my whole life and especially the study years. Without you, this accomplishment would not have been possible. Thank you.

Declaration

I hereby declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the Methodical instructions for observing the ethical principles in the preparation of university thesis.

Prague, 14. August 2020

Abstract

Digitisation of cancer recognition in histopathological images is researched topic in recent years, and automated computerised analysis based on deep neural networks has shown potential advantages as a diagnostic strategy. In this thesis, we develop a method for solving the task of automatic metastases detection in whole-slide lymph node images. We are motivated mainly by three existing grand challenges from the histopathologic area: Histopathologic cancer detection challenge by Kaggle, CAMELYON16 and CAMELYON17. First, the baseline solution using ResNet-50 architecture is developed in order of solving the patch classification as defined in Kaggle's challenge. Baseline solution is then extended, and the method is improved to perform the task of tumour segmentation. We propose to use DeepLabV3 architecture and compare it with Fully Convolutional Network and UNet architectures. DeepLabV3 proves to be the most capable model for tumour segmentation. Slide-level and patient-level aggregation are implemented using two classifiers – Random forest and XGBoost. The evaluation shows that their performance is comparable.

The proposed solution is tested and uploaded to the above mentioned grand challenges. For all three challenges, our solution proves to be competitive among other participants.

Keywords: deep learning, machine learning, pathology, breast cancer, classification, segmentation, biomedical imaging, neural network

Supervisor: prof. Dr. Ing. Jan Kybic

Abstrakt

Digitalizace procesu detekce rakoviny v histopatologických snímcích je předmětem výzkumu posledních let a automatizovaná počítačová analýza založená na hlubokých neuronových sítích ukázala potenciální výhody jako diagnostická strategie. V této práci vyvíjíme metodu pro řešení úlohy automatické detekce metastáz v histologických snímcích lymfatických uzlin. Motivací jsou zejména tyto tři existující soutěže z histologické oblasti: soutěž v detekci rakoviny od Kaggle, CAMELYON16 a CAMELYON17. Nejdříve je vyvinuto základní řešení využívající architekturu ResNet-50 pro klasifikaci patchů, stejně jako je definováno v Kaggle soutěži. Toto řešení je poté rozšířeno a metoda je vylepšena tak, aby prováděla segmentaci nádorů. Navrhujeme použití architektury DeepLabV3 a její porovnání s architekturami Fully Convolutional Network a UNet. DeepLabV3 se ukazuje jako nejschopnější model pro segmentaci nádorů. Následná agregace na úrovni snímků a na úrovni pacientů je implementována pomocí dvou klasifikátorů - Random forest a XGBoost. Evaluace ukazuje, že výkon obou klasifikátorů je srovnatelný.

Navržené řešení je otestováno a nahráno do výše uvedených soutěží. Pro všechny tři soutěže se naše řešení ukázalo jako konkurenceschopné.

Klíčová slova: hluboké učení, strojové učení, patologie, rakovina prsu, klasifikace, segmentace, biomedicínské zobrazování, neuronová síť

Překlad názvu: Automatická detekce metastáz v histologických obrázcích lymfatických uzlin pomocí hlubokých neuronových sítí

Contents

List of abbreviations and acronyms

AI	**A**rtificial **I**ntelligence
API	**A**pplication **P**rogramming **I**nterface
ASAP	**A**utomated **S**lide **A**nalysis **P**latform
AUC	**A**rea **U**nder the ROC **C**urve
BACH	**B**re**A**st **C**ancer **H**istology images
BW	**B**lack and **W**hite
CAMELYON16	**CA**ncer **ME**tastases in **LY**mph n**O**des challe**N**ge 2016
CAMELYON17	**CA**ncer **ME**tastases in **LY**mph n**O**des challe**N**ge 2017
CNN	**C**onvolutional **N**eural **N**etwork
CSV	**C**omma-**S**eparated **V**alues
CWZ	**C**anisius-**W**ilhelmina **H**ospital
DCNN	**D**eep **C**onvolutional **N**eural **N**etwork
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
FCN	**F**ully **C**onvolutional **N**etwork
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
FROC	**F**ree-response **R**eceiver **O**perating **C**haracteristic
H&E	**H**ematoxylin and **E**osin
HMS	**H**arvard **M**edical **S**chool

HSI	**H**ue-**S**aturation-**I**ntensity
HSV	**H**ue-**S**aturation-**V**alue
IoU	**I**ntersection **o**ver **U**nion
ITC	**I**solated **T**umor **C**ells
LPON	**L**aboratorium **P**athologie **O**ost-**N**ederland
MGH	**M**assachusetts **G**eneral **H**ospital
MIT	**M**assachusetts **I**nstitute of **T**echnology
ML	**M**achine **L**earning
PANDA	**P**rostate c**AN**cer gra**D**e **A**ssessment
PCam	**P**atch**Cam**elyon
pN-stage	**p**athologic **N stage**
QWK	**Q**uadratic **W**eighted **K**appa
RGB	**R**ed-**G**reen-**B**lue
ROC	**R**eceiver **O**perating **C**haracteristic
RST	**R**ijnstate Ho**S**pi**T**al
RUMC	**R**adboud **U**niversity **M**edical **C**entre
TIFF	**T**agged **I**mage **F**ile **F**ormat
TN	**T**rue **N**egative
TNM	**T**umor, **N**odes, **M**etastasis
TP	**T**rue **P**ositive
TUPAC16	**TU**mor **P**roliferation **A**ssessment **C**hallenge 2016
UMCU	**U**niversity **M**edical **C**entre **U**trecht
WHO	**W**orld **H**ealth **O**rganization
WOTC	**W**ith**O**ut a **T**ime **C**onstraint
WSI	**W**hole **S**lide **I**mage
WTC	**W**ith a **T**ime **C**onstraint
XML	e**X**tensible **M**arkup **L**anguage

Chapter 1

Introduction

According to the World Health Organization (WHO), breast cancer is the most frequently diagnosed cancer in Czech women. In 2018, there were 7 436 newly diagnosed patients with breast cancer, which accounts for 25 % of all diagnosed women cases, and 1 580 patients died from this disease [1]. In the present research, histopathologic image analysis is the standard method applied in the clinical practice to diagnose breast cancer. Even though the prognosis for patients diagnosed with breast cancer is usually good, the survival rate declines if cancer metastasises [2]. That makes recognising the metastases in lymph node sections one of the most important prognostic factors.

In the process of histology image analysis for cancer diagnosis, pathologist standardly visually observes the tissue, its distribution and regularities of cell shapes. After that process, pathologist decides whether there are some cancerous tissue regions and determines the malignancy level [3]. However, this diagnostic procedure is time-consuming and small metastases are very difficult to detect even for experienced pathologists [4]. Fortunately, computer-based image analysis has become a rapidly expanding field within the past few years [3] and whole-slide scanners are now commonly used for digitising glass slides at high resolution. This process partially allows automation of the histopathologic image analysis for cancer diagnosis, but there is still a great potential to improve and fully automate this task and help the pathologists to reduce their workload.

1.1 Motivation

Considering the recent improvements in the field of machine learning (ML) algorithms and whole-slide imaging, the task of fully automated analysis of histopathologic images started to be more approachable than ever. The availability of many digitalised whole-slide images resulted in increasing interest of the medical image analysis community, and numerous histopathologic imaging challenges in cancer diagnosis arose lately to improve the efficiency and accuracy of this task. Commonly, a clinically relevant task, like cancer detecting or grading, predicting prognosis or identifying metastasis, is defined by organisers, who provide a sufficiently comprehensive and diverse collection of data called *dataset*. Participants use the dataset to develop an ML algorithm appropriate for the specified task, which is subsequently evaluated by the challenge organisers. Typically, the submission deadline follows a workshop or conference, where participants with best-scored algorithms discuss their approaches and solutions. This procedure led to quick progress in automated histopathology image analysis and allowed a meaningful comparison of algorithms with promising results.

Many successful medical imaging challenges were organised in recent years. In histopathology field, it was, for example, breast cancer histology images challenge (BACH) [5], tumour proliferation assessment challenge (TUPAC16) [6] and ongoing prostate cancer grade assessment challenge (PANDA) [7]. This thesis is mainly motivated by three existing challenges – Histopathologic cancer detection challenge by Kaggle [8–10], Cancer metastases in lymph nodes challenge 2016 (CAMELYON16) [10] and Cancer metastases in lymph nodes challenge 2017 (CAMELYON17) [11].

1.1.1 Histopathological cancer detection challenge by Kaggle

This challenge[1] aims to create an algorithm to identify metastatic cancer in small image patches taken from large digital pathology scans. The data for this challenge is a slightly modified version of the PatchCamelyon (PCam) dataset, which was derived from the CAMELYON16 dataset [8–10]. Kaggle runs this competition since 2019.

The task of this competition is very straight-forward – a clinically-relevant task of the metastasis detection is presented as a binary image classification task. Models for this task are easily trainable in a couple of hours, and its performance is evaluated on the area under the receiver operating characteristic (ROC) curve. That makes this competition an excellent resource for fundamental research on topics as digital pathology, automatic tumour detection and whole-slide imaging.

[1] Available at `https://www.kaggle.com/c/histopathologic-cancer-detection/`.

1.1.2 CAMELYON16 challenge

The goal of this challenge[2] is to develop an algorithm for automated detection of metastases in whole-slide images of lymph node sections [10]. Two medical centres in the Netherlands provided an extensive dataset. This competition consists of two tasks [10]:

1. **Slide-based evaluation:** Algorithms are evaluated for their ability to discriminate every whole-slide image as either containing or lacking metastases. For the evaluation, the ROC curve is used.

2. **Lesion-based evaluation:** Algorithms are evaluated for their ability to identify individual micro-metastases and macro-metastases in whole-slide images. For the evaluation, the free-response receiver operating characteristic (FROC) curve is used.

Different evaluation metrics for every task resulted in two independent algorithm rankings. Challenge was opened to new entries only in 2016.

1.1.3 CAMELYON17 challenge

The goal of this challenge[3] is, same as in CAMELYON16 challenge, to develop an algorithm for automated detection of metastases in whole-slide images of lymph node sections. Compared to the CAMELYON16 challenge, the dataset is notably extended – data were provided by five medical centres. Challenge is open to new entries since 2017.

The task of the competition developed from slide-level analysis to patient-level analysis. In this challenge, artificial patients are created. There are five slides provided for each patient, and every slide corresponds to one lymph node section. This approach combines the detection and classification of metastases in multiple lymph node slides, assigned to one patient, into one outcome corresponding to the patient pN-stage, closely described in Chapter 2 [11]. This brings the task closer to clinical practice. Usually, many lymph node slides are prepared for the patient, and aggregating results of more slides is a necessary step to involve an algorithm for automated detection of metastases in daily medical practice. For the evaluation of the results, the five-class quadratic weighted kappa is used.

[2] Available at `https://camelyon16.grand-challenge.org/`.

[3] Available at `https://camelyon17.grand-challenge.org/`.

1.2 Goals

The main focus of this work is to develop a method for solving the task of the detection of metastases in whole-slide lymph node images using deep convolutional neural networks (DCNNs), as defined in the Kaggle Histopathological Cancer Detection, CAMELYON16 and CAMELYON17 challenges. To achieve that, it is necessary to get familiar with related work from the literature and current state-of-the-art methods.

In the following chapters, a baseline solution for patch classification using deep neural networks (DNNs) will be created and tested on the data from the Kaggle Histopathological cancer detection challenge. This technique will be improved, and patches will be aggregated to provide the full slide segmentation and slide-level classification as required by the CAMELYON16 challenge. The patient-level aggregation will extend the slide-level solution as required by the CAMELYON17. Both slide-level and patient-level results will be evaluated experimentally on provided datasets, and the final solution will be submitted to the CAMELYON17 challenge to compare the performance of our method with state-of-the-art.

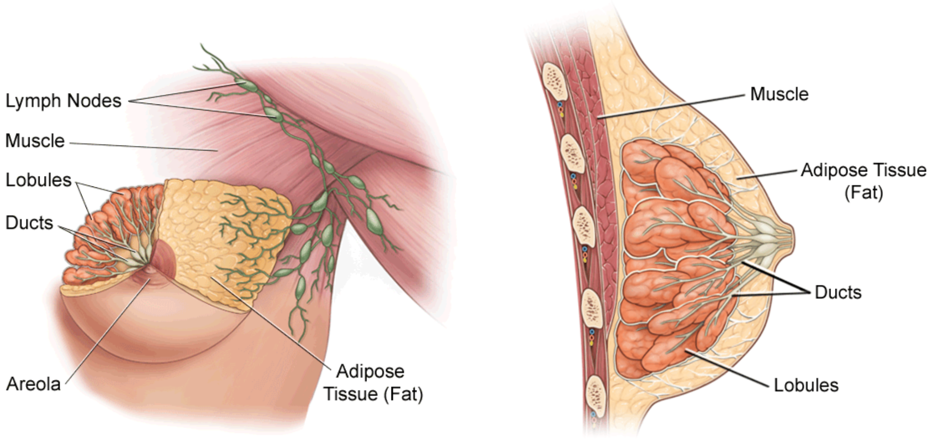
Moreover, some parts of this work will be expanded with additional information from the medical field to analyse the problematics comprehensively, localise weaknesses of our method and provide the reader with a better understanding of the medical background.

Chapter 2

Medical background

2.1 Anatomy of the breast

As different parts of the breast will be referenced repeatedly, a better understanding of its anatomy will help us deal with the task. A healthy female breast, shown in Figure 2.1, consists of 15 to 20 globes of glandular tissue, called *lobes* [13]. Each of the lobes is made up of smaller *lobules* – glands that produce milk. These lobules are arranged in clusters, similarly as grapes, and connected by milk ducts, which carry the milk to the nipple [14]. Lobes are supported by the fibrous connective *stroma* forming a latticed framework, travelling through the breast and inserting into the dermis. That provides remarkable mobility while still supporting the breast [13, 15]. The remainder of the breast is formed by fat cells called *adipose tissue*, which fills the space between the lobes and fibrous stroma. Breast cancer typically starts to form



(a) : Front view

(b) : Side view

Figure 2.1: Detailed illustration of the adult female breast anatomy, taken and edited from [12].

in the structure of lobes and ducts [16].

2.2 Lymphatic system

The lymphatic system, running throughout the entire body, together with other lymphoid organs and tissues (the spleen, thymus, tonsils and other tissues), provides a structural basis of the immune system and plays a crucial role in body protection [17]. Main functions of the lymphatic system are to provide a return route of the lymph into the blood system and defend the body against infection [18].

The lymphatic system consists of three main parts [17]:

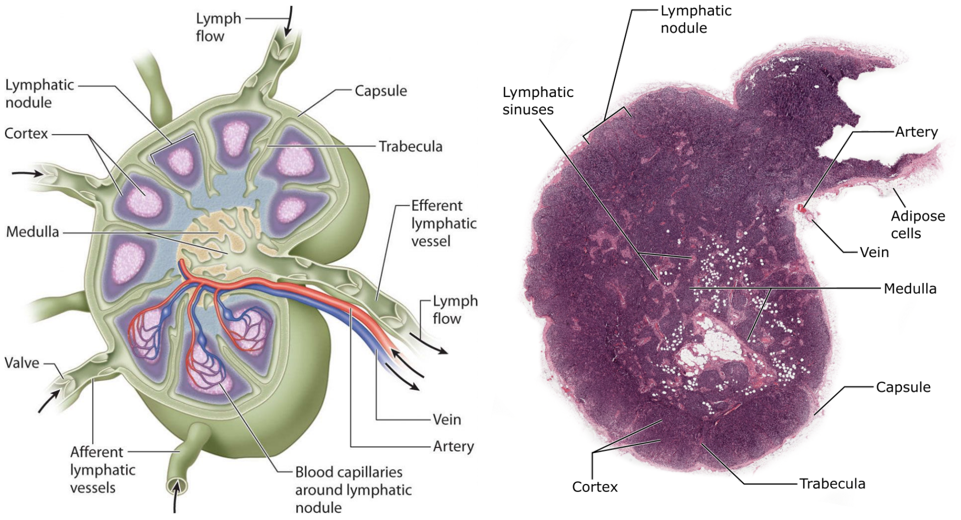
1. a network of *lymphatic vessels*
2. a fluid inside of the vessels called *lymph* – colourless fluid located between the cells in all body tissues, that contains white blood cells called lymphocytes and circulates throughout the lymphatic system
3. *lymph nodes* – cleanse the flowing lymph

2.2.1 Lymph nodes

Lymph nodes are small, bean-shaped glands composed of lymphatic tissue, widely distributed along the lymphatic routes [19]. Simplified illustration of the lymph node is shown in Figure 2.2. Clusters of lymph nodes nearest to the breast are located in the armpit (called *axillary lymph nodes*), above the collarbone and in the chest [14]. Axillary lymph nodes provide a majority of the drainage basin for the breast. According to [15], approximately 97 % of the breast lymphatics drain to the axillary lymph nodes, the remaining 3 % drain to the mammary lymph nodes.

Each node is covered by a fibrous capsule that extends inside the tissue a strand called *trabecula*. The lymph node tissue is differentiated into two distinct regions – the *cortex*, located under the capsule, and the *medulla* [17]. The most important formations of the cortex and medulla are *lymphatic nodules*. Each nodule contains lymphocytes, and during an immune response, these nodules develop into centres fighting the infection. Also, a series of *lymphatic sinuses*, filled with lymph flowing from lymphatic vessels to the nodule, are scattered throughout the node [17, 20].

The primary function of the lymph node is to filter flowing lymph circulating through the lymph vessels – all lymph formed in tissues must always pass at least one node before entering back the blood circulation [14, 18, 19]. Lymph is very similar to blood plasma – it contains lymphocytes and macrophages



(a) : Illustrative lymph node image **(b) :** Histopathological lymph node image

Figure 2.2: Detailed illustration of the lymph node anatomy compared to authentic histopathological lymph node image. Illustration taken from [21], histopathological image taken from CAMELYON dataset [22].

cells, but it may also contain microorganisms, waste products and other undesired substances from the tissue [17]. Lymph nodes are responsible for trapping these particles and filtering various pathogens found within the body – macrophages and lymphocytes attack and kill them.

Since the lymph nodes play a central role in filtering undesired substances from the cells, it makes them vulnerable to cancer. As was said in Section 2.1, breast cancer typically starts to form in the structure of lobes and ducts [16]. Cancerous cells located in the lobes or ducts start to spread from the tissue via lymph, and they may be trapped in a lymph node, where they start to proliferate. That makes axillary lymph nodes the first place where breast cancer is likely to spread, and recognising metastases in them is one of the most important prognostic factors in breast cancer [14, 19].

2.3 Digital pathology

Digital pathology is a rapidly expanding sub-field of pathology that allows conversion of the classical glass slide, extracted by a pathologist, into a digital image called *whole-slide image* (WSI) that can be uploaded to a computer for viewing and complete electronic management [23]. It represents a fundamental change in the way pathological specimens are viewed. Nowadays, in clinical diagnosis practice, rapid adoption of digital pathology is happening, because manual pathology examination via microscope is time-consuming, tedious and not effective [11, 23, 24]. Compared to that, digital pathology has many

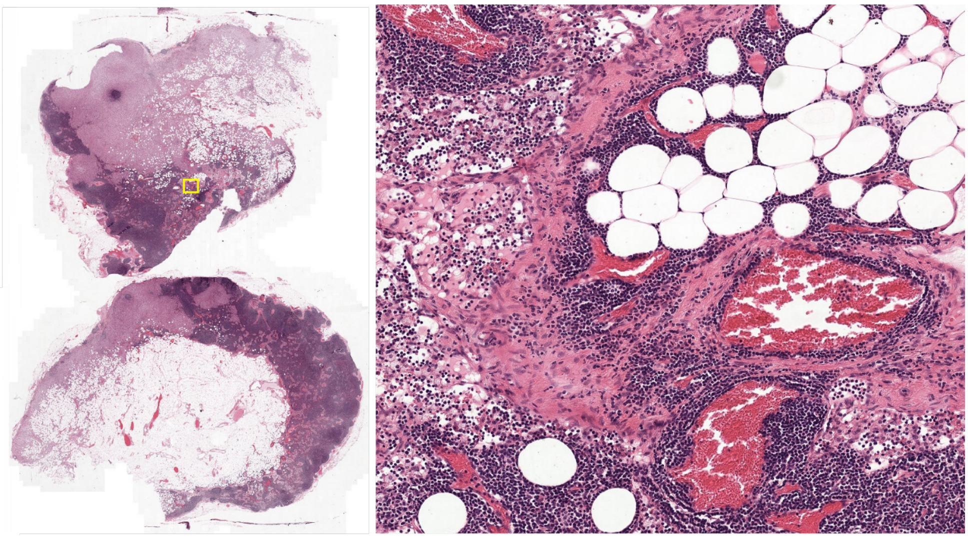


Figure 2.3: The low-resolution WSI of lymph node section stained with H&E compared to the zoomed detail. Cell nuclei (blue), red blood cells (red), extracellular material and other cell bodies (pink), adipose cells and air spaces (white). Tissue sample taken from CAMELYON dataset [22].

advantages. For example, the permanence of digital files, reproducibility, ability to access all patient's slides at any time, annotate them, make special visualisations or draft reports. Furthermore, with the recent improvements of whole-slide scanners, digital pathology is more approachable, and most of the slides started to be stored in high-resolution digital formats. This process, called *whole-slide imaging*, allows a complex computerised slide analysis, and histopathological examination moved from viewing glass slides under the microscope to analysing images on the computer monitor [23, 24].

2.3.1 Whole-slide imaging

Whole slide imaging includes the digitisation of the entire histology slide. The process consists of five main parts: slide preparation, scanning, storing, editing and displaying [24].

Appropriate slide preparation is crucial for the successful whole slide imaging procedure. Firstly, the tissue intended for observation is carefully excised, fixed in formalin and infiltrated with paraffin wax. Then, a micrometres thin slices of the tissue are cut. These tissue slices are placed on glass and stained [25]. For this purpose, different stains are used. Most widely used in medical diagnosis is the *hematoxylin and eosin* (H&E) stain. As shown in Figure 2.3, blue colour of the cell nucleus is obtained by hematoxylin, pink colour of the cell membrane and extracellular structure showing a general overview of the tissue is obtained by eosin, and adipose tissue appears as empty space [26].

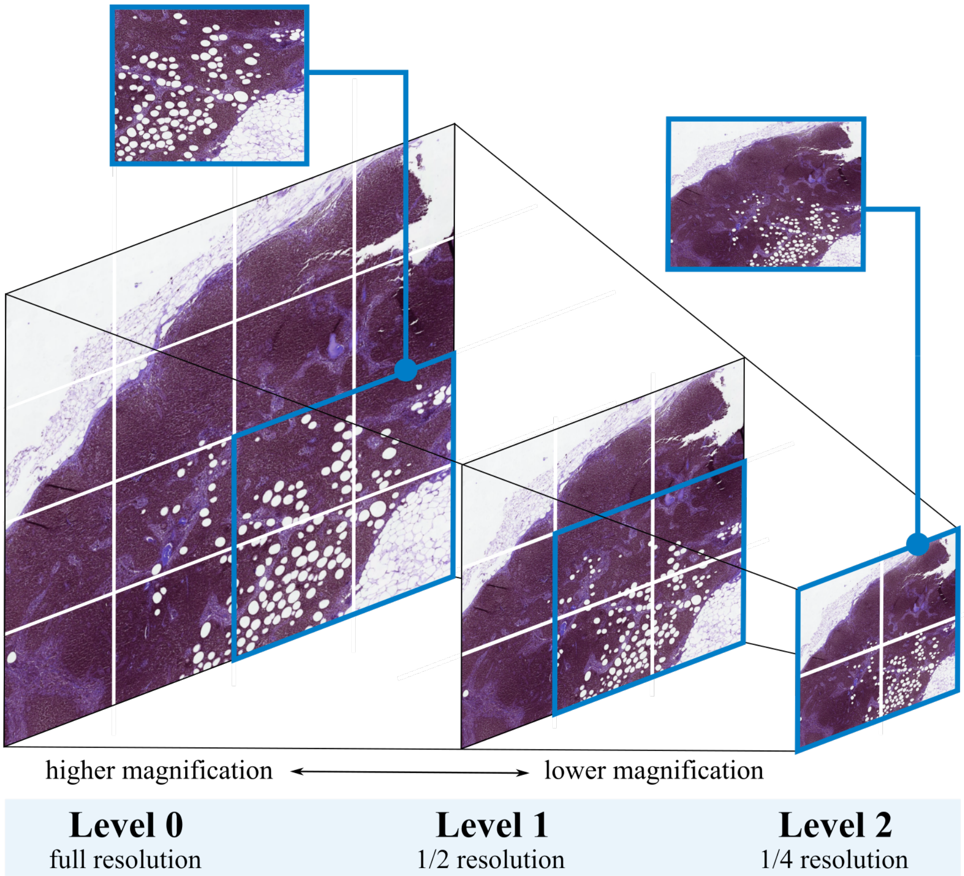


Figure 2.4: The multi-resolution pyramid structure of a WSI. Images at various magnifications are presented as series of tiles – higher resolution means more tiles. The full resolution is presented as level 0, and every following level has a half resolution. With the same amount of tiles, lower level number means a more detailed view. Tissue sample taken from CAMELYON dataset [22].

Whole-slide scanners provide scanning of the slide tile by tile. Captured tiles with tissue sections are then stored as a series of tiles and digitally assembled to generate an image of the entire slide [24]. The slide must be captured at sufficiently high resolution – standardly the ×20 or ×40 magnification is used – to copy the workflow achieved with a manual microscope observation. Although scanning magnification is determined by used objective, resolution of the digitalised image is defined as a minimum distance at which two distinct objects can be identified as separate events. It is typically expressed in units of µm per pixel. A standard WSI scanned at ×40 magnification has a resolution of approximately 25 µm per pixel [24].

Despite the image compression methods, a single WSI's file size often exceeds units of GB with an image size of approximately 200000 × 100000 pixels on the highest resolution level. That makes almost impossible viewing entire slide at high resolution. However, when a pathologist examines tissue at high magnification, only a small field of view is visible on the monitor, so the image does not need to be loaded entirely. For this purpose, slide is stored in a *multi-resolution pyramid structure* as illustrated in Figure 2.4. WSI scanned at, for example, ×40 magnification is accompanied by the same image downsampled at ×10, ×2.5 and ×1.25 magnification, and these images are usually embedded within a single file [24].

Editing and displaying slides using standard image tools and libraries are often a challenge. However, specialised image-viewers are currently developed to improve pathologist's routine with WSI navigating, viewing and annotating. These systems are usually distributed along with the scanner and adapted to the user's needs. Unlike in the clinical practise, in research applications, direct access to the WSI files is often preferred, and numerous tools have been developed to enable it [24].

2.4 Breast cancer

Breast cancer is a type of cancer that begins in the breast and almost always affects women. Cancer cells usually form a tumour, that can be observed by the doctor or felt as a lump. The term 'breast cancer' is used when abnormal cells begin to grow out of control and develop a malignant tumour [16]. It may invade surrounding healthy cells and possibly spread to other parts of the body.

A *tumour* is a mass of tissue created when cells fail to follow normal controls of cell division and start to multiply without control [17]. In breast, we might find two types of tumours [16]:

- *benign tumours* – strictly local, not aggressive toward surrounding tissue
- *malignant tumours* – cancerous, aggressive, invade their surroundings

As the benign tumour is non-cancerous and its cells remain compacted, it is usually not removed. In contrast, if the malignant tumour is found, the doctor performs a diagnostic test to determine the severity of the tumour and plans the treatment [16, 17].

Malignant tumours are dangerous mainly because of the cells that form the tumour. They tend to break away from their primary source and travel to other parts of the body, usually through the lymphatic system, where they form a secondary tumour. This process is called *metastasis* [17].

2.4.1 Diagnosis and staging

Determining the severity of the tumour and extent of metastases is key to deciding on the patient's prognosis and future treatment. An internationally

Category	Size
Macro-metastasis	Larger than 2 mm
Micro-metastasis	Larger than 0.2 mm and containing more than 200 cells, but not larger than 2 mm
Isolated tumour cells	Single tumour cells or a cluster of tumour cells not larger than 0.2 mm or less than 200 cells

Table 2.1: Rules for assigning single cells or clusters of metastasized tumour cells to a metastasis category, taken from [11].

accepted strategy to classify the extent of cancer is the *tumour, nearby lymph nodes, distant metastasis* (TNM) *staging system* [27]. This system is widely adopted by doctors for various cancer types. In breast cancer, it takes into account the size of the tumour (T-stage), whether cancer has spread to nearby lymph nodes (N-stage) and whether the tumour has metastasized to other parts of the body (M-stage) [11, 27].

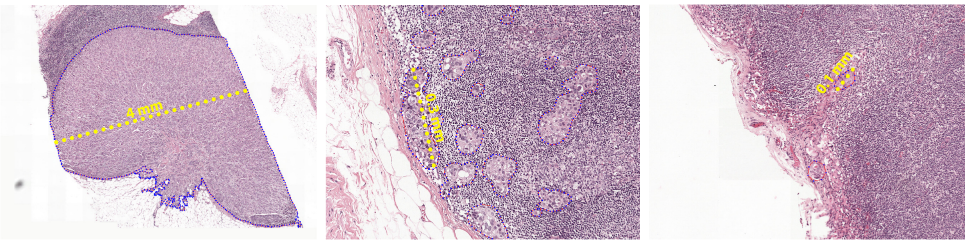
As was said in Section 2.2, axillary lymph nodes usually are the first location breast cancer metastasizes to. As a result of this, the first step in determining the cancer stage is detecting metastases in regional lymph nodes, which is almost always assessed with the help of *sentinel lymph node biopsy*[1] [11, 17]. In this procedure, a blue dye and/or radioactive tracer is injected near the tumour. As this substance starts to spread, first lymph nodes reached by it are marked as sentinel nodes. With this knowledge, the doctor can identify the most likely metastasized nodes to which the tumour drains. Subsequently, these nodes are excised, adjusted to the WSI format and taken for further pathologic examination [11, 22]. If the sentinel nodes contain cancer, additional nodes may be examined to understand better how far the disease has spread [14].

During the microscopic assessment, the pathologist screens the WSI to find out whether it contains tumour cells or not. If a cluster of metastasized tumour cells is found, depending on its size, it may be classified into one of three categories: isolated tumour cells (ITC), micro-metastases or macro-metastases [11, 13, 22]. Detailed size criteria for each category provides Table 2.1 and

Assignning the pN-stage

After the WSIs observation and tumour-size classification according to the found metastasis clusters is done, a pathological N-stage (pN-stage) is assigned to the patient. This categorization is based mainly on metastasis size and

[1] Screening procedures like mammography are vital only for the early detection. However, most breast cancers patients are diagnosed after symptoms have already appeared, and more radical methods are needed.



(a) : Macro-metastasis **(b) :** Micro-metastasis **(c) :** Isolated tumour cells

Figure 2.5: Representative samples of different types of breast cancer metastases size, taken from [22].

pN-stage	Slide labels
pN0	No micro-metastases or macro-metastases or ITC found
pN0(i+)	Only ITC found
pN1mi	Micro-metastases found, but no macro-metastases found
pN1	Metastases found in 1 – 3 lymph nodes, of which at least 1 is a macro-metastasis
pN2	Metastases found in 4 – 9 lymph nodes, of which at least 1 is a macro-metastasis

Table 2.2: pN-stages used in the CAMELYON17 challenge, taken from [11].

a number of nodes invaded by metastases. However, some categories are dependent on the anatomical location of the lymph nodes, extra molecular tests or a big number of lymph nodes to observe [11, 22]. Considering that, a simplified version of the pN-staging system[2] indicated in Table 2.2 is used in the CAMELYON17 challenge to keep the dataset size within reasonable limits [11, 27].

2.4.2 Treatment

The options of treatment depend on the obtained TNM stage and other factors, like age, family history or general health of the patient [16]. A higher number of the assigned stage means worse prognosis [16, 27]. In clinical practice, the treatment procedure differs from patient to patient, but there are some general patterns repeated for patients with a similar diagnosis:

- For **early-staged** patients, which make up approximately 60 % of all breast cancer patients, the prognosis is very positive – approximately 98 % of them will survive for five years [2]. They usually undergo surgery sometimes followed by radiation [16].
- For patients with **locally-advanced stage**, which make up approximately 33 % of all breast cancer patients, the prognosis is worse – around

[2] For a full listing, refer to [27].

84 % of them will survive for five years [2]. These patients also undergo surgery preceded and followed by radiation [16].

- For patients with **advanced or metastatic stage**, which make up approximately 5 % of all breast cancer patients, the prognosis is the worst – roughly 24 % of them will survive for five years [2]. Taking care of these patients usually involves systematic treatment regimens like hormone therapy, chemotherapy or radiation [16].

Chapter 3

Data description

To accurately train DL models and evaluate their performance, large and well-annotated datasets are needed. That is a problem, especially in the medical field, where sharing the data is often difficult. In the context of CAMELYON16 and CAMELYON17 challenge, a public dataset with numerous annotated WSIs of lymph node sections was released [22]. That opened up the research question of detecting metastases in lymph node tissue to a large community, which would normally not have access to required datasets.

3.1 CAMELYON dataset

This dataset was collected at multiple Dutch medical centres to ensure the slide heterogeneity [22]. It contains 399 WSIs for the CAMELYON16 and 1 000 WSIs for the CAMELYON17, which results in unique 1 399 WSIs in total and approximately three terabytes of image data. Part of the dataset with a reference, called *train dataset*, was released to allow participants to build their algorithms. The rest of the dataset, called *test dataset*, was released without a reference to enable participants to submit their algorithm output for evaluation on a predefined set of metrics [22]. The whole dataset is publicly available at the CAMELYON17 website[1].

3.1.1 Data selection

In total, five medical centres in the Netherlands collected the data – Radboud University Medical Centre (RUMC), University Medical Centre Utrecht (UMCU), Rijnstate Hospital (RST), Canisius-Wilhelmina Hospital (CWZ) and Laboratorium Pathologie Oost-Nederland (LPON) [22]. Low-resolution

[1] Available at `https://camelyon17.grand-challenge.org/Data/` after registering in the competition.

	Total WSIs		**Metastases**		
Centre	Train	Test	None	Micro	Macro
RUMC	170	79	150	51	48
UMCU	100	50	90	26	34
Total	270	129	240	77	82

Table 3.1: WSI-level characteristics for the CAMELYON16 part of the dataset, taken and edited from [10, 22].

	Total WSIs		**Metastases (Train)**			
Centre	Train	Test	None	ITC	Micro	Macro
CWZ	100	100	64	11	10	15
LPON	100	100	64	7	4	25
RST	100	100	60	7	22	11
RUMC	100	100	60	8	13	19
UMCU	100	100	75	2	8	15
Total	500	500	323	35	57	85

Table 3.2: WSI-level characteristics for the CAMELYON17 part of the dataset, taken and edited from [11, 22].

example of a digitised slide from each centre can be seen in Figure 3.1.

We can associate two stages of data acquisition in CAMELYON16 and CAMELYON17 challenge. Within the CAMELYON16 challenge, only data from two centres (RUMC and UMCU) were collected, no slides with only ITC were included [10]. During the CAMELYON17 challenge, data were collected from all five centres, slides containing only ITC were also included [11, 22]. The distribution of slides in CAMELYON16 and CAMELYON17 challenge can be found in Tables 3.1 and 3.2.

3.1.2 Data digitisation and labelling

Data selection was followed by the process of digitisation. As scans were taken in various centres using different tissue preparation protocols, staining procedures and scanning equipment, the data were entered with scan and H&E staining procedure variability [22]. Generally, in pathology, scan's appearance differs from centre to centre. Using DL models trained on slides from only one centre may lead to issues with a model's ability to generalise [28]. Organisers of the CAMELYON challenge included slides from five centres to manage this issue and ensure sufficient data diversity leading to greater robustness of submitted algorithms [22].

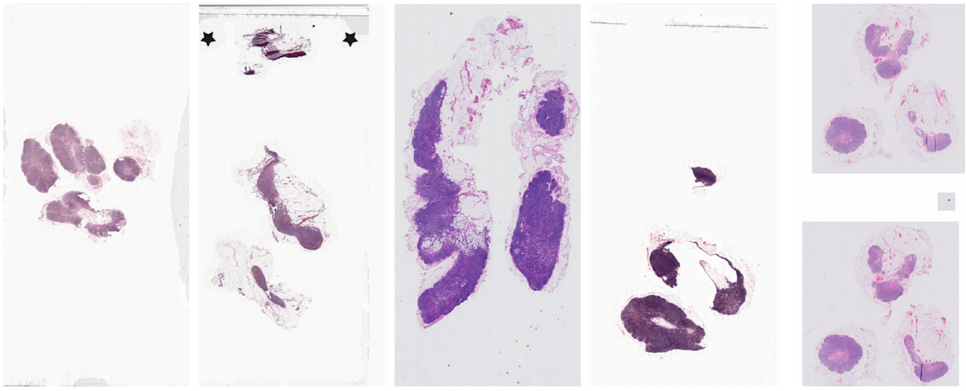


Figure 3.1: Low-resolution examples of WSI from each of the five centres providing data, taken from [22].

	Total patients		**Stages (Train)**				
Centre	Train	Test	pN0	pN0(i+)	pN1mi	pN1	pN2
CWZ	20	20	4	3	5	7	1
LPON	20	20	6	2	2	7	3
RST	20	20	4	2	6	5	3
RUMC	20	20	3	2	4	8	3
UMCU	20	20	8	2	4	3	3
Total	100	100	25	11	21	30	13

Table 3.3: Patient-level characteristics for the CAMELYON17 part of the dataset, taken and edited from [22].

Slides from all five centres were converted to a generic tagged image file format (TIFF). After that, at least one experienced pathologist examined each WSI and labelled it using the *slide-level labels* indicating the largest metastasis located within the WSI. Additionally, all 399 WSIs belonging to the CAMELYON16 part of the dataset and 50 WSIs from the CAMELYON17 part of the dataset (10 WSIs per every medical centre) were exhaustively annotated [22]. These precisely annotated borders around metastatic tissue, called *lesion-level annotations*, were provided simultaneously with the dataset as extensible markup language (XML) files containing coordinates of contours vertices at the highest resolution level of the image. Some of the slides involve more tissue sections of the same lymph node. In that case, only one of them was exhaustively annotated. These slides are indicated in a text file attached to the dataset [11].

After the slide-level labelling process, to simulate clinical conditions, so-called *artificial patients* were created from all slides in the CAMELYON17 part of the dataset. Each artificial patient consists of exactly five lymph node slides taken from one medical centre [22]. In clinical practice, there are many

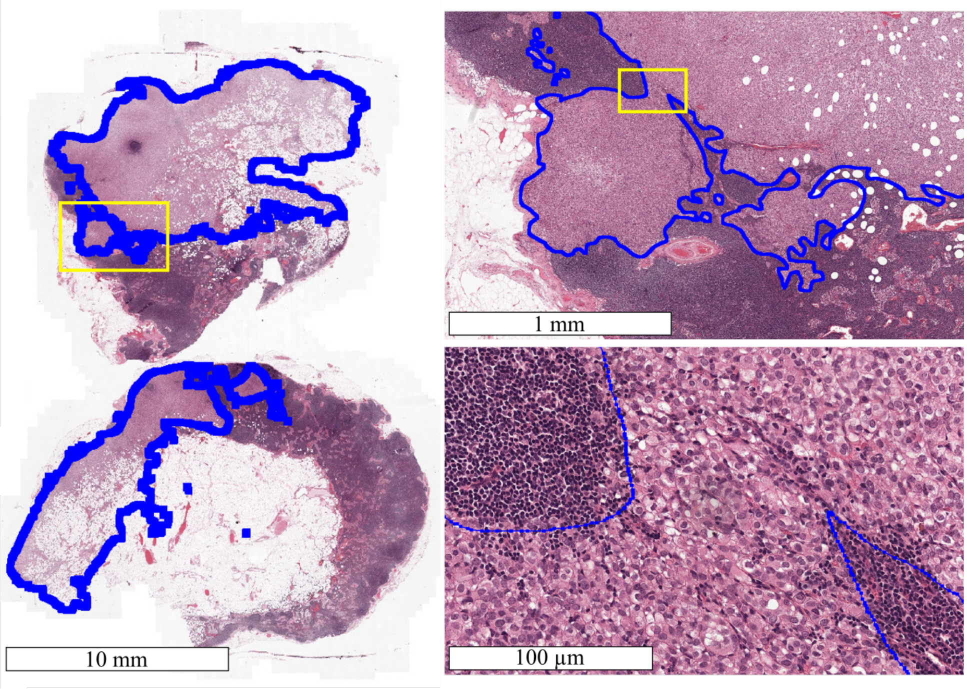


Figure 3.2: Illustration of a WSI visualized by the ASAP software at multiple magnifications demonstrating the zooming workflow performed by pathologists. Blue curves, loaded from the attached XML file, were drawn by a pathologist and highlight borders of found tumours. Tissue sample taken from CAMELYON dataset [22].

lymph nodes per each real patient. Unfortunately, the size of the CAMELYON dataset would grow beyond acceptable limits. Therefore, all slides from real patients were heavily mixed and assembled into artificial patients. Then, slides of every artificial patient were examined by an experienced pathologist to assess the *patient-level labels* [22]. Distribution of these labels across the medical centres describes Table 3.3. Both slide-level labels and patient-level labels for the train part of the CAMELYON dataset were provided to able participating algorithms to perform fully automated pN-staging.

3.1.3 Tools for data usage

Accessing WSI using standard image tools is often a challenge because these tools usually work with images, that can be easily uncompressed [22]. Unfortunately, the size of the uncompressed WSI may be several gigabytes. Therefore, special tools were developed to manipulate images like this. For operating with WSIs from CAMELYON dataset, mainly two tools are recommended by the organisers – OpenSlide library and ASAP software [11, 22].

OpenSlide is a C library providing a simple interface to read WSIs of various formats. Python and Java application programming interface (API) is also available [29]. *Automated slide analysis platform* (ASAP) is a publicly available software package for viewing the WSIs, their annotations a and algorithmic results. It was released simultaneously with the CAMELYON16 challenge by the challenge's organisers [30]. Using this tool, the slide might be explored via a Google Maps-like interface, and if the lesion-annotation is provided, it can also be loaded. Example of a WSI with annotated tumour visualized by the ASAP software illustrates Figure 3.2.

3.2 PatchCamelyon dataset

Dataset used in the Histopathologic cancer detection challenge by Kaggle is a slightly modified version of the PCam dataset[2]. Original PCam dataset, due to the probabilistic sampling strategy, contains duplicate patches. Kaggle removed them and provided participants with the edited dataset maintaining the same data and splits as the PCam benchmark [8].

PCam is a huge, image classification dataset providing over 327 000 small patches of size 96 × 96 pixels extracted from the CAMELYON dataset to simplify the task of metastasis detection [8]. Each patch is annotated with a binary label – a positive label indicates that the patch's central 32 × 32 pixel region contains at least one pixel of metastasis, a negative label indicates the opposite. If the tumour tissue is located in the outer region of the patch, it does not count as a positive label and it only provides additional information about the surrounding tissue [9,31]. Example of both positively and negatively labelled samples are illustrated in Figure 3.3.

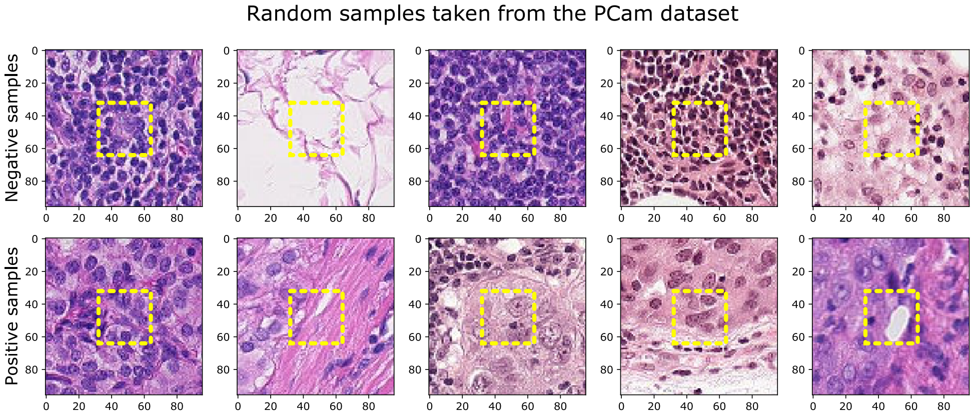


Figure 3.3: Randomly extracted patches with highlighted central 32 × 32 pixel region and both positive and negative labels from the PCam dataset. Patches taken from [31].

[2] Available at `https://github.com/basveeling/pcam`.

The original PCam dataset is divided into a training part, consisting of 262 144 patches, validation part and test part, both consisting of 32 786 patches [9, 31]. The edited PCam dataset from Kaggle is divided into a training part, consisting of 220 025 patches and test part, consisting of 57 458 patches. All splits have a balanced number of positive and negative labelled samples and follow the initial train/test split from the CAMELYON dataset. These patches were sampled by iteratively choosing a WSI and selecting a patch with or without a metastatic tissue with probability p adjusted to keep the balance. Patches containing nothing but background were filtered out [9, 31].

All patches in the dataset come as TIFF formatted images. An additional comma-separated values (CSV) file is attached to provide ground-truth for patches in the train part of the dataset [8]. Also, extra CSV file, describing from which CAMELYON WSI were patches extracted, is attached. However, this information is not used in training, nor evaluating [9, 31].

Chapter 4

Related work

Over the past several decades, there have been significant advances in the field of breast cancer recognition from histopathological images. In the past, the breast tissue specimens were examined for cancer using a microscope, which carried many difficulties, for example, the fragility of observed glass slides or the need for specialised storage rooms [32]. With the growing size of cancer cases and inconsistent results across different pathologists, an automated objective solution for examining tissue slides started to be highly desirable.

The possibility of digitising glass slides opened the door to computer-based histopathology image analysis, called *digital pathology*, already in the 1980s. However, the poor scanner's quality and limited memory prevented it from being used in clinical practice [33]. Most significant advances in digital pathology were made in the late 1990s by Wetzel and Gilbertson – they developed the first automated WSI system [32, 33]. With the advent of whole-slide imaging, WSIs started to be scan and load into a computer, and pathological laboratories in clinical practice are currently undergoing an extensive transformation toward a fully digital workflow [34].

As the computing power and whole-slide imaging adoption grow, various WSI datasets are available. Along with recent advances in artificial intelligence (AI) tools, which provided state-of-the-art results in many fields, significant progress in the application of deep learning (DL) to automated histopathology analysis was made.

The most successful DL tool for image analysis is a *convolutional neural network* (CNN) [35]. CNNs were applied to medical image analysis already in 1995 by [36]. Despite the promising result, the area of neural networks application in medical image analysis was not significantly investigated until various techniques for efficient deep neural network (DNN) training were developed in the past decade. Since then, CNN methods have approached many histopathological problems. For example, nuclei segmentation [37], signet ring cell detection [38] and also lymph node metastasis detection [10].

4.1 Grand challenges

Initially, applications of DL methods in histopathology appeared only at workshops and conferences. Then, since 2015, the amount of published papers in journals started to grow rapidly [35]. That is linked to the increasing number of *grand challenges*[1] on the topic of histopathological imaging. These challenges encourage the medical image community and researchers to collectively work on various histopathological image analysis tasks using DL based solutions by providing comprehensive labelled WSI datasets. Tasks are usually clinically relevant, and, as can be seen from the results of many grand challenges, the quick development of digital pathology analysis is strongly improved by techniques that challenge's participants present [35]. Grand challenges also allow a standardised comparison of algorithms – in scientific literature, authors present results on their own, often using their own evaluation metrics and data, which make presented algorithm uncomparable with related work [40].

As was said in Chapter 1, many successful histopathological grand challenges were recently organised. Some of the most significant from the field of breast cancer recognition are TUPAC16 [6] with the tumour proliferation scores prediction, CAMELYON16 [10] with the lymph node metastasis detection and BACH 2018 [5] with automatic classification of breast cancer in histology images. This work focuses mainly on three existing breast cancer recognition challenges – Histopathological cancer detection challenge by Kaggle, CAMELYON16 and CAMELYON17. Following sections will provide a brief overview of the state-of-the-art in each of them.

4.1.1 Histopathological cancer detection challenge by Kaggle

The aim of this challenge is to create an algorithm to identify metastatic tissue in histopathological scans of lymph node sections [8]. As organisers prepared small image patches from CAMELYON dataset and collected them into the PCam dataset, the task stays quite straight-forward – a binary image classification problem.

Submitted algorithms are sorted by their performance using the AUC score. The AUC score ranges from 1.000 to 0.308 for all 1 149 participants[2]. As the challenge does not require additional documentation of submitted algorithms, there is no way to describe the winning methods in more detail. According to

[1] The term grand challenge represents an important but very challenging problem set by some institution with the intention of encouraging possible solutions [39]. Grand challenges in the field of medical image analysis are available at `https://grand-challenge.org/`.

[2] According to the challenge's leaderboard, available at `https://www.kaggle.com/c/histopathologic-cancer-detection/leaderboard`.

challenge's discussion[3] and shared notebooks[4], frequently used CNN models are, for example, DenseNet169 [41] or ResNet-9 [42]. Many participants use data augmentation too.

4.1.2 CAMELYON16 challenge

The aim of this competition was to investigate the potential of ML algorithms for lymph node metastasis detection and compare these algorithms with the pathologist's performance [10]. It was the first grand challenge that provided participants with comprehensive annotated WSI's dataset [35], which allowed for training deep models, such as 22-layer GoogLeNet [43] or 101-layer ResNet [42]. This challenge was closed to new submissions in 2016.

As was said in Chapter 1, two tasks with their own rankings were defined in this challenge: classification of every whole-slide image as either containing or lacking metastases (*task 1*) and identification of individual metastases in whole-slide images (*task 2*) [10].

Performance of pathologists

To establish a baseline performance score for pathologists, one professional pathologist marked every metastasis in the CAMELYON16 challenge's test slides on a computer screen without any time constraint (WOTC) [10]. After that, to imitate the routine pathology diagnostic workflow, 11 experienced pathologists were asked to independently assess the challenge's test slides using a light microscope. The assessment was performed with a time constraint (WTC) set as a flexible 2-hour time limit [10].

The pathologist WOTC required roughly 30 hours. In task 1, the pathologist WOTC achieved a sensitivity of 93.8 %, a specificity of 98.7 %, and an AUC of 0.966. In task 2, the production of false-positives was zero, but 27.6 % of metastases were not identified [10].

The pathologists WTC required a median of 120 minutes. In task 1, they achieved a mean sensitivity of 62.8 %, a mean specificity of 98.5 % and a mean AUC of 0.810. In task 2, for macrometastases detection, pathologists achieved a mean sensitivity of 92.9 % and a mean AUC of 0.964. For micrometastases detection, pathologists achieved a mean sensitivity of 38.3 % and a mean AUC of 0.685. 37.1 % of the slides with only micrometastases were missed even by the best performing pathologists. Specificity remained high, which indicates that the rate of false-positives was not high [10].

[3] Available at `https://www.kaggle.com/c/histopathologic-cancer-detection/discussion/`.

[4] Available at `https://www.kaggle.com/c/histopathologic-cancer-detection/notebooks/`.

	Task 1: Metastasis identification	Task 2: Metastases classification	Algorithm model		
Team	**FROC score**	**AUC**	**Deep learning**	**Architecture**	**Comments**
HMS and MIT II	0.807	0.994	Yes	GoogLeNet	Ensemble of 2 networks; stain standardization; extensive data augmentation; hard example mining
HMS and MGH III	0.760	0.976	Yes	ResNet	Fine-tuned pretrained network; fully convolutional network
HMS and MGH I	0.596	0.964	Yes	GoogLeNet	Fine-tuned pretrained network
CULab III	0.703	0.940	Yes	VGG-16	Fine-tuned pretrained network; fully convolutional network
HMS and MIT I	0.693	0.923	Yes	GoogLeNet	Ensemble of 2 networks; hard example mining
Pathologist WOTC	0.724	0.966	–	–	Expert pathologist who assessed without a time constraint
Mean pathologists WTC	–	0.810	–	–	The mean performance of 11 pathologists in a simulation exercise designed to mimic the routine workflow of diagnostic pathology with a flexible 2-h time limit

Table 4.1: Overview of methods and results of the top five submitted algorithms (upper part) compared to pathologists performance (lower part) for task 1 and 2 in the CAMELYON16 challenge, taken and edited from [10].

Performance of algorithms

The majority of submitted algorithms used deep learning-based methods. Some participants used other ML approaches, like texture features extraction combined with supervised classifiers (support vector machines or random forest classifiers) [10]. Overall, algorithms using DCNNs performed significantly better – the top-performing algorithms in both tasks all used DCNN as the underlying methodology. Most popular architectures among the top-performing algorithms were the GoogLeNet, VGG-16 [44] and ResNet. All of them performed similarly or even outperformed the top pathologists WTC both in micro and macrometastases detection. Table 4.1 describes a detailed comparison of top-performing algorithms compared to pathologists performance.

In task 1, submitted algorithms were sorted by their performance using the AUC score. The AUC score ranged from 0.994 to 0.556 for all 32 participants[5]. The best performing algorithm by team Harvard Medical School (HMS) and Massachusetts Institute of Technology (MIT) II was presented in [45]. This

[5] According to the CAMELYON16 challenge's leaderboard, available at `https://camelyon16.grand-challenge.org/Results/`.

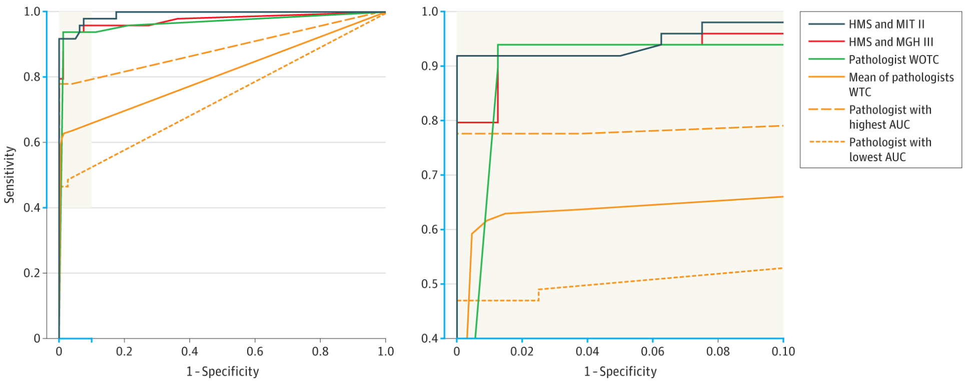


Figure 4.1: ROC curves of top two performing algorithms compared to pathologists for metastases classification task (task 1), taken from [10].

method used an ensemble of two GoogLeNet architectures – one trained with and one without a hard example mining. With an AUC of 0.994, it outperformed other submissions, pathologists WTC and surprisingly also pathologist WOTC [10]. The second-best performing algorithm by team HMS and Massachusetts General Hospital (MGH) III used a fully convolutional ResNet-101 architecture [46]. It achieved an AUC of 0.976 and excelled among other algorithms with the highest AUC in detecting macrometastases [10]. Figure 4.1 shows the comparison of the top two submitted algorithms and pathologists performance.

In task 2, submitted algorithms were sorted by their performance using an FROC true-positive fraction score. The FROC score ranged from 0.807 to 0.097 for all 32 participants[5]. The best performing algorithm from team HMS and MIT II achieved an FROC of 0.807. The second-best performing algorithm by team HMS and MGH III achieved an FROC of 0.760 [10]. Figure 4.2 shows the comparison of the top five submitted algorithms and pathologist WOTC performance. The top-performing algorithm achieved a similar FROC score as the pathologist WOTC when producing a mean of 1.25 false-positive lesions on 100 slides. It also achieved a better FROC score when allowing slightly more false-positive lesions.

4.1.3 CAMELYON17 challenge

CAMELYON16 challenge aimed to improve the task of automated breast cancer metastases detection in single WSI. However, this task is is too simplified and the method of evaluation is less relevant for clinical practice – pathologists during the examination usually observe more than one slide per patient. To make the task workable in clinical conditions, the following key changes were made in the CAMELYON17 challenge [11]:

- instead of single WSI classification, the task focuses on patient-level

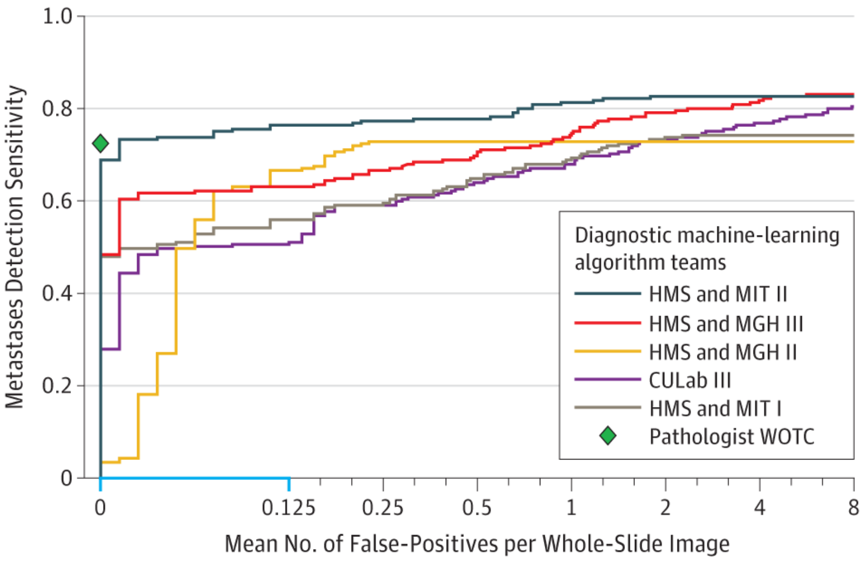


Figure 4.2: FROC curves of top five performing algorithms compared to pathologist WOTC for metastases identification task (task 2), taken from [10].

pN-stage gained from multiple WSIs

- during the evaluation, ITCs are taken into account to predict the pN-stage correctly
- WSIs are provided by five centres instead of only two – dataset size increased from 399 to 1399 WSIs, which brought wider staining diversity across laboratories and possibility to train deeper models

In the following paragraphs, best performin methods, according to CAMELYON17 leaderboard, will be described. The challenge is still open to new submissions. To participate in the challenge, teams are provided to upload a file describing the method they used simultaneously with their solution. Unfortunately, as there is no template for the description file and the conference with a presentation of best algorithms already took place in 2017, some top-performing algorithms are documented poorly. Therefore, the following methods overview might not be exhaustive.

Summary of algorithms

As the CAMELYON17 challenge is open to new submissions, participants tend to use newer and newer DL methods, and the performance of algorithms is still growing [11]. Compared to CAMELYON16 challenge, the dataset was greatly extended and more complex models with numerous supporting methods can be applied [22]. Despite the difference of submitted algorithms, almost all of them follows these fundamental algorithm steps: preprocessing, slide-level classification, slide-level postprocessing and patient-level classification [11].

All teams start with the **preprocessing** step to identify regions with a tissue in the WSI. Mostly Otsu's adaptive threshold [47] at a low resolution level is used with variability in applied colour space, for example, RGB (red-green-blue), HSV (hue-saturation-value) or HSI (hue-saturation-intensity) [11]. To filter tissue regions more precisely, some teams also use morphological operations, for example, median filtering, connected component analysis or size filtering [11].

To perform the **slide-level classification**, all teams train various CNNs on the tiles extracted from the identified tissue regions. In addition, almost all teams perform the extensive data augmentation strategy, and some of them use stain normalisation algorithms [48] to provide a uniform colour distribution [11]. With the recent deployments in the field of semantic segmentation, the state-of-the-art methods have improved from a patch-wise to a pixel-wise classification level. For this purpose, models like DeepLab [49] or UNet [50] are often used [11]. Table 4.2 closely describes models used by top-ranked teams.

In the **slide-level postprocessing**, metastasis-likelihood maps are generated from the test slides using trained CNNs. To select metastasis candidates appropriately, most teams threshold likelihood maps [11]. Some of them also remove small objects to reduce the number of false-positive detections.

The **patient-level classification** consists of predicting the slide-level label (class of WSI) and final patient-level label (pN-stage). In most cases, several features from post-processed likelihood maps are extracted and fed into the classifier, mostly random forest, to determine the slide-level label (negative, ITC, micrometastasis or macrometastasis) [11]. The features are, for example, number of detected metastases or area of the largest detected object. The final patient-level pN-stage is mostly predicted using the same rules as the official pN-staging system determines [11].

Performance of algorithms

Submitted algorithms are sorted by their performance using the quadratic-weighted κ score. The κ score ranges from 0.9570 to -0.2203 for all 102 participants[6]. The overview of currently top-ranked algorithms compared to top-ranked algorithms presented in the CAMELYON17 conference in 2017 provides Table 4.2. Even though the methods have improved significantly since 2017, almost all submitted algorithms still have in common their poor identification of ITC [11].

The best performing algorithm by Deep Bio Inc. team uses a DeepLabV3+ model supported by automated hard example mining process. The slide-level

[6] According to the CAMELYON17 challenge's leaderboard, available at `https://camelyon17.grand-challenge.org/evaluation/leaderboard/`.

Team	**κ score**	**Architecture**	**Ensemble (Size)**	**Slide-level classifier**	**Hard example mining**	**Data augmentation**
Deep Bio Inc.	0.9570	DeepLabV3+	No	DBSCAN	Yes	Yes
Nicolas Pinchaud	0.9386	DeepLab	Yes (3)	RF	Yes	Yes
SenseTime	0.9243	–	–	–	–	Yes
IITM, India	0.9090	DenseNet, Inception-ResNetV2, DeepLabV3+	Yes (3)	RF	Yes	Yes
Ozymandias	0.9085	ResNet-101	No	XGBoost	No	Yes
Lunit	0.8993	ResNet-101	Yes (3)	RF	No	Yes
HMS-MGH-CCDS	0.8806	ResNet-101	No	RF	Yes	Yes
VCA-TUe	0.8729	GoogLeNet	No	DBSCAN	Yes	Yes
MIL-GPAT	0.8567	GoogLeNet, ResNet-50	Yes (3)	RF	Yes	No
Indica Labs	0.8554	VGG	No	Simple heuristic	No	Yes

Table 4.2: Overview of methods and results of the top five submitted algorithms according to CAMELYON17 leaderboard (upper part) compared to top five algorithms from the CAMELYON17 conference in 2017 (lower part), taken and edited from [11] and CAMELYON17 leaderboard.

classification is processed using the DBSCAN algorithm [51]. The second-best performing algorithm by the team of Nicolas Pinchaud was proposed in [52]. They profit from using an ensemble approach – several DeepLabV3 models learned on different pixel resolutions compounded together. Extensive data augmentation and online hard example mining are also used. The slide-level classification is processed using a random forest classifier.

Chapter 5

Methods

This chapter aims to clarify our method used to solve a given problem, namely the automatic detection of metastases in images. First, we provide an overview of the theoretical background. Then we describe the core of the work itself - an end-to-end pipeline for solving the task as required by the Kaggle challenge and CAMELYON challenges.

5.1 Theoretical framework

The following paragraphs summarize the theoretical techniques used in this work. We focus on understanding the basic principles of these algorithms. That will help us to orient ourselves in the rest of the chapter easily.

5.1.1 Otsu's adaptive thresholding algorithm

Otsu's adaptive thresholding algorithm is a thresholding technique for adaptive binarization of images. It scans all the possible threshold values and tries to find the optimal one [53].

This technique assumes that the image consists of an only foreground and background objects with well-distinguished pixel distributions. That means we have a *bimodal image* with two peaks in its histogram [47]. For that image, we can iterate over all the possible thresholding values and take the value approximately in the middle of those peaks. In other words, the algorithm minimizes the within-class variance of the foreground and background colour distribution [47].

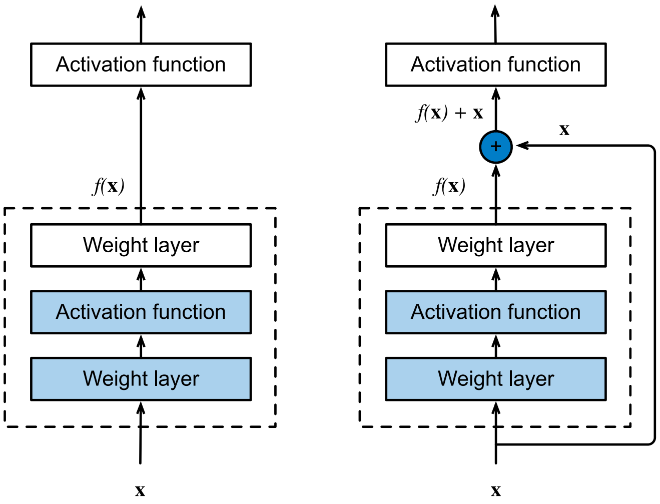


Figure 5.1: Comparison of a regular block (left) and a residual identity shortcut connection block (right), taken from [54].

5.1.2 Convolutional neural networks

ResNet

ResNet was introduced in [42] in 2015. They presented an *identity shortcut connection* that skips one or more layers in the network, as shows Figure 5.1 [42]. That is the core idea for the ResNet architecture.

The identity block does not have any parameters, and it only adds the output from the previous layer to the next layer [42]. That allows us to simply stack these identity block which should not degrade the network performance. As a result, there is more ability to train deeper networks and reach better results [42].

Double or triple-layer skips with nonlinearities (ReLU) and batch normalization in between are implemented in most ResNet models [42].

Fully convolutional network

One of the earliest CNN used for semantic segmentation is the Fully convolutional network (FCN). The idea of it was introduced in [46] in 2014. The FCN architecture assembles a stack of convolutional layers in an *encoder-decoder fashion*.

The encoder part, downsampling the input image and extracting features just as in standard CNN, is followed by the decoder part, which uses one transposed convolutional layer to upsample obtained features to a full-resolution

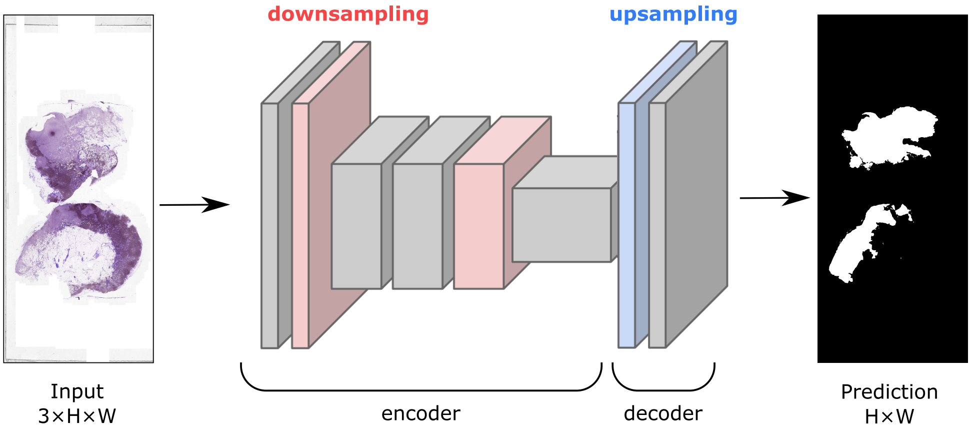


Figure 5.2: The structure of the FCN architecture, taken and edited from [55].

segmentation map [46]. This can be seen in Figure 5.2.

UNet

UNet was initially developed for the usage in biomedical image segmentation tasks, proposed in [50] in 2015, but afterwards reused in many other segmentation problems. It is built on the concept of FCN with respect to the encoder-decoder structure. The main differences are that the encoder and decoder parts are mirrored, which means more upsampling layers, and using *skip connections* to concatenate layers in encoding and decoding part [50].

Skip connections directly sum one layer in the encoding part with the decoding layer while ignoring all the layers in between. That allows the network to reconstruct the spatial information lost during the downsampling process [50]. A simple visualization of the UNet structure provides Figure 5.3.

DeepLabV3

One of the newest state-of-the-art semantic segmentation models is a third version of the DeepLab model, called DeepLabV3. DeepLab was introduced in [49] in 2016, and multiple improvements have been made since then. This model is also based on encoder-decoder architecture. The main difference that distinguishes this model from all others is using a so-called *atrous convolution* for the upsampling process [49].

This atrous convolution simply expands the field of the filter's view using the parameter r called atrous rate. It defines the stride at which the input image is sampled [49]. Choosing the atrous rate as $r = 1$ corresponds to the standard convolution, DeepLab is using values of 6, 12 and 18 [49].

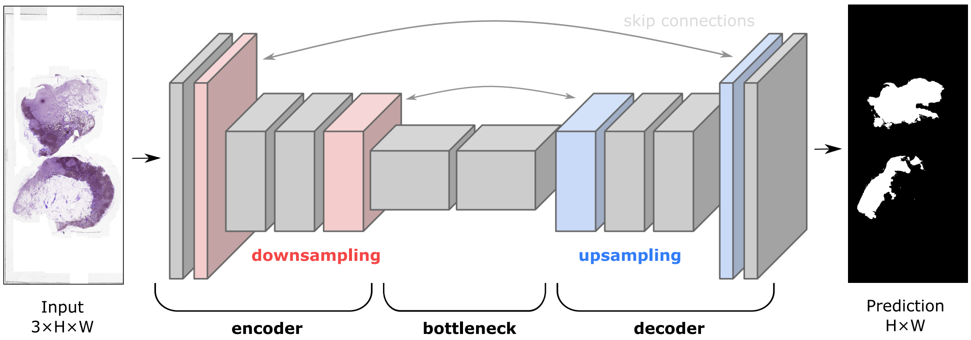


Figure 5.3: The structure of the UNet architecture, taken and edited from [55].

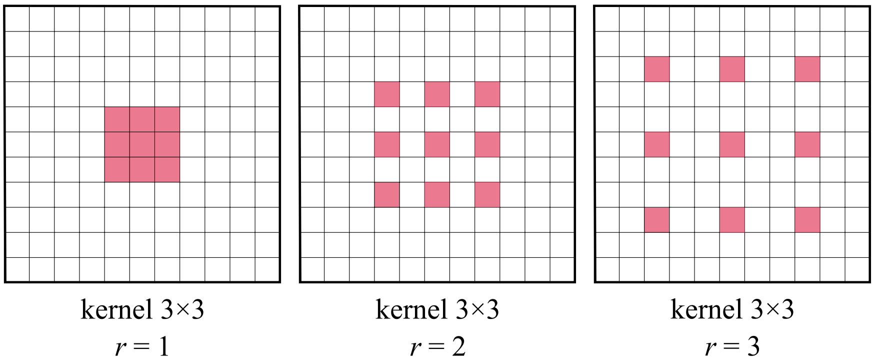


Figure 5.4: Illustration of atrous convolution for the purpouses of the DeepLab model.

This methodology profits from the flexibility of adjusting the filter's field of view to incorporate a larger context but still preserving the same number of parameters [49].

5.1.3 Convolutional neural network's tuning

Transfer learning

Transfer learning is a deep learning technique. It profits from taking a model trained for a specific task and reusing it as the starting point for a model on another related task [56]. Instead of training our model from scratch, we transfer learned weights from another model and use them as a starting point [57]. That leads to an easier model learning, especially in the first few layers of the network, where training can profit from the common features that have already been pre-trained and are similar for the majority of convolutional neural networks [57].

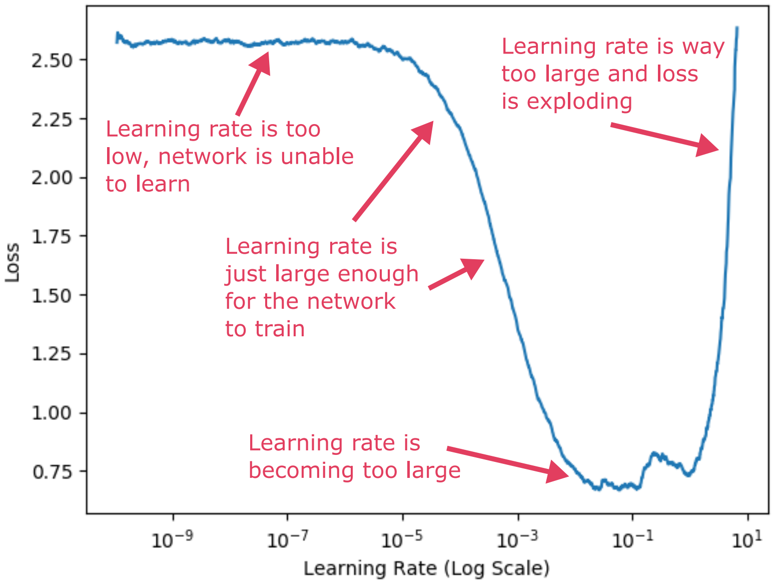


Figure 5.5: An auxiliary plot with detailed description for determining the most suitable learning rate using the learning rate searching algorithm from [58]. We aim to choose a large learning rate as it helps to regularize the training, but if we choose a value that is too large, the training will diverge [58]. The most optimal learning rate is just before the loss starts to increase exponentially. However, as the learning rate corresponding to the minimum value is at the edge between improving and diverging, it is important to choose a value, for which the loss is still descending [58]. Taken and edited from [58].

Initial learning rate searching process

In this section, we propose to use a special learning rate searching technique presented in [58]. According to the technique, the process of finding the learning rate is pretty simple. Over an epoch, our optimizer starts with a very low learning rate that is multiplied by a certain factor at each mini-batch until it reaches a very high value and starts to diverge. We record the loss at each iteration and once we are finished, plotting those losses against the learning rate helps us to find the optimal learning rate [58]. We can determine the most suitable learning rate following the observation described in Figure 5.5.

5.1.4 Other machine-learning classifiers

Random forest

Random forest is an ensemble classification algorithm consisting of decision trees [59]. That means, instead of using a single classifier, we use multiple classifiers to make the prediction.

First, we need to know how to train a single decision tree. When the decision tree receives a training dataset, it starts to run it through the tree. Each decision node in the tree needs to have its own rule to determine which branch to choose [59]. As the vector moves down the tree, the tree select a feature that allows it to split the training data into two branches. It continues with this process until it has no more features to divide. After that, it assigns a class label to each of the leaves containing a subset of the original dataset [59].

While training, our goal is to construct the most fitting tree for all our samples [59]. In other words, we need to create the most describing set of features that the tree is looking for in input samples. As using only a single decision tree might result in poor performance, we improve it using an ensemble of them. While training, at each node, we select the best feature for splitting from a random subset of the available features [59].

For the final prediction, every tree makes the decision individually, and the classifier's output is based on the majority voting strategy – the class with most tree's votes is chosen as the final one [59]. That results in the availability to capture more complex feature patterns and reduce the chance of overfitting [59].

XGBoost

XGBoost is an optimized decision-tree-based algorithm based on an extreme gradient boosting [60]. Its implementation is an open-source software library[1] which was developed as a research project and presented in [60].

The XGBoost is an ensemble technique that uses an iterative approach [60]. It profits from the so-called *boosting* – instead of training the decision trees individually and ensembling them afterwards, we iteratively create and train one tree, which tries to correct the mistakes made by the previous ones, and subsequently add it to them. This process is repeated, and trees are added as long as the performance increases [60].

Specifically, in the extreme gradient boosting, models are trained to correct the errors utilizing a gradient descent algorithm optimized through tree-pruning, parallel processing and regularization [60].

[1] Available at `https://github.com/dmlc/xgboost`.

5.1.5 Loss functions

Cross-entropy loss

The cross-entropy loss is a most commonly used loss function. It is defined as

$$H(p,q) = -\sum_i p_i \log q_i, \tag{5.1}$$

where p_i stands for the ground truth label for the i-th class and q_i is the label predicted by the network for the i-th class. This scoring is repeated for all patches. For the binary cross-entropy loss, $i = 2$.

Soft dice loss

The soft dice loss is based on the Dice coefficient, which is a measure of overlap between the prediction and ground truth [61, 62]. That makes the final loss function more immune to the data-imbalance issue.

The coefficient's value ranges between 0 and 1, where 0 means no overlap and 1 means complete overlap. It can be written as

$$S(p,q) = \frac{2\sum_i p_i q_i}{\sum_i p_i^2 + \sum_i q_i^2}, \tag{5.2}$$

where p_i stands for the ground truth label for the i-th pixel and q_i is the label predicted by the network for the i-th pixel [61]. This scoring is repeated for all patches.

In order to use this coefficient as a loss function, we need to convert it to a form that can be minimized. To do so, we simply use the following function:

$$D(p,q) = 1 - S(p,q). \tag{5.3}$$

This function is known as the soft dice loss.

5.1.6 K-fold cross-validation

K-fold cross-validation is a commonly used technique for model evaluation. This method is used mainly for tasks whose amount of data is small, and its further reduction may suffer from very biased results [64].

The process of K-fold cross-validating is following: we randomly shuffle the dataset, split it into K groups, take one group as a test fold and the remaining as a training fold, fit a model, evaluate it, and repeat this run for K times. After this process, we summarize the model's performance for each

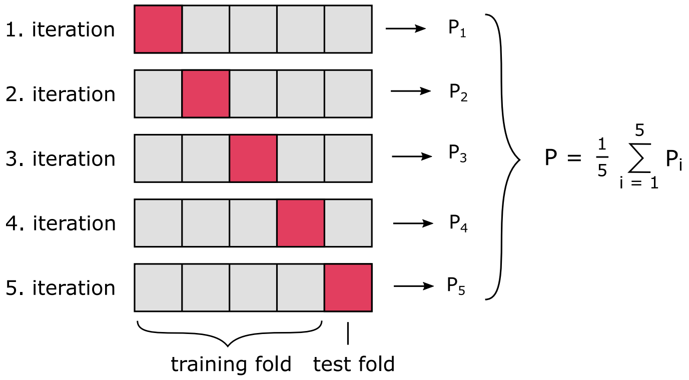


Figure 5.6: 5-fold cross-validation process visualization, taken and edited from [63].

run and compute the score using the arithmetic mean over all K runs [64]. The process is graphically expressed in Figure 5.6.

This method is popular mainly because of its simplicity and ability to result in less biased estimation of the model performance than other methods [64].

5.2 Baseline solution for the purposes of Kaggle competition

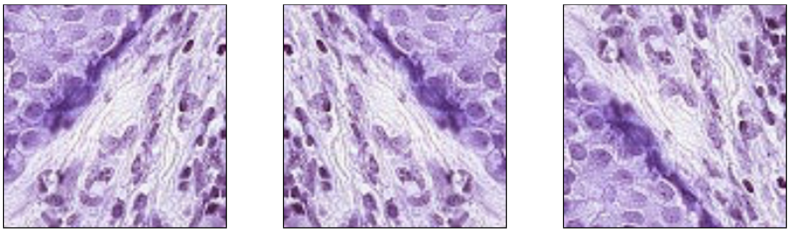
The following section introduces a baseline solution for the metastasis detection task. To get familiar with it, we use Kaggle's Histopathological cancer detection challenge and try to detect metastases using a simple classification problem – if the patch contains metastases or not. From the results of this assignment, we can then easily develop a larger pipeline, which can solve even more complicated tasks. A simplified version of the entire classification process designed by us illustrates Figure 5.8.

5.2.1 Dataset preparation

Initially, as mentioned in Chapter 3, we are provided with train and test data only, and we need to split original training dataset into two parts. One to train the model and one to validate our model's results. Table 5.1 shows that the negative/positive ratio in the original training dataset is close to 60/40 – classes are pretty well balanced. It is essential to maintain the same ratios of negative and positive samples in both training and validation dataset. To achieve it, a stratified sampling strategy is used. After splitting, 90 % of the original dataset is the training data, and 10 % is the validation data. A detailed proportion of positive and negative labels in dataset describes Table 5.1.

	Before split	**After split**		
Label	Train	Train	Validation	Test
Negative	130 908	117 817	13 091	–
Positive	89 117	80 205	8 912	–
Total	220 025	198 022	22 003	57 458

Table 5.1: Data distribution in the PCam dataset before and after split.



(a) : Original patch **(b) :** Horizontal flip **(c) :** Vertical flip

Figure 5.7: Example patch from the PCam dataset before and after applying augmentations.

5.2.2 Patches preparation

As was said in Chapter 3, the label of each image is influenced only by the centre region of 32 × 32 pixels. For the purpose of this work, we use a full image size, as there might be some useful information about the surroundings which would be lost after cropping the image too tight. At the same time, we normalize patches and resize them to 224 × 224 pixels – the chosen architecture was initially pre-trained on a different dataset, and we must respect it and copy its properties.

To avoid overfitting, data augmentation is used. One of the key augmentation for patches taken from histopathological slides is horizontal and vertical flip, because there is little importance on how the initial slide is oriented. For this reason, these two augmentations are implemented.

5.2.3 Convolutional neural network

After the data preparation process, we need to train a model to predict the correct label for each patch in the test part of the dataset. Because it is our first encounter with a histopathological type of data, we start with a relatively simple convolutional neural network, already pre-trained on another dataset. This process is called *transfer learning* and is described in Section 5.1.3 in more detail.

In our case, a 50-layer residual network, called *Resnet-50*, with weights pre-

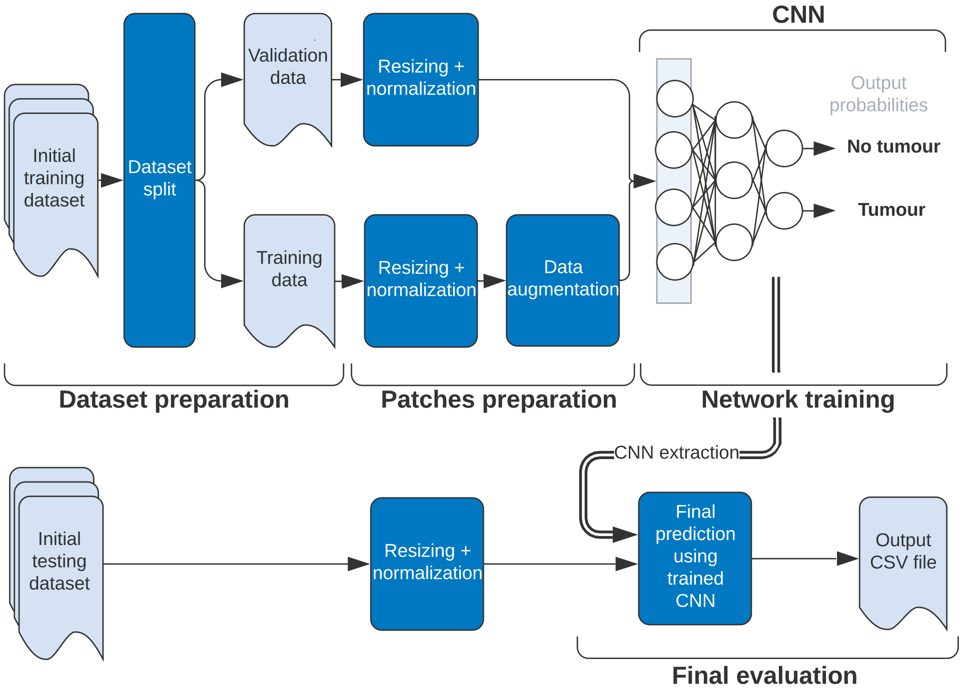


Figure 5.8: Visualization of the whole pipeline for the task of detecting a metastasis tissue as defined in Kaggle challenge. First, dataset is loaded and preprocessed to be suitable for the CNN. Then, chosen CNN is trained on the training data. Trained weights are then used for the final prediction for the test dataset. Then, the pipeline outputs final CSV file with patches names and their probabilities of containing tumour.

trained on the ImageNet dataset, is used. This network is closely described in Section 5.1.2.

5.2.4 Training parameters

To train the ResNet model properly, we define a well-chosen *hyperparameters setup*. These parameters are independent of the training process, and their value is set before starting it [56]. They can possibly save us a lot of time if we choose them wisely because they directly affect the behaviour of a trained model. To choose the model's hyperparameters, there are several *hyperparameter-tuning strategies*, such as random search, grid search or Bayesian optimization [65].

In our case, the searching focuses mainly on tuning the learning rate. For a correctly chosen learning rate, we use a special learning rate searching approach presented in [58] and described in Section 5.1.3. We can determine the most suitable maximum learning rate from Figure B.1. For the purpose of this work, we choose $3 \cdot 10^{-4}$ as the initial value for the learning rate.

Parameter	Value
Learning rate	Adaptive, with initial value $3 \cdot 10^{-4}$
Batch size	64
Number of epochs	40
Loss function	Binary closs-entropy loss

Table 5.2: Parameters selected and fine-tuned for the ResNet-50 model's training.

Concerning the results given by the learning rate searching algorithm, we use the chosen learning rate value as the initial value for the training process. To receive better results and prevent the model from overfitting, the learning rate is adjusted during the training. Implementation of this so-called *adaptive learning rate* is pretty straightforward. We watch over the performance of the model, and if no improvements are seen, we decrease the learning rate.

Apart from the learning rate, we also define other training parameters, like the number of epochs, batch size or loss function. The number of epochs is chosen to enable the model to maximize its performance but still prevent the overfitting. Batch size is chosen as the maximum number of patches that can be simultaneously trained concerning our hardware capacity. The loss function is chosen considering the dataset we were given. As the dataset's classes are balanced pretty well, we choose simple binary cross-entropy loss function for our classification problem, described in 5.1.5.

In the previous paragraphs, we have clarified our selection of some training parameters. The overview of all the specified training parameter's values summarizes Table 5.2.

5.2.5 Final submission

After the process of training, we fed the trained CNN with patches from the test set, and save its output for each of them. Then, we store to final CSV file patches names and respective probabilities of their centre 32×32 region containing at least one pixel of tumour tissue in a format required by the organizers. This file is subsequently submitted to the official challenge's submission web page.

5.3 Extended solution for the purposes of CAMELYON competitions

As we have already constructed the baseline solution using the Kaggle competition, we can now extend it. The CAMELYON challenge requires us to move the task from patch-level classification to providing the full slide segmentation and slide-level classification with final patient-level aggregation and patient's pN-stage prediction.

The end-to-end pipeline designed for the task of predicting the pN-stage of the patient can be divided into following essential steps: slide preprocessing, patch-level segmentation, slide-level classification and patient-level classification. The whole pipeline, in a simplified version, images Figure 5.13.

5.3.1 Slide preprocessing

This section describes the preprocessing step designed for the purposes of the CAMELYON16 and CAMELYON17 challenge.

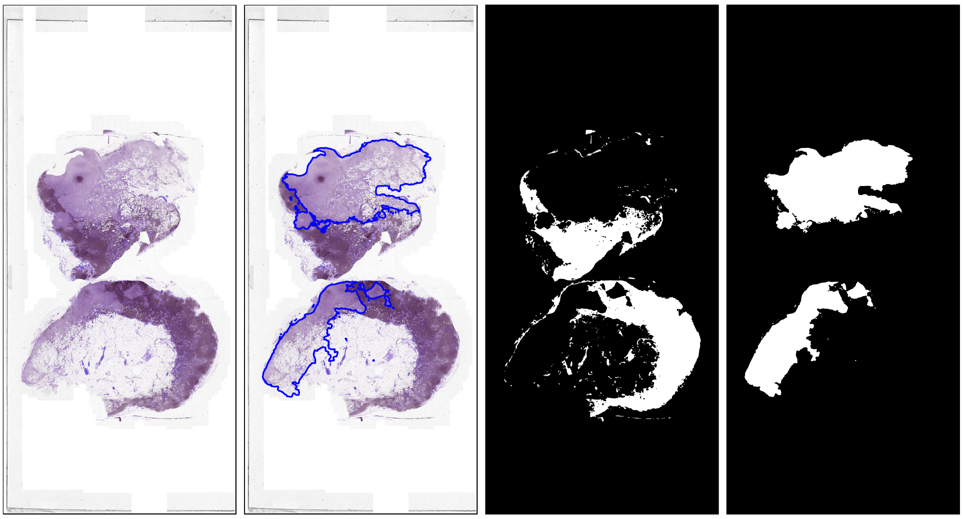
Identification of tissue regions on the WSIs

Identifying the tissue on the WSI is a crucial step for the final algorithm efficiency. On the WSIs, there are typically large areas containing nothing more than the background. These regions do not need to be processed as they do not carry any useful information. For this reason, we use simple filtering and thresholding steps to remove them.

Firstly, we use the *Otsu's adaptive thresholding algorithm technique* [47], described in Section 5.1.1. In our case, we need to get rid of the white background colour. Therefore, the RGB colour space of the WSI is transformed into the HSV colour space. In the HSV colour space, there is a small within-class variance between the tissue sections colour and background colour value, and Otsu's thresholding algorithm can easily find the right threshold [66]. We apply the thresholding technique to the hue and saturation component.

After the thresholding, an additional morphological operation called *morphological hole-filling* is processed to refine the thresholded map. This operation is useful for filling small holes inside the foreground objects, in our case detected tissue sections. Also, a median filter is applied to remove small isolated objects and smoothen detected regions. Additionally, on some of the slides, the attachment mark used during the staining and scanning process is visible, usually black-coloured with a shape of cross, star or circle. Detect these regions as tissue-ones is undesirable. To resolve it, we use a simple thresholding algorithm: we convert the filtered map from RGB colour space to grayscale colour space and unmark every pixel, previously marked as a tissue one, lower than a hand-tuned threshold value. A visualization of tissue detection algorithm steps applied on an example WSI is illustrated in Figure A.1.

The whole tissue segmentation routine is processed on a low-resolution



(a) : Original slide

(b) : Slide with marked tumour locations

(c) : BW mask of non-tumour regions

(d) : BW mask of tumour regions

Figure 5.9: Visualization of an example WSI taken from the CAMELYON dataset along with tumour bordering and BW mask presenting the non-tumour regions and tumour regions. Black stands for no-tumour pixels, white presents tumour pixels. BW map presenting non-tumour regions is created from the tissue region map after subtracting all tumours annotated by the organizers.

version of the slide and consequently projected to the required resolution. Identifying tissue regions directly on the full-resolution slide would take an unreasonable amount of time, and our memory source is limited as well. Oppositely, if we instead use the lower-resolution version and resize it after the segmenting routine, the process would take much less time and memory, and the difference from the map segmented on the full resolution (level 0) would be insignificant. In our case, we detect tissue regions on a 32-times downsampled version of the original WSI (level 5).

Patches extraction

Since the WSIs are extremely large images, CNN cannot handle them as the input directly. To resolve it, we slice the slides to small patches with fixed size and train CNN with them. The performance of CNN is highly affected by the number of extracted patches and their size. Therefore, the process of creating a dataset cannot be underestimated.

We propose a simple random patch extraction strategy. At first, we generate the map with tissue regions accordingly to the previous section and resize it to the required level. At the same time, we generate a map with tumour regions using the annotations provided by organizers. Using these two masks, we are able to prepare a map representing non-tumour tissue regions as

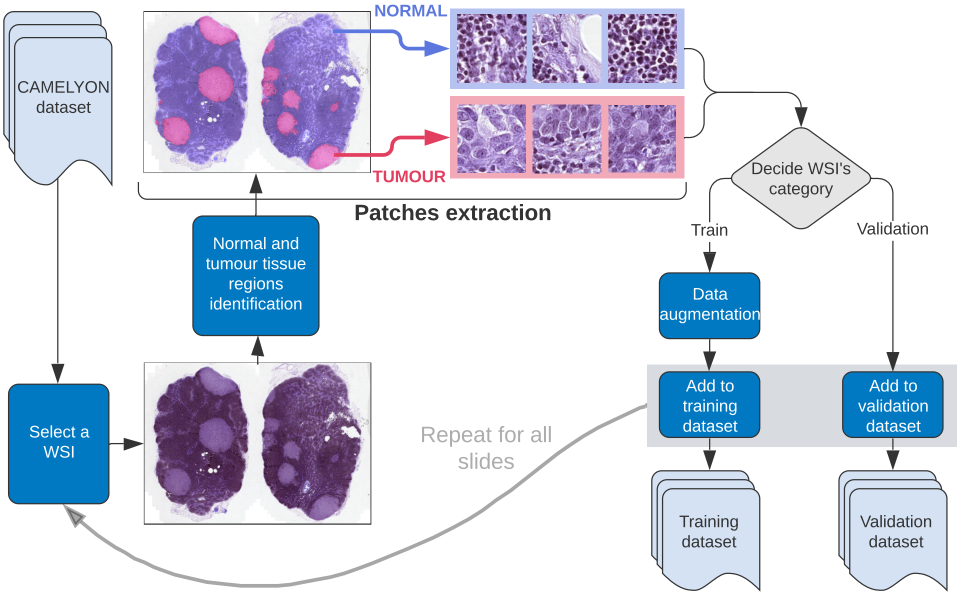


Figure 5.10: Visualization of the patches extracting process. First, a WSI is loaded. Then, tumour regions and healthy-tissue regions are detected. Patches are subsequently sampled from both these regions, and, with respect to the WSI's origin, we store extracted patches to the training or validation dataset. This process is repeated for all WSIs that should be sampled.

presented in Figure 5.9. After that, we sample a fixed number of patches from both tumour and non-tumour regions from the slide downsampled to the required level. To perform that, we randomly choose coordinates from the region marked as tumour or non-tumour and extract a patch of fixed size with chosen coordinates in the middle of the patch. The process of sampling the patches from CAMELYON dataset illustrates Figure 5.10.

Using this extracting strategy approach, healthy and tumorous tissue regions are equally distributed across the dataset, and the dataset size is still within tolerable limits. Moreover, various areas of the slide are accessed and extracted, for example, areas located at the edge of the tissue region, which are often misclassified. With using the random sample strategy, the probability of sampling this area is the same as of the area in the middle of the tissue mass.

For the training part of the dataset, training slides from both CAMELYON16 and CAMELYON17 challenge are used to extract patches. For the validation part of the dataset, testing slides from CAMELYON16 are used to extract patches. Testing slides from CAMELYON16 are also used for the submission to the CAMELYON16 challenge. As we do not have official annotations for testing slides from CAMELYON17 challenge, these slides are used only for the final evaluation and submission to the CAMELYON17 challenge.

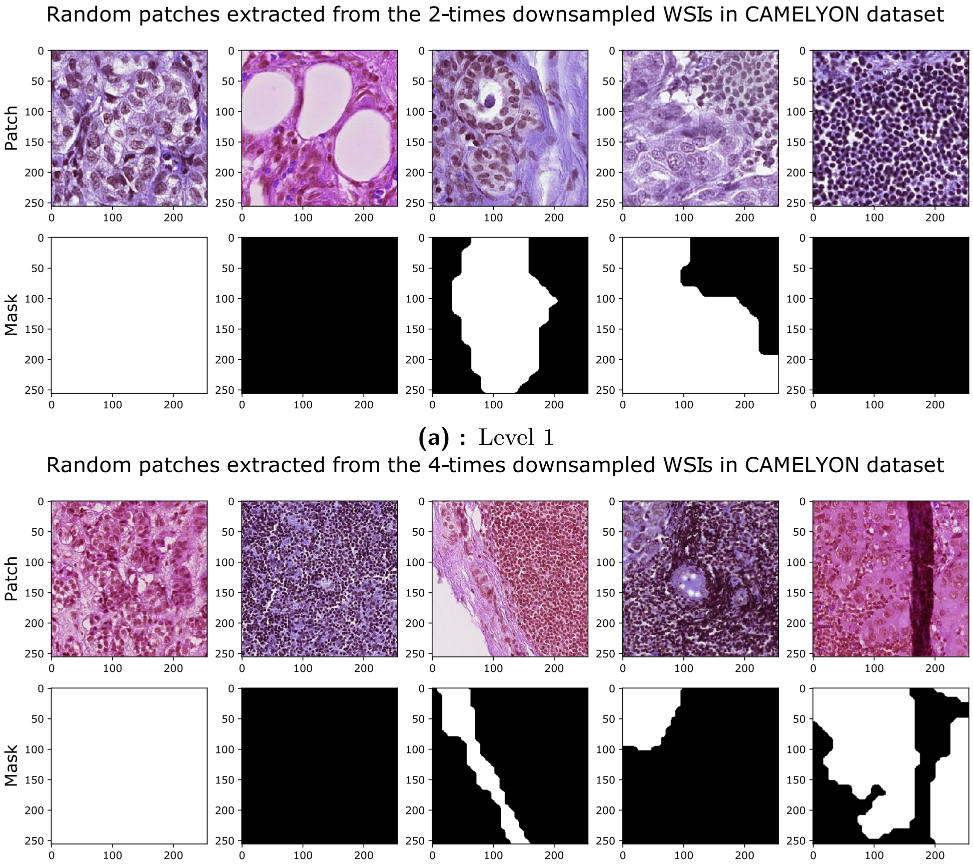
We sample tumorous patches using regions annotated by the organizers. Unfortunately, not all slides containing tumour tissue are exhaustively annotated. Therefore, non-tumour patches are sampled only from fully annotated slides to prevent generating false-negative samples. We generate 500 patches from tumour regions (if they are present) and 250 patches from non-tumour regions per each slide. The size of each patch is 256×256 pixels. As we want to observe if the results of our algorithm are affected by the size of the field captured on one patch, we decide to extract patches from two different versions of the original WSIs (level 0) – one is a two-times downsampled version of the original WSI (level 1) and the second one is a four-times downsampled version of the original WSI (level 2). That results into two independent datasets, both with 225 238 training patches and 55 995 validation patches.

Unlike in the Kaggle competition, in the CAMELYON challenge, we solve given problem as a segmentation task. That means we need to extract patches masks as well to train the model appropriately. We do it simultaneously with the patch extracting process. Just as patches are extracted from the WSI, masks are extracted from the generated tumour-region maps. These grayscale maps are generated on a resolution corresponding to the 32-times downsampled version of the original WSI (level 5) and after that resized to the level 1 or level 2, according to the dataset version. As the edges of tumours on the resized maps are blurred, we need to edit them in the order of model training. The model can be adequately trained for only two classes – white represents tumour, and black represents no tumour. However, with the blurry edges, we have 256 possible grayscale values corresponding to one pixel. To transform the map into a binary one, we use a simple thresholding algorithm: if the pixel's value is lower than a hand-crafted threshold, we convert it to black. Otherwise, we convert it to white. Doing so, we ensure that the extracted patches masks contain only two unique pixel values.

A sample of the final version of extracted patches and their masks is shown in Figure 5.11 for level 1 and level 2.

Data augmentation

Similarly to the Kaggle competition, data augmentation is used to increase the generalization of the model. As slides from more medical centres are used, model needs to adapt well to various staining and scanning conditions. For the model training, we use these augmentations: random flip, random rotation, random brightness, random contrast and random HSV editing. Example of these augmentations is presented in Figure 5.12. Each of the augmentations performs a specific action with probability $p = 0.5$ – they can flip and rotate the patch, increase or decrease brightness and contrast, and edit the HSV colour distribution. This set of augmentations incorporates a wide range of possible patches variations and allows the model to be trained robustly.


Random patches extracted from the 2-times downsampled WSIs in CAMELYON dataset
Patch
Mask
0
50
100
150
200
250
0
100
200
Random patches extracted from the 4-times downsampled WSIs in CAMELYON dataset

(a) : Level 1

(b) : Level 2

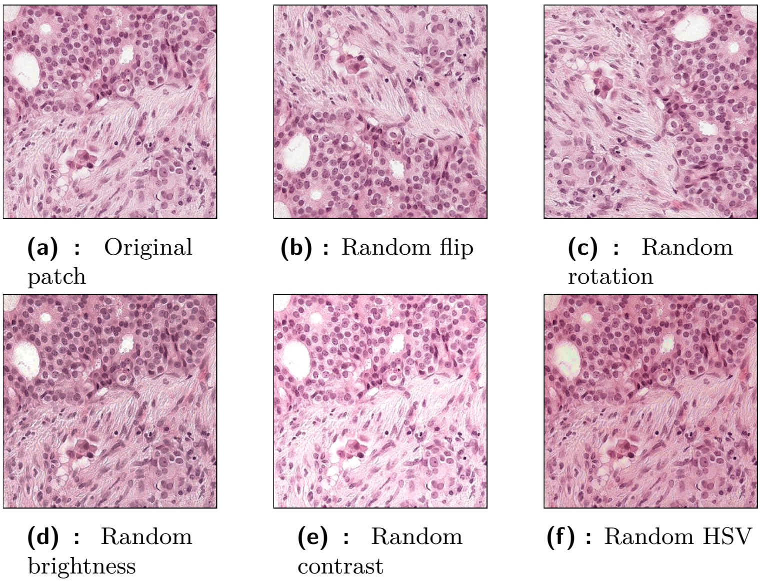
Figure 5.11: A random sample of patches extracted from the CAMELYON dataset using our algorithm. Along with the patches, their masks with tumour locations are extracted as well.

5.3.2 Patch-level segmentation

As we have already indicated, we moved the task from classifying patches as either containing metastasis or not to the segmentation of tumour regions on the patches. What that means is, instead of making a patch-level prediction, we make a prediction on a pixel-level. We aim to classify every pixel into one of two classes – containing tumour or tumour-free. That allows us to evaluate every slide on a pixel-wise level and bring the task of tumour localization closer to the clinical practice.

Convolutional neural network

To accomplish the segmentation task, we propose to use a well-known CNN designed for the semantic segmentation task – DeepLabV3 model with a ResNet-101 backbone. We also compare the performance of the DeepLabV3



(a) : Original patch

(b) : Random flip

(c) : Random rotation

(d) : Random brightness

(e) : Random contrast

(f) : Random HSV

Figure 5.12: Example patch from the CAMELYON dataset before and after applying augmentations.

model with two other popular CNNs designed for the semantic segmentation tasks – UNet with a ResNet-50 backbone and Fully Convolutional Network with a ResNet-50 backbone. All three models are deepely described in Section 5.1.2. In this work, we do not use pre-trained versions of mentioned models.

Each of the used models has specified a so-called *backbone*. The term 'backbone' refers to a CNN performing the feature extraction [67]. Rest of the segmentation framework is subsequently built around the extractor. Taking this into account, we can choose the most suitable backbone for our task. As we have already observed this area in the baseline solution in Section 5.2, we choose the same architecture, ResNet, as the backbone for all three models.

Training parameters

Similarly to the ResNet-50 used for the Kaggle competition, we need to define a well-chosen hyperparameters setup for our segmentation CNNs. We again aim mainly on tuning the initial learning rate value. For that purpose, we use the process presented in Section 5.1.3. The curves for our models generated by the learning rate searching process demonstrate Figures B.2, B.3 and B.4.

The main difference between the training parameters used for the model in Kaggle competition and in CAMELYON competition is the choice of batch size and number of epochs. As we moved from the classification task to the segmentation one, we operate with much more parameters and model's complexity. Thanks to that, we have to take into the account the maximum memory capacity we can operate with, which results in relatively small batch

Parameter	Value		
	DeepLabV3 with a ResNet-101 backbone	FCN with a ResNet-50 backbone	UNet with a ResNet-50 backbone
Learning rate	Adaptive, with initial value $5 \cdot 10^{-6}$	Adaptive, with initial value $1 \cdot 10^{-5}$	Adaptive, with initial value $5 \cdot 10^{-4}$
Batch size	16	32	32
Number of epochs	8	10	10
Loss function	Soft dice loss	Soft dice loss	Soft dice loss
Pretrained	No	No	No

Table 5.3: Parameters selected and fine-tuned for the models performing the segmentation task.

size. The complexity of the models is also reflected in the number of epochs processed during the training. That is reduced because each epoch lasts much longer.

Another crucial parameter in our setup is a loss function. The choice of it is closely tied to the type of the task and data we are facing [56]. Our segmentation task might be understood as a binary classification problem at a pixel level – if the pixel contains tumour (white) or not (black). To use this approach, we need a balanced dataset. However, in our dataset, we can find many patches that are entirely white or black, and we could potentially have troubles training the model accurately [61].To deal with our imbalanced dataset, we propose to use another popular loss function for segmentation tasks called *soft dice loss function*, described in Section 5.1.5.

More details about parameters selection for each of our models are described in Table 5.3.

5.3.3 Slide-level classification

The process described in Section 5.3.2 results in a model trained to take as input a whole slide images patch and the ground truth segmentation mask, and produce a tumour probability mask for the patch. The intensity value of every pixel expresses the probability of being a metastasis. White colour means that pixel is a part of a tumorous region with probability 1, black colour means that pixel is not in a tumorous region with probability 1. Now, we can move the results of CNN's training to the slide level.

Metastasis-likelihood maps generation

Since our task is to decide on the spread of metastasis throughout the WSI, we need to aggregate patches predictions for each of the test WSI to a metastasis-likelihood map. To do so, we load chosen WSI, apply our tissue identifying algorithm presented in Section 5.3.1, and cut all the detected area into patches using a grid. The rest of the undetected area is directly classified as non-tumour and left out from the rest of the map-generating process. It is done mainly because of our time dispositions – as our models are trained on patches sampled at a high-resolution version of WSIs, the tumour probability maps also have to be generated at a high resolution. That would lead to tens of thousands patches per every WSI proceeded by our CNNs. Concerning the fact that there are hundreds of test WSIs, we decide to skip the regions not detected as tissue.

Patches are subsequently run through the trained models, and the output probability maps are merged into a metastasis-likelihood map for each WSI. After that, generated tumour probability maps are resized to the 32-times downsampled version of the original slide and directly used to perform the slide-based evaluation and lesion-based evaluation for CAMELYON16 challenge, and slide-based evaluation for CAMELYON17 challenge.

Lesion-based detection and slide-based classification for the purposes of CAMELYON16 challenge

For the **lesion-based detection task**, we follow the official CAMELYON16 requirements and aim to detect every tumour object located within each test WSI with its probability of being a tumour. To achieve that, we take generated tumour probability map and threshold it using value 0.99. Then, we identify all connected regions on it and for each of the regions compute the area they occupy. To prevent our algorithm from generating too many FP's, we use a grid search to find an optimal threshold value for the minimal size of the area that should be taken as a proper tumour, not as a segment. We search for two threshold values in total, one for the model trained on patches extracted from level 1, and one for the model trained on patches extracted from level 2. These values are, respectively, 40 and 66 pixels.

After filtering objects smaller than the specified threshold value, we iterate over all remaining objects, use their central point as the estimated tumour locations and their probabilities (confidence scores) as the object's probability of being a tumour. After locating all tumour objects within one slide, we store their coordinates and confidence scores to a CSV file in a format required by the organizers. According to the fact that the CAMELYON16 competition has been already ended, we evaluate the results ourselves. Competition's organizations provide participants with an official evaluation script, which is

	Value	
Parameter	Random forest	XGBoost
Number of estimators	50	100
Maximal depth of the classifier	80	3
Learning rate	–	0.05
Subsample ratio	–	0.6
Maximum number of classifier's features	$\log_2$(features)	–
Minimum number of samples per leaf	5	–
Minimum number of samples required to split	20	–

Table 5.4: Parameters selected and fine-tuned for the Random forest and XGBoost classifiers.

fed with prepared CSV files and outputs the final FROC score.

As there is no official evaluation script for the CAMELYON16 **slide-level classification task**, and the challenge is no longer open to new submissions, we decide to skip the task within this competition. However, we perform the slide-level classification in Section 5.3.3 as a part of the pipeline for the CAMELYON17 challenge.

Slide-based classification for the purposes of CAMELYON17 challenge

The slide-based classification for the purposes of CAMELYON17 challenge takes, same as in the CAMELYON16 evaluation, as the input generated tumour probability maps. The aim of the slide-based classification is to take a tumour probability heatmap and return the label of the largest found tumour (negative, ITC, micrometastasis or macrometastasis). To resolve it, we propose to use two independent classifiers, Random forest and XGBoost (described in Section 5.1.4), observe their advantages and weaknesses and use the one that suits our problem more.

Both classifiers are trained on features extracted from metastasis-likelihood maps with the size of a 32-times downsampled version of the original WSIs from the training dataset. To enable classifiers to perform at their best, before we start with the training, we fine-tune the hyperparameters setup. The most optimal training parameters are found using a simple grid search, and information about final hyperparameters setup for both classifiers provides Table 5.4.

When classifiers are ready to train, we need to prepare the training data. To do so, we design 25 hand-crafted features, mainly focusing on geometrical and morphological aspects of the heatmap. Features and their descriptions are listed in Table 5.5. To lower the risk of a bias introduced by a selecting specific threshold value, and create a more robust classifier, we extract features for maps thresholded independently on three different values, namely 0.5, 0.9 and 0.99. Therefore, the classifier is fed with 75 features in total per one WSI.

As our training set contains only 500 slides with slide-level labels, the final performance of classifiers is evaluated using the 5-Fold cross-validation, described in Section 5.1.6, to ensure unbiased results.

5.3.4 Patient-level classification

The patient-level classification means to predict the final patient's pN-stage according to the task defined in the CAMELYON17 challenge. That is performed with no additional training. We determine the patient's pN-stage by aggregating all patient's slide predictions received by the trained classifier described in Section 5.3.3, and using the official pN-staging system's rules described in Table 2.2.

Results of all the test patient's slides are stored to one CSV file, which is subsequently submitted to the official CAMELYON17 submission web page.

5.4 Implementation

The following section summarizes the implementation of proposed methods for the metastasis detection task described in previous sections. Scripts, created for purposes of this work, resulted in a total of 20 files with more than 5 500 lines of code. Individual files together create a large end-to-end pipeline solving the task of metastasis detection. Main pipeline's components are divided into five folders in style similar to the methods defined above.

5.4.1 Baseline solution's scripts

In the first folder, called `1_baseline_solution`, we can find the implementation of Section 5.2. The `train_network.py` performs the whole process of model's training and validation for solving the task of Kaggle challenge. The process of searching for an initial learning rate is also implemented in this file. Final evaluating and creating the submission CSV file is performed by `evaluate_kaggle.py`.

5.4.2 Preprocessing and visualization's scripts

In the second folder, called `2_preprocessing and visualization`, we can find the implementation of Section 5.3.1. The tissue sections are identified in `make_masks.py` where also other visualizations are created. After that, created maps with highlighted tissue regions are saved using the `save_masks.py`. Subsequently, we can run `make_patches.py`, which generates a given number of patches from both normal and tumour regions on the WSI using previously generated tumour maps. Patches from specific WSI are stored in a prepared folder using `save_patches.py`. Finally, a dataset containing extracted patches for training and testing purposes is created using `create_dataset.py`. Information about it is stored in a CSV file.

5.4.3 Patch-level segmentation's scripts

Patch-level segmentation, described in Section 5.3.2, is implemented in folder `3_patch_level_segmentation`. Model for the task of patch-level segmentation is prepared, trained and validated in `train_network.py`. Multiple processes are implemented in this file. In addition to the model's training, we can also evaluate its performance, find initial learning rate, or save patches with extremely high or low losses.

5.4.4 Slide-level classification's scripts

Section 5.3.3, describing slide-level classification process, is implemented in folder `4_slide_level_classification`. Metastasis-likelihood maps are generated in `generate_maps.py`, and final evaluation of the lesion-based detection task for purposes of CAMELYON16 challenge is performed by `evaluate_c16.py`. This file takes generated tumour probability maps as input, finds every tumour is located on the slide and writes it to the submission CSV file.

To perform the slide-based classification for the purposes of CAMELYON17 challenge, features from the tumour probability maps are generated and stored using `generate_features.py`. These features are then fed into XGBoost and Random forest classifiers in `train_classifier.py`. This script trains both classifiers and evaluates them using 5-fold cross-validation. The grid search for optimal classifier's parameters is also implemented in this file.

5.4.5 Patient-level classification's scripts

Patient-level classification, described in Section 5.3.4 is implemented in `5_patient_level_classification`, and the core of the classification is performed by `evaluate_c17.py`. This script makes the final patient's pN-stage prediction using features extracted from the tumour probability maps. After predicting the slide-level stage for every WSI using the classifier, slides belonging to one patient are aggregated, and pN-stage is predicted and stored to CSV file. The final score for training patients can be tested using `official_evaluation.py`. This script, provided by organizers, calculates inter-annotator agreement with quadratic weighted kappa for training slides.

5.4.6 Additional scripts

For running the above-mentioned scripts smoothly, some additional scripts are prepared. The `configuration.py` holds all the necessary configuration parameters needed for the whole framework to work properly. All possible parameters are controlled through this file. `utils.py`, `train_utils.py` and `plot_utils.py` are scripts providing some additional functionalities, which can be used within the whole pipeline. Some of the utilities are more general, for example, renaming a folder or confusion matrix plotting, and some may be used in an exact part of the pipeline, for example, defining the dataset or loss functions. `laplotter.py` generates a nice plot of the accuracy and loss curve generated during the training process. It is taken and edited from [69].

Feature	**Description**	**Max value**	**Mean value**
Area of the largest connected region	–	No	Yes
Length of the major axis of the largest connected region	Length of the major axis of the ellipse with the same second moments as the region	Yes	Yes
Perimeter of the largest connected region	–	Yes	Yes
Maximum pixel value of the largest connected region	–	Yes	Yes
Mean pixel value of the largest connected region	–	Yes	Yes
Eccentricity of the largest connected region	Ratio of the focal distance over the major axis length of the ellipse having the same second moments as the region	Yes	Yes
Extent of the largest connected region	Ratio of pixels in the region to pixels in the total bounding box	Yes	Yes
Solidity of the largest connected region	Ratio of pixels in the region to pixels of the convex hull	Yes	Yes
Number of connected regions in total	–	No	No
Area predicted as a tumour in total	–	No	No

Table 5.5: List of designed features extracted from the metastasis-likelihood maps for the purposes of classifiers training along with their descriptions (if they are needed for better understanding, taken from [68]). Most of the features are extracted from the largest connected region. To find it, we detect all connected components on a single WSI, compute the area they occupy, and select the largest one. If the feature has also *Yes* in the **Max value** column, additional feature with the maximum value selected from all detected tumouros objects within one slide is stored as well. If the feature has also *Yes* in the **Mean value** column, additional feature with mean value computed across all detected tumouros objects within one slide is stored as well.

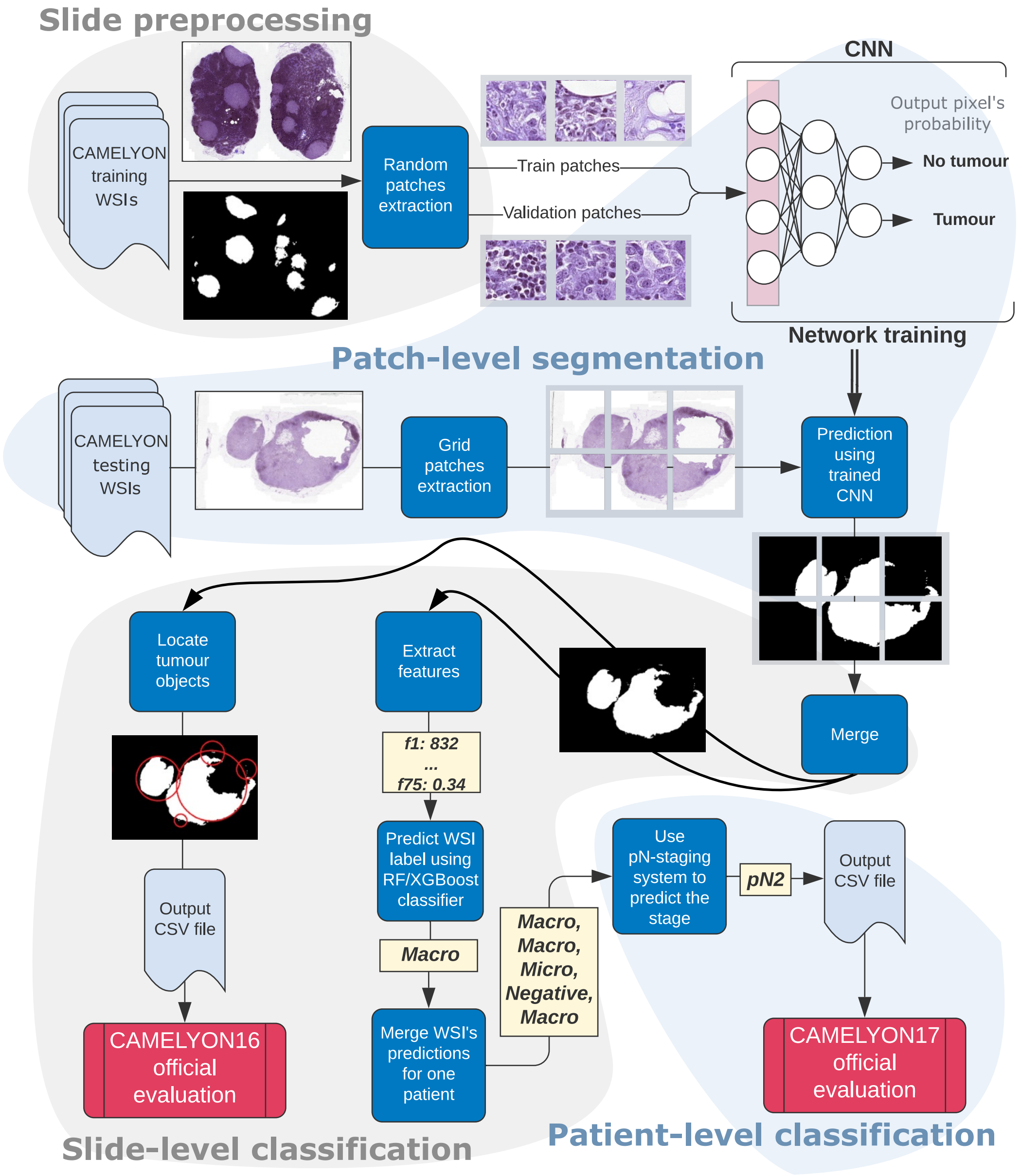

Slide preprocessing
CAMELYON training WSIs
Random patches extraction
Train patches
Validation patches
CNN
Output pixel's probability
No tumour
Tumour
Network training
Patch-level segmentation
CAMELYON testing WSIs
Grid patches extraction
Prediction using trained CNN
Merge
Locate tumour objects
Extract features
f1: 832
...
f75: 0.34
Predict WSI label using RF/XGBoost classifier
Macro
Merge WSI's predictions for one patient
Macro, Macro, Micro, Negative, Macro
Use pN-staging system to predict the stage
pN2
Output CSV file
Output CSV file
CAMELYON16 official evaluation
CAMELYON17 official evaluation
Slide-level classification
Patient-level classification

Figure 5.13: Visualization representing the whole framework for tumour detection from the slide preprocessing to the final CAMELYON16 and CAMELYON17 evaluation. First, patches for the training process are prepared and network is trained. After that, we select a slide from testing dataset, cut it into patches and run through the trained network. Outputted segmented tumour regions are merged into one big tumour probability map belonging to the input WSI. Map is then directed for the slide-level and patient-level classification for CAMELYON16 and CAMELYON17 purposes.

Chapter 6

Experiments and results

6.1 Evaluation metrics

In this section, we declare all the metrics used for the evaluation of trained models and classifiers.

6.1.1 Sensitivity, specificity, precision

First, we need to define some essential concepts to construct more complicated metrics.

Suppose we have a binary classification problem. Given a classifier and a sample, there are four possible outputs. If the sample is positive and is classified as positive, we call it a true positive (TP). If it is classified as negative, we call it a false negative (FN). If the sample is negative and is classified as negative, we call it a true negative (TN). If it is classified as positive, we call it a false positive (FP) [70].

Sensitivity (recall, TP rate) is defined as

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \tag{6.1}$$

Specificity (selectivity, TN rate) is defined as

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{6.2}$$

Precision (positive predictive value) is defined as

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{6.3}$$

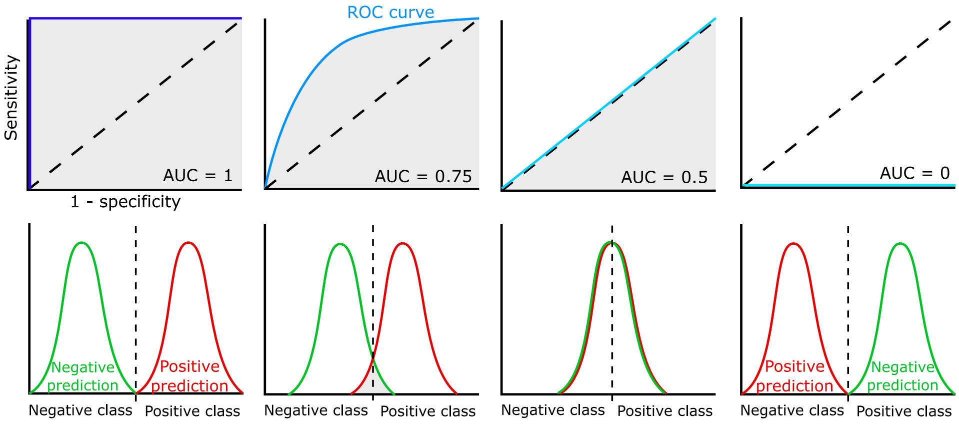


Figure 6.1: Visualization of ROC curves and corresponding AUC scores. The AUC = 1 means a perfect classifier, whereas A = 0 means that the classifier assign the opposite label from a true class.

Intersection over union (IoU) is defined as

$$\mathrm{IoU} = \frac{TP}{TP + FP + FN}. \tag{6.4}$$

The definition of Dice coefficient is the same as in Equation 5.2.

6.1.2 ROC

The ROC is a two-dimensional graph showing the performance of a classification model at all classification thresholds. It is plotted as sensitivity on the Y axis against $1-$ specificity on the X axis. It tells how much our model can distinguish between classes [70].

AUC is the area under the ROC curve. It is equal to the probability that a classifier will rank a randomly chosen positive sample as positive [71]. The connection between ROC and AUC is shown in Figure 6.1.

This metrics is used for evaluating the Histopathological cancer detection challenge by Kaggle submissions.

6.1.3 FROC

This metrics is similar to ROC analysis. The only difference is that the $1-$ specificity on the X axis is replaced by the average number of false positives per sample [72].

This metrics is used evaluating the CAMELYON16 submissions. The final score obtained in this challenge is defined as the average sensitivity at six predefined FP rates: $\frac{1}{4}, \frac{1}{2}, 1, 2, 4$ and 8 FPs per whole slide image. All detections further than a specific distance from the ground truth annotations ale counted as FPs. If multiple detections for a single tumour are obtained, they are counted as single TP finding. None of the detections is counted as FP [11].

6.1.4 Quadratic weighted kappa

Given n test samples and m categories, we denote n_{ij} as the number of samples from the i-th category assigned to the j-th category, r_i as the total number of samples from i-th category, s_j as the total number of samples assigned to j-th category, and w_{ij} as the disagreement weight associated with i-th and j-th categories [11]. Then, we can define the weight matrix as

$$w_{ij} = (i - j)^2, \quad i, j \in 1 \dots m, \tag{6.5}$$

the mean observed degree of disagreement as

$$D_o = \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{m} n_{ij} w_{ij}, \tag{6.6}$$

and the mean degree of disagreement expected by chance as

$$D_e = \frac{1}{n^2} \sum_{i=1}^{m} \sum_{j=1}^{m} r_i s_j w_{ij}. \tag{6.7}$$

The final quadratic weighted kappa (QWK) is defined as

$$\kappa_w = \frac{D_e - D_o}{D_e}. \tag{6.8}$$

The κ_w ranges from -1 to $+1$, where a negative value means lower than chance agreement, zero means exact chance agreement, and a positive value means higher than chance agreement [11].

For evaluating the CAMELYON17 submissions, five class QWK, where the classes are the pN-stages, is used.

6.2 Baseline solution for the purposes of Kaggle competition

In this section, we analyse results obtained from the CNN training for purposes of Kaggle challenge, as proposed in Section 5.2. We train the ResNet-50 model for 40 epochs. We can observe accuracy and loss curves belonging to this training process in Figure 6.2. Finalising the model after 40 epochs in

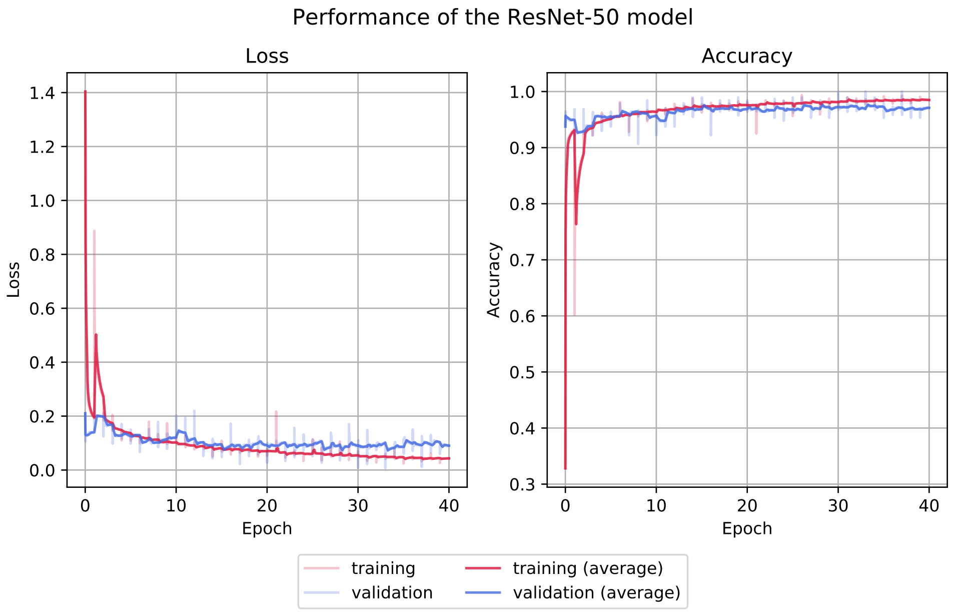


Figure 6.2: Learning curves presenting the performance of the ResNet-50 model.

training gives us 97.05% accuracy and 0.091 loss for the validation dataset. Learning takes approximately 26 hours.

Analysing the loss curve, we can see that our training and validation loss is rapidly decreasing until stabilising to a steady range of values after a few epochs. After that, both of them are slowly decreasing while the training progresses. There are no signs of significant overfitting or underfitting.

Analysing the graph of training and validation accuracy curve, we can see that the validation accuracy separates from the training one a bit at the end of the training. This means that our model is slightly overfitted in the last few epochs. If we would train further, the model would only memorise features from the training set [56].

From observing the curves, the best result of training is somewhere around epoch 33. After this epoch, validation accuracy starts to decrease slowly. Finalising the model after the epoch 33 in training gives us 97.44% accuracy and 0.084 loss for the validation dataset. Learning takes approximately 23 hours. A confusion matrix for the model is shown in Figure 6.3.

The challenge's submissions are evaluated on the area under the ROC curve. To get a grasp of what roughly we can expect after uploading the results file to the official evaluation, we first calculate the AUC score for the model evaluated on the validation part of the dataset on our own. The AUC score for the validation part of the dataset is 0.995 and calculated ROC curve is shown in Figure 6.7

To obtain official AUC score for test dataset, we run a model evaluation

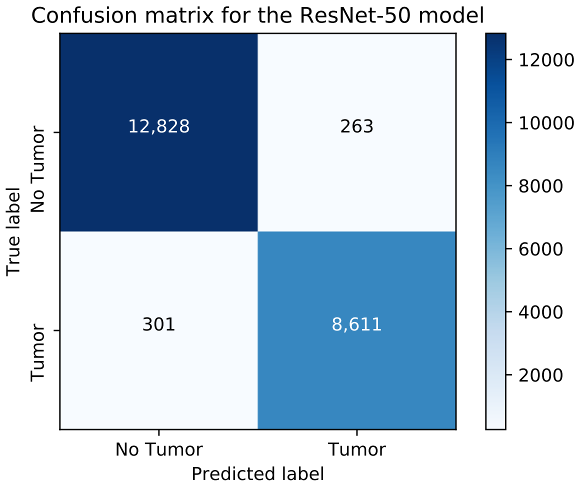


Figure 6.3: Confusion matrix for the ResNet-50 model tested on the validation part of the PCam dataset.

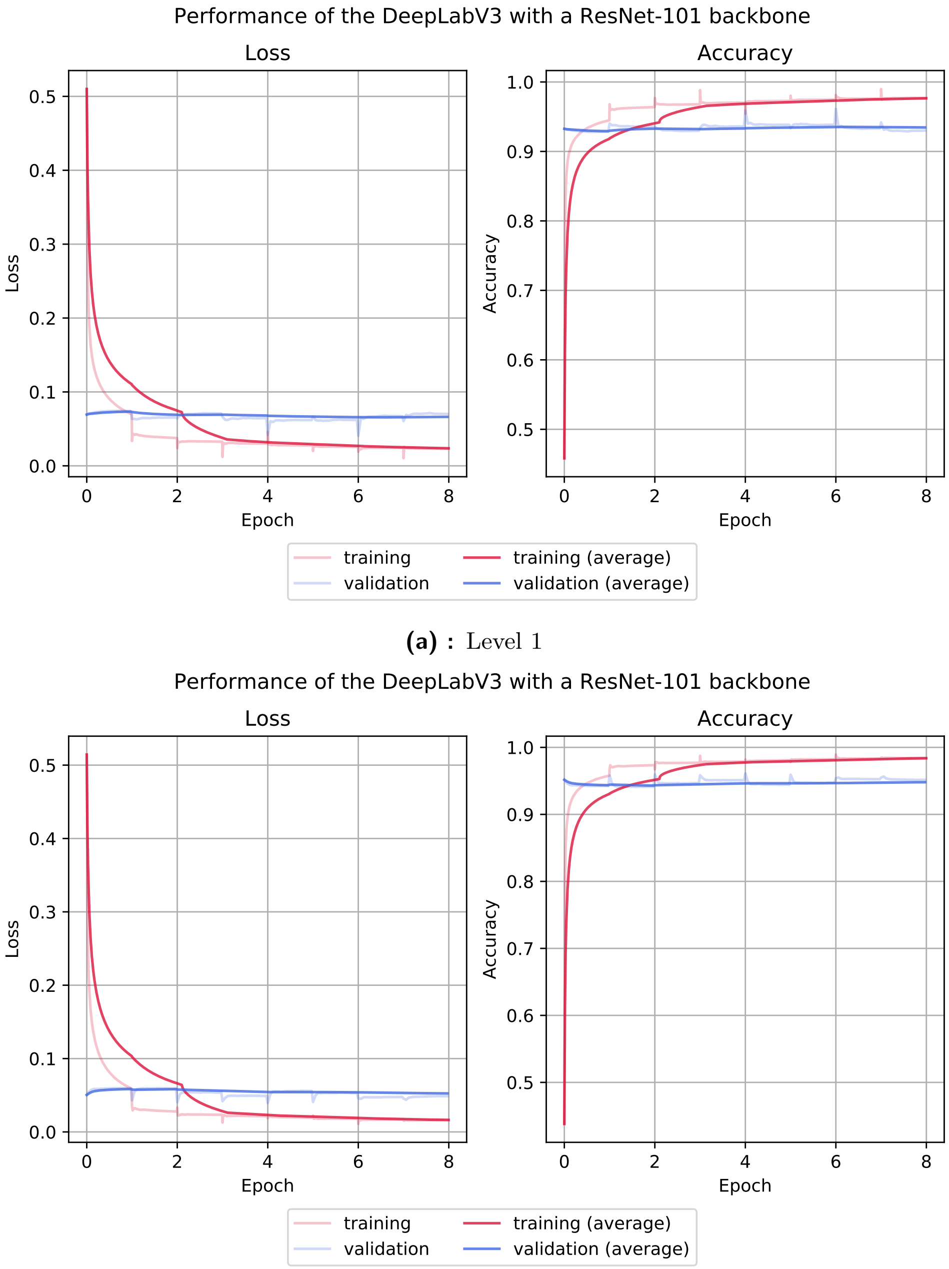
over all test patches and prepare the submission file. After uploading it to the official evaluation web page, the obtained final AUC score computed on the test dataset is 0.952.

6.3 Extended solution for the purposes of CAMELYON competitions

In this section, we analyse experiments and results obtained in order to perform metastasis detection tasks as described in CAMELYON competitions. Following experiments correspond with the methods described in Section 5.3.

6.3.1 Patch-level segmentation

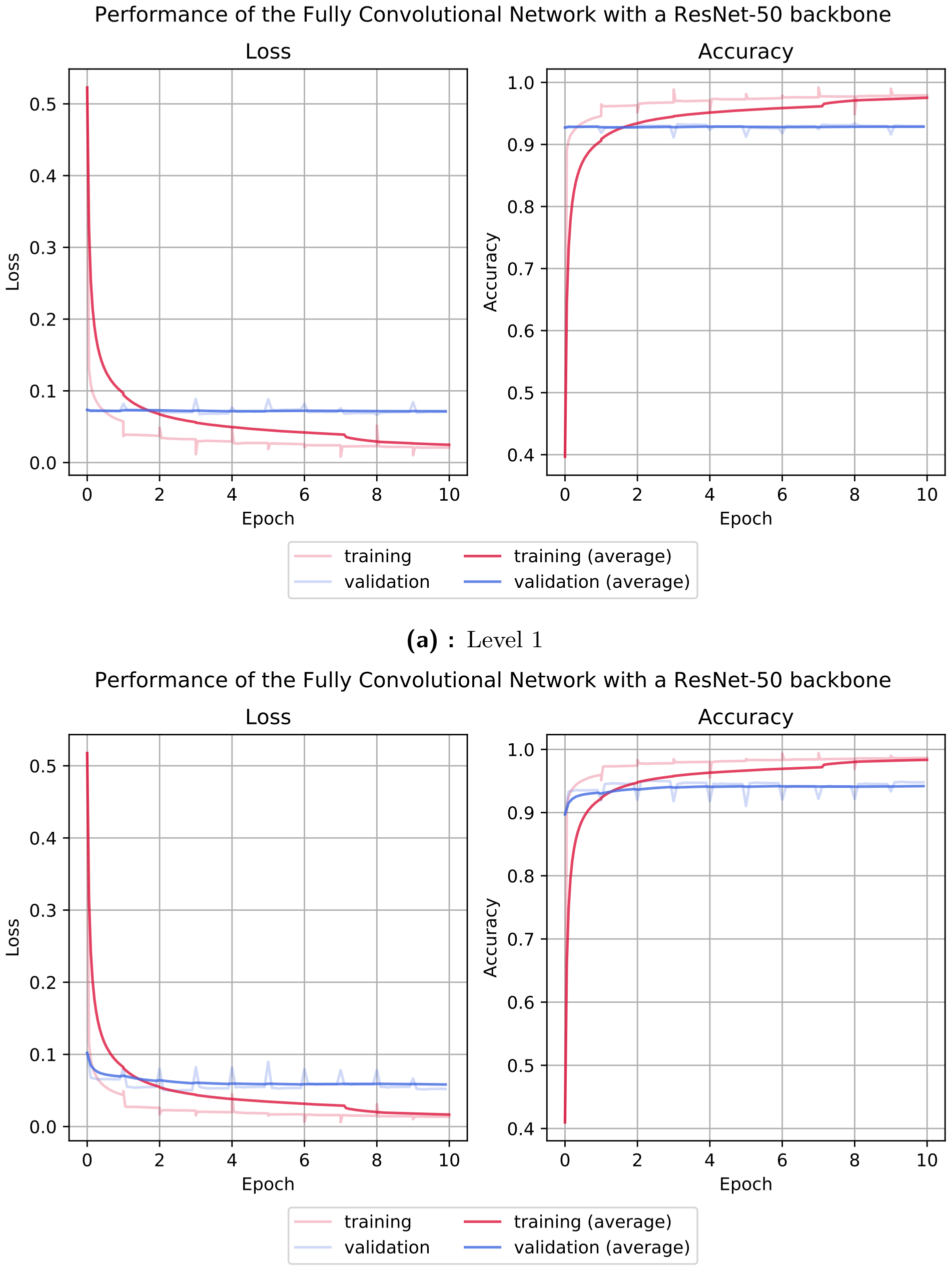
To segment tumours on patch level, we prepare the DeepLabV3 model and train it for eight epochs. The accuracy and loss curve belonging to this training process can be observed in Figure 6.4. We also suggest comparing the proposed model with other well-known CNN architectures for the task of segmentation. For that purpose, the FCN model and UNet model are simultaneously trained for ten epochs. Their accuracy and loss training curves may be observed in Figures 6.5 and 6.6.



(a) : Level 1

(b) : Level 2

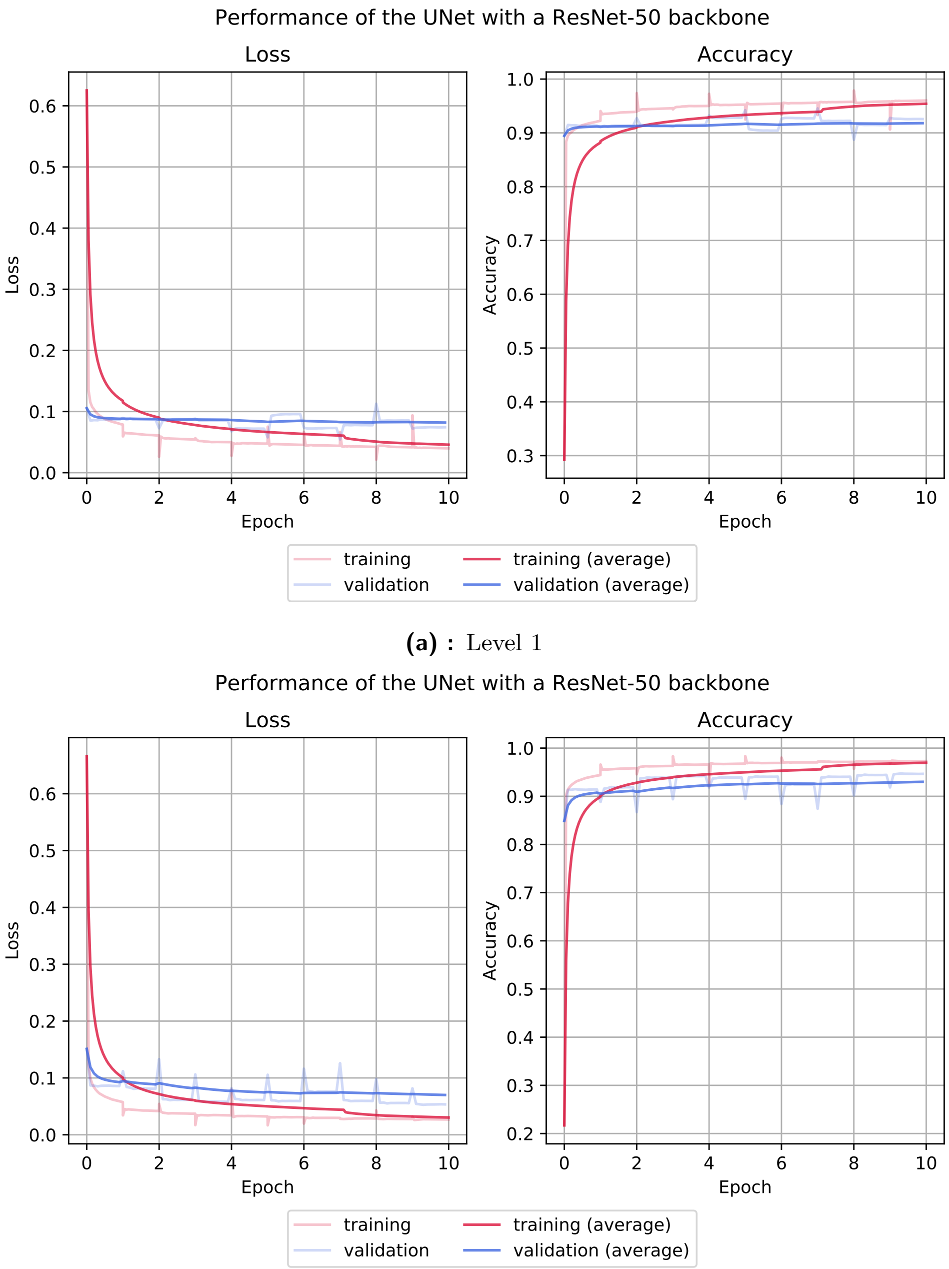
Figure 6.4: Learning curves presenting the performance of the DeepLabV3 with a ResNet-101 backbone.



(a) : Level 1

(b) : Level 2

Figure 6.5: Learning curves presenting the performance of the FCN with a ResNet-50 backbone.



(a) : Level 1

(b) : Level 2

Figure 6.6: Learning curves presenting the performance of the UNet with a ResNet-50 backbone.

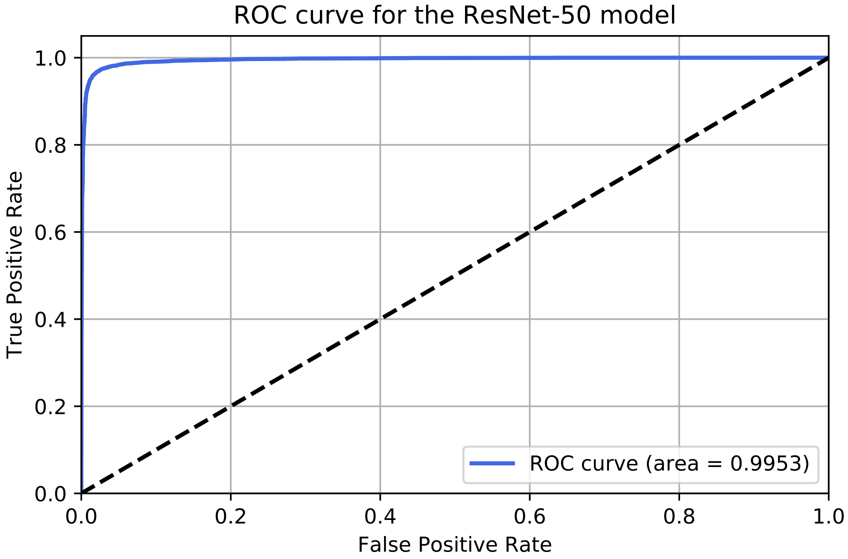


Figure 6.7: ROC curve with its AUC score computed for the trained ResNet-50 model evaluated on the validation data.

By analysing these three graphs, we can see that all three models follow a similar style of learning. They learn very quickly to a certain value and then stagnate for the rest of the training process. Also, training and validation curves begin to separate in the early part of the training for all three models. That is a sign of overfitting.

Since in later epochs our models may be overfitted, we examine results of the whole training process and try to find an epoch with the best performance. Detailed results of this analysis for all three models are described in detail in Table 6.1. From the results summarised in this table, it is evident that we achieve the best results in all aspects with the DeepLabV3 with a ResNet-101 backbone and we decide to use it in the rest of the pipeline. Figure 6.9 illustrates some random samples of patch-level predictions performed by the DeepLabV3 model.

6.3.2 Slide-level and patient-level classification

After the patch-level segmentation, we use DeepLabV3 model to aggregate patch-level predictions. An example of the aggregated metastasis-likelihood map for a WSI from the CAMELYON16 dataset shows Figure 6.10. We repeat the aggregating process for two patches resolutions – level 1 and level 2, to verify which resolution works better.

After patch-aggregating process, we use generated tumour probability maps to perform the **lesion-based detection task as required by CAMELYON16 challenge** and described in Section 5.3.3. For all CAMELYON16

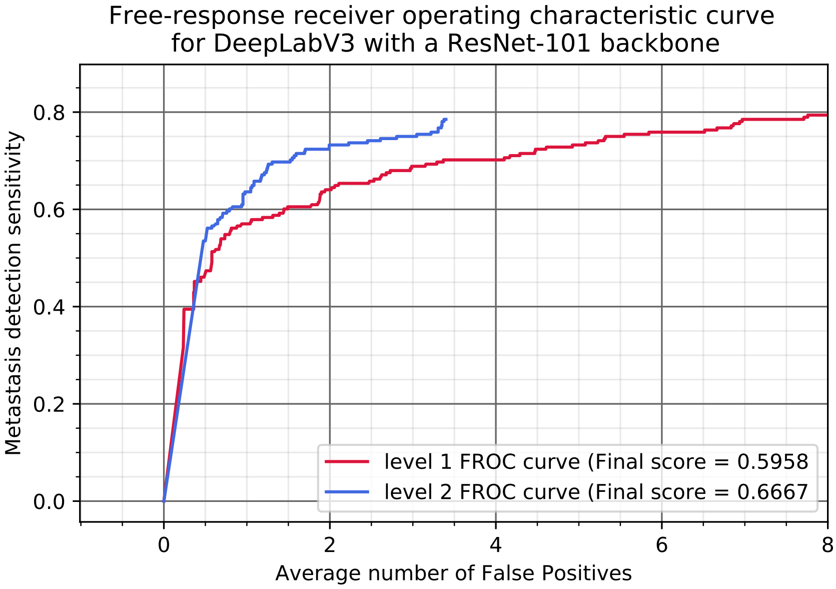


Figure 6.8: FROC curves with their final scores computed for the trained DeepLabV3 with a ResNet-101 backbone evaluated on the testing part of CAMELYON16 dataset.

test slides, tumour objects are detected and stored to CSV files. Then, we use the official evaluation script provided by organisers to obtain the final FROC score. After running this script, the obtained final FROC score computed on CAMELYON16 test slides is 0.5958 for level 1 and 0.6667 for level 2. The evaluation script also generates an FROC curve corresponding to the obtained FROC score. For both levels, this FROC curve is visualized in Figure 6.8.

To perform the **slide-based classification for the purposes of CAMELYON17 challenge** as described in Section 5.3.3, we train two classifiers, Random forest and XGBoost. Both classifiers are trained using the tumour probability maps generated from patches extracted at level 2, as this level works better in the lesion-based detection task evaluated in the previous paragraph.

Results pertaining to the classifiers training process are summarised in Table 6.2. From this table, we can observe that both classifiers achieve comparable accuracy values for training data, but there is a markable difference between QWK of these two classifiers. It may be a sign of overfitting. To confirm this assumption, we make two uploads to the official evaluation system – one with the Random forest classifier and one with theXGBoost classifier.

Confusion matrix for each fold of the Random forest's K-fold cross-validation process are imaged in Figure C.1. Confusion matrix for each fold of the XGBoost's K-fold cross-validation process are imaged in Figure C.2. We can observe from these matrices that both classifiers perform poorly in distinguishing the ITC cases from negative cases. Almost all slides containing ITCs

	DeepLabV3		**FCN**		**UNet**	
Parameter	level 1	level 2	level 1	level 2	level 1	level 2
Time per epoch	8.5 hours		8 hours		7.5 hours	
Best epoch	5.	7.	4.	4.	5.	10.
Training accuracy	97.271	**98.333**	97.070	97.808	95.247	97.315
Validation accuracy	93.820	**95.259**	93.175	94.962	92.806	94.632
Training loss	0.028	**0.017**	0.029	0.022	0.048	0.027
Validation loss	0.062	**0.048**	0.068	0.051	0.072	0.054
Validation sensitivity	**0.8812**	0.8634	0.8510	0.8435	0.8503	0.8513
Validation specificity	0.9635	**0.9796**	0.9675	0.9782	0.9625	0.9751
Validation precision	0.9144	**0.9276**	0.9207	0.9213	0.9094	0.9118
Validation IoU	**0.8141**	0.8089	0.7929	0.7869	0.7840	0.7866
Validation Dice coefficient	**0.8975**	0.8944	0.8845	0.8807	0.8789	0.8806

Table 6.1: Results of the CNN training processes for all three models and patches sampled at level 1 and level 2. Apart from the accuracy and loss values, we also provide other metrics to ensure unbiased model comparison. The first part of the table provides information about training time and the epoch from which the results are obtained. For each parameter, the column with the best result is marked in bold.

Parameter	**Random forest**	**XGBoost**
Training accuracy	0.8500	0.8540
Training QWK	0.8934	0.9715

Table 6.2: Results of the Random forest and XGBoost training processes. As we use K-fold cross-validation for the evaluating, training accuracy is computed as a mean over all folds.

are classified as slides with no tumour. Apart from that, both classifiers seem to work pretty well.

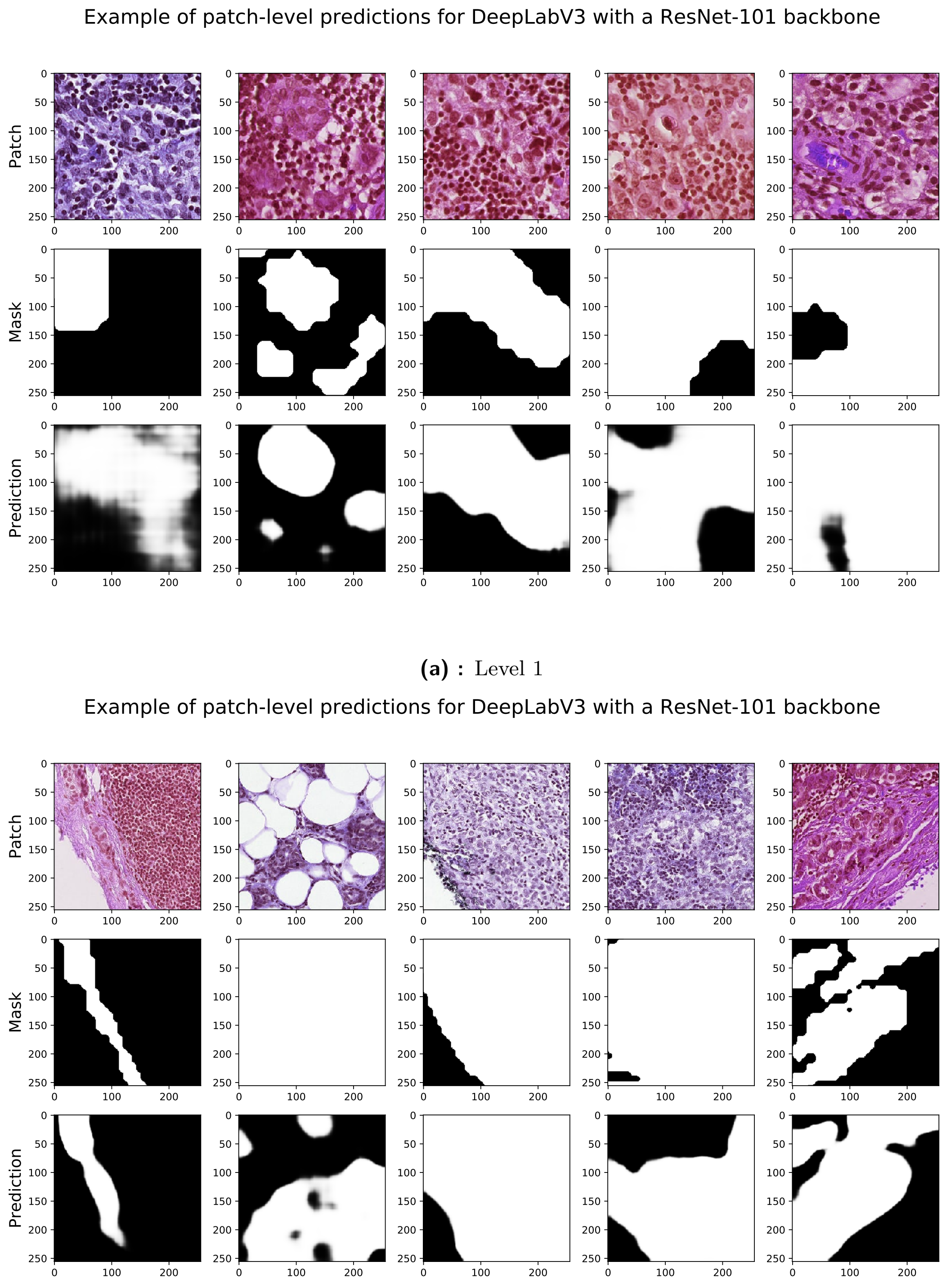
To perform the **patient-based classification for the purposes of CAMELYON17 challenge** as described in Section 5.3.3, we aggregate predictions made by two trained classifiers, Random forest and XGBoost, for one patient and make the final pN-stage prediction. We store predictions for all patients in CAMELYON17 test dataset and generate the final CSV file. For the evaluation of the results, organisers use the five-class QWK. To obtain official QWK score for test dataset, we run patient-level classification over all test patients and prepare two submission CSV files – one using the Random forest as the slide-level classifier, and one using the XGBoost as the slide-level classifier. After uploading it to the official evaluation web page, the obtained QWK score computed on the test dataset is 0.8381 for the pipeline with Random forest, and 0.8457 for the pipeline with XGBoost. As we see, we obtain better results with XGBoost classifier.

6.4 Comparison with the state-of-the-art

In this section, we compare obtained scores with other participants of the above-mentioned challenges.

- For the Kaggle challenge, we obtained a ROC score of 0.9519, which puts our solution on 509. place out of 1149 submissions. The score ranges from 1.0000 to 0.3080 for all 1 149 participants.
- For the CAMELYON16 challenge's lesion-based detection task, we obtained an FROC score of 0.6670, which puts our solution on 6. place out of 32 submissions. The score ranges from 0.8070 to 0.0970 for all 32 participants.
- For the CAMELYON17 challenge's patient-based classification, we obtained a quadratic weighted kappa score of 0.8457, which puts our solution on 28. place out of 102 submissions. The score ranges from 0.9750 to −0.2203 for all 102 participants.

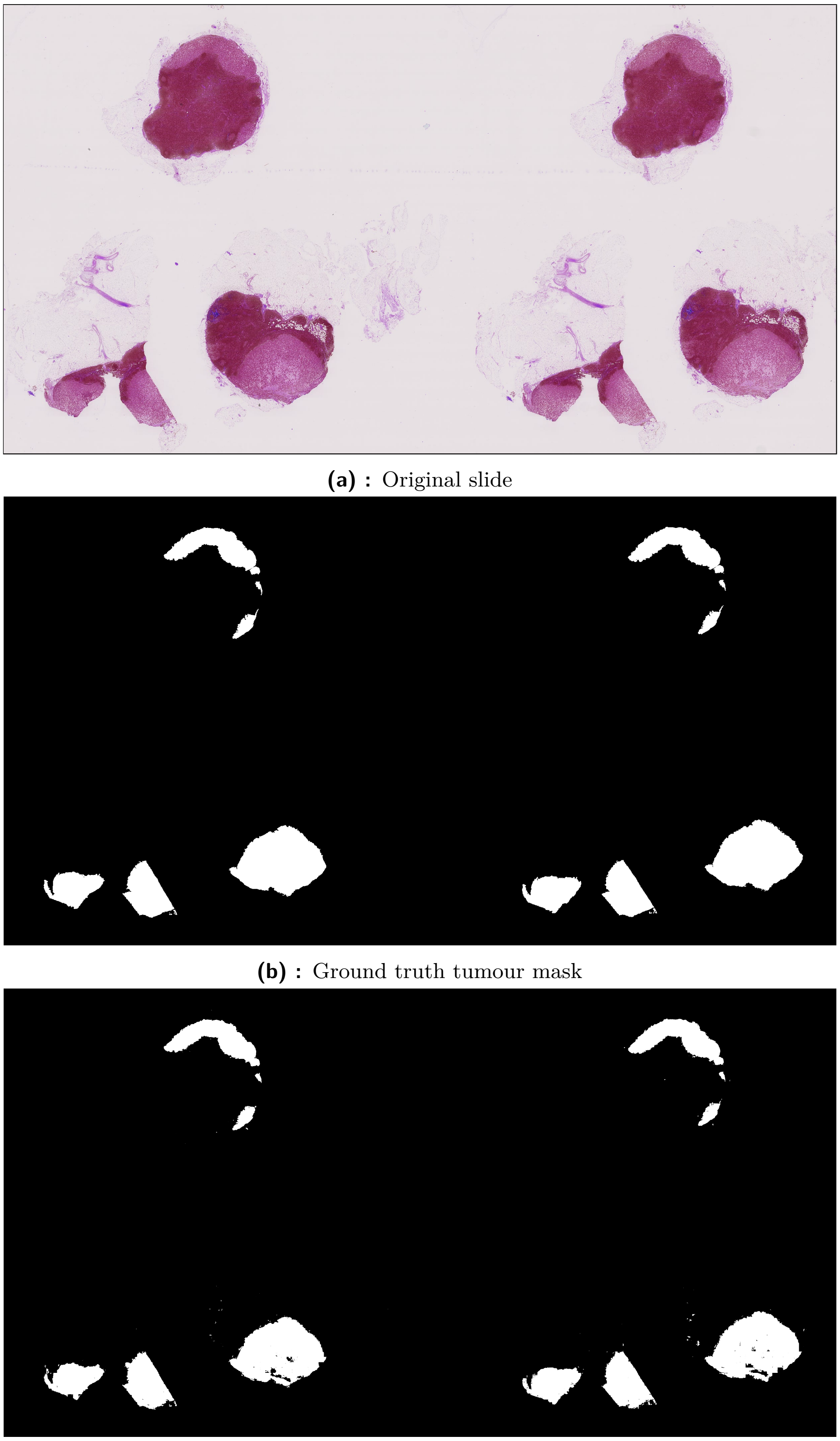
From the obtained results is evident, that our tumour detection system is competitive among other participants.


Example of patch-level predictions for DeepLabV3 with a ResNet-101 backbone
Patch
Mask
Prediction
0
50
100
150
200
250
0
100
200
Example of patch-level predictions for DeepLabV3 with a ResNet-101 backbone
Patch
Mask
Prediction

(a) : Level 1

(b) : Level 2

Figure 6.9: A radom sample of patch-level predictions for DeepLabV3 with a ResNet101 backbone.



(a) : Original slide

(b) : Ground truth tumour mask

(c) : Predicted tumour mask

Figure 6.10: Comparison of a WSI's ground truth tumour mask from the CAMELYON dataset and our generated tumour mask using the DeepLabV3 model.

Chapter 7

Discussion

In this chapter, we try to analyse the developed algorithm, clarify its behaviour and investigate its weaknesses and strangenesses.

7.1 Encountered problems

This section summarises the major problems we encountered while developing and evaluating our tumour detection framework.

7.1.1 Inaccurate pixel-wise tumour annotation

CNN model is often confused in places that are not precisely marked by pathologists themselves. For example, large formations of adipose cells are often inaccurately marked as tumour tissue region. The fat cells might be overgrown with the tumour or are located in the immediate vicinity, and it is understandable that pathologists cannot label tumours in such detail. However, the model is often confused in evaluating these regions. Example of the described problem is shown in Figure 7.1

7.1.2 Poor ITC detection

Confusion matrices presented in Appendix C reveals that our model struggles with identifying isolated tumour cells. Almost all slides containing only ITCs are labelled as negative. That may be caused by a lack of training slides with the ITC label. In combination with the random patch-sampling strategy, we might not obtain enough training samples from the ITC's regions.

In the clinical practice, lymph nodes containing only ITC are not counted as positive lymph nodes. Additionally, they are often missed by pathologists

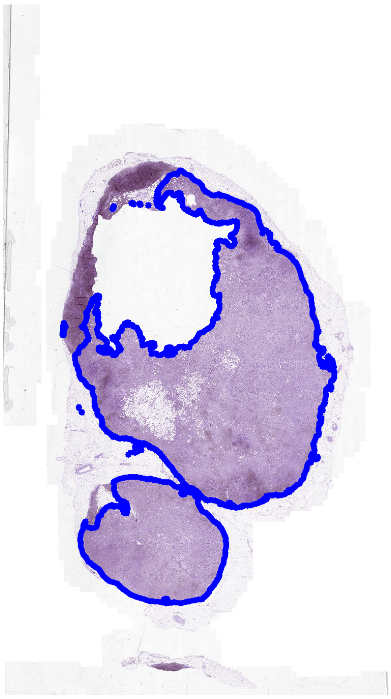


Figure 7.1: The weakness of a pixel-wise annotation are scattered tumour cells. In this case, tumour annotation also contains non-tumour cells, mainly adipose tissue cells (white cells inside the bordered tumour). While training, these false positive cases might lead to reduced performance of our model.

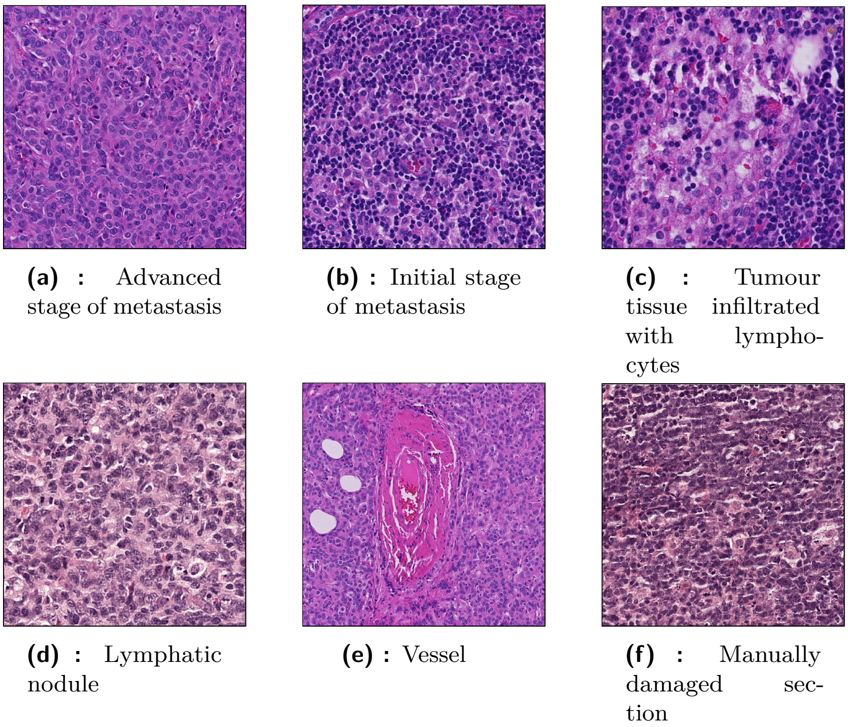
too. The interesting thing is that cancer cells, in small amounts, are common in the human body. It is a natural process of the organism, which the body can solve on its own without any further supportive treatment. Therefore, poor ITC detection is not a major problem if we want to bring this system to clinical practice. However, ITC regions still might be important for early cancer detection in some cases, and pathologists have to report them if no micro or macro-metastases are found [11].

We suggest performing additional patch-extraction process from the regions containing ITCs. This strategy is called *hard example mining*. Using it, the model can be trained more effectively using important samples from the most misclassified areas.

7.1.3 Misclassified regions

There are some WSI's regions that were often misclassified during the patch-level segmentation by our DeepLabV3 model. We can summarise them into the following categories:

- Tumour tissue infiltrated with lymphocytes – In practice, this means that the body has started a defensive reaction and is trying to fight the



(a) : Advanced stage of metastasis

(b) : Initial stage of metastasis

(c) : Tumour tissue infiltrated with lymphocytes

(d) : Lymphatic nodule

(e) : Vessel

(f) : Manually damaged section

Figure 7.2: Samples of often misclassified tissue regions. For example, the tissue of lymphatic nodule is easily interchangeable with the advanced stage of metastasis even though it is a healthy section. In opposite to that, initial stage of metastasis might confuse our model, because is infiltrated with a lot of healthy cells.

tumour (and might be connected to better prognosis as well). The tissue at that moment does not even look healthy and does not even have a typical tumour structure. Our model cannot handle it very much.

- Lymphatic nodules – Lymph nodules should be classified as healthy tissue, but the model often confuses them with a tumor. That may be caused due to the tissue structure, which is similar to the tumourous at first glance.

- Initial stage of metastasis – model fails to detect tumour cells that are not inside a large mass of overgrown metastasis but are more of early metastasis. These cells are heavily diffused with healthy cells and are very difficult to distinguish.

- Manually damaged sections – Some images suffer from poor technical processing, for example, a poorly executed cut. Such unusualness can then confuse our model.

- Contaminations, vessels, nerves – These are unique finds that do not appear very much in the training dataset. The model then has the problem to segment them correctly.

Above listed often misclassified regions are shown in Figure 7.2. To improve model performance on listed areas, we suggest, same as in the case of poor ITC detection problem, to perform an additional patch-extraction process.

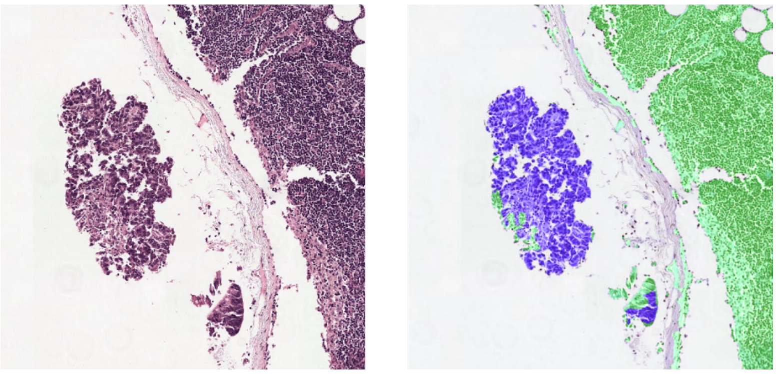


Figure 7.3: Comparison of a detail extracted from the original slide and the output prediction of our model. Green stands for TN, dark blue stands for FP. According to the official annotation, the dark blue section should be a no-tumour tissue. However, with medical assistance, we revealed that our model's prediction is right and the tissue metastasic.

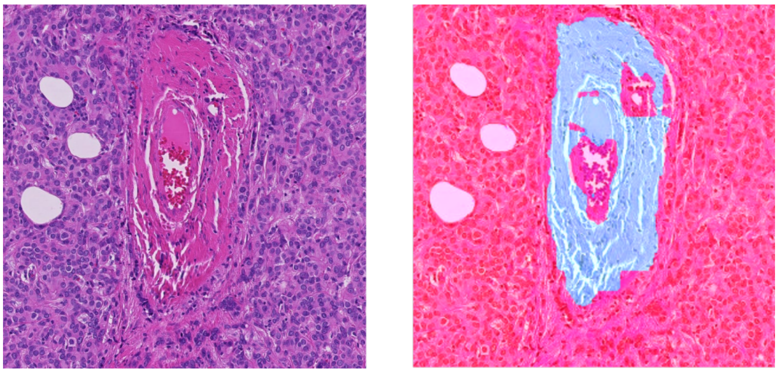


Figure 7.4: Comparison of a detail extracted from the original slide and the output prediction of our model. Red stands for TP and light blue for FN. According to the official annotation, the light blue section should be a tumour tissue. However, with medical assistance, we revealed that our model's prediction is right and it is a vessel.

7.2 Results analysis

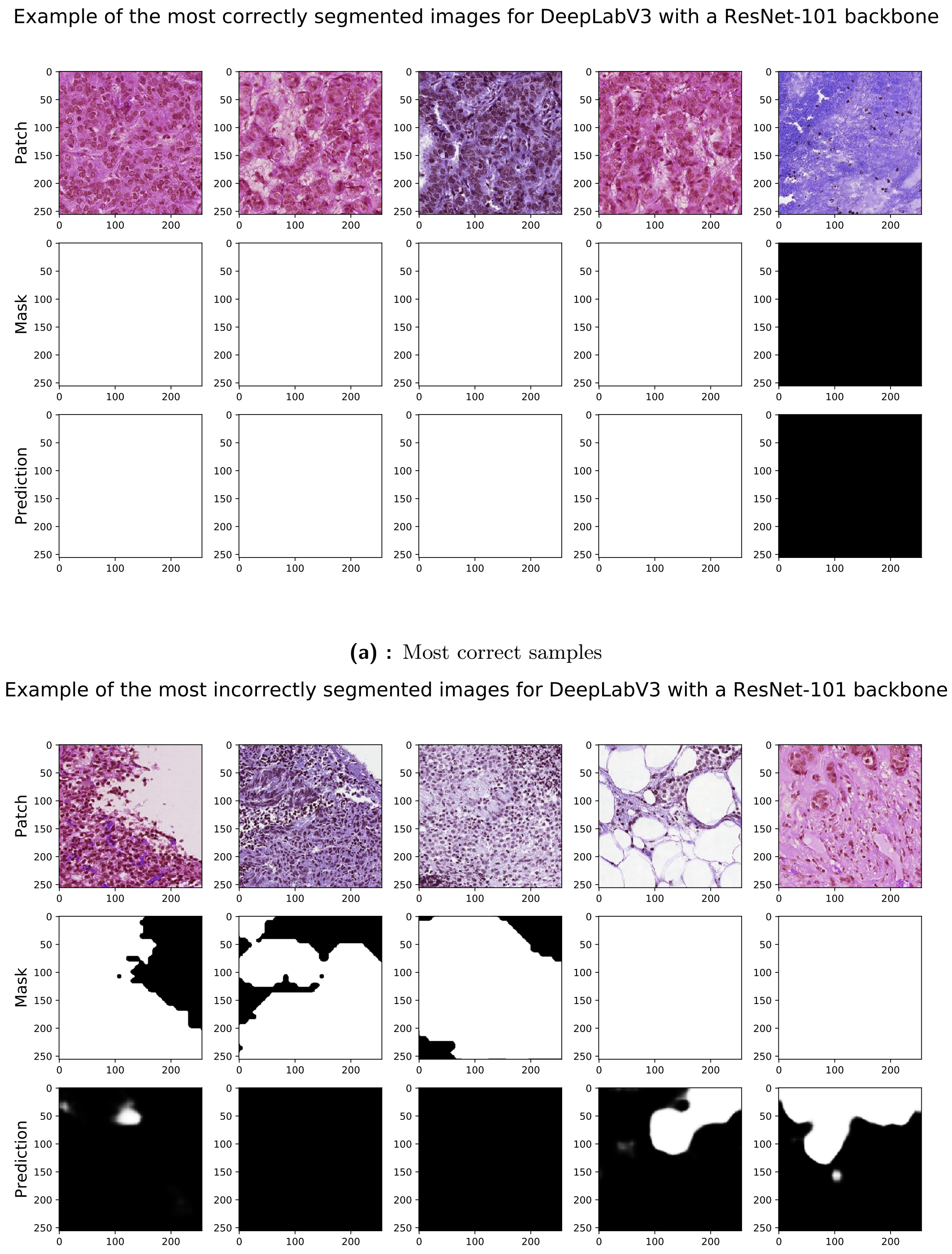
In this section, we perform the analysis of the final results with some additional notes.

Observing the confusion matrices presented in Appendix C, we notice that our algorithm is highly capable of detecting macrometastases, and poorly performs in ITC detecting. As we said before, poor ITC detection is not a major problem if we want to bring this system to clinical practice. However, misclassification of any of the patient's micro or macrometastasis could have fatal consequences in clinical conditions. Taking this fact into account, there is still room for improvement, even though the detection of micro and macrometastases is pretty accurate.

We also notice that results for the patch-based segmentation, presented in Table 6.1 are better for the model trained with the patches extracted from the 4-times downsampled WSIs than for the one trained with patches extracted from the 2-times downsampled WSIs. This can be caused by the fact that the smaller patches from level 1 cannot capture a wider context of the surroundings. Visualisation of the DeepLabV3 final prediction with metrics obtained at both level is visualised in Figure D.2.

We can observe an example of most correctly and incorrectly segmented patches for the DeepLabV3 model trained on patches from level 2 in Figure 7.5. Generally, the trained model is very confident in segmenting fully tumorous areas.

During the detailed results examination, we discovered some interesting facts. For example, our algorithm is capable of detecting a tumour that is not officially annotated, as shows Figure 7.3. In Figure 7.4, the model also correctly segments a vessel as non-tumour tissue even though the official annotations declares it as a tumour. More examples with full-slide visualisation of the model's performance projected into detailed metrics maps presents Figure D.1.



Example of the most correctly segmented images for DeepLabV3 with a ResNet-101 backbone

(a) : Most correct samples

Example of the most incorrectly segmented images for DeepLabV3 with a ResNet-101 backbone

(b) : Most incorrect samples

Figure 7.5: Most correctly and incorrectly segmented patches using DeepLabV3 with a ResNet101 backbone and dataset with patches on level 2.

Chapter 8

Conclusion

In this thesis, we proposed a method for solving the task of the detection of metastases in whole-slide lymph node images using deep convolutional neural networks. To achieve that, we designed an end-to-end system for cancer detection and tumour segmentation.

We created a baseline solution for patch classification with the ResNet-50 architecture. Using the results of the model's training, we further extended the tumour detection task. We switched from the classification task to the segmentation one and aimed to the full-slide tumour segmentation. To achieve that, we proposed to train three segmentation networks, FCN, UNet and DeepLabV3, and choose the best performing one. Most promising results for the patch-level segmentation were obtained using the DeepLabV3 model trained on patches extracted from 4-times downsampled version of original WSIs. For this reason, we used this model's output predictions in the rest of the pipeline.

After the process of patch segmentation, we aggregated the predictions to obtain a full-slide tumour prediction and perform the slide-level classification. To resolve that for the purposes of CAMELYON16 challenge, we created an algorithm for object detection, which localises tumours and store their location. For purposes of CAMELYON17 challenge, we trained two machine-learning classifiers, Random forest and XGBoost, to predict the slide-level label. According to the evaluated results, XGBoost classifier outperformed Random forest.

The slide-level classification was the primary goal of this thesis. However, we decided to extend this pipeline with patient-level classification. The main motivation for this extension was the possibility of involving the system that predicts the patient pN-stage in clinical practice. A pipeline of this format may partially replace the time-consuming routine that a pathologist has to perform to obtain the final pN-stage label for the patient.

We actively participated in Kaggle's Histopathological cancer detection challenge and CAMELYON17 challenge and passively participated in CAMELYON16 challenge using the official evaluation script provided by the organisers and evaluating the submission ourselves. Our results were compared with other participants. For the Kaggle challenge, we obtained a ROC score of 0.9519, which puts our solution on 509. place out of all 1149 submissions. For the CAMELYON16 challenge's lesion-based detection task, we obtained an FROC score of 0.667, which puts our solution on 6. place out of 32 submissions. For the CAMELYON17 challenge's patient-based classification, we obtained a quadratic weighted kappa score of 0.8457, which puts our solution on 28. place out of 102 submissions.

Implementation of the proposed method resulted in a repository for the task of tumour detection with 20 files and more than 5 500 lines of code. Prepared scripts cover the whole pipeline, from slide preparation to patient-level classification, and may be reused in any related task.

Bibliography

[1] World Health Organization, "Czechia Fact Sheets," p. 2, 2018. [Online]. Available: https://gco.iarc.fr/today/data/factsheets/populations/203-czechia-fact-sheets.pdf (Accessed 2020-06-19).

[2] C. K. e. Howlader N, Noone AM, Krapcho M, Miller D, Brest A, Yu M, Ruhl J, Tatalovich Z, Mariotto A, Lewis DR, Chen HS, Feuer EJ, "SEER Cancer Statistics Review, 1975-2017," 2020. [Online]. Available: https://seer.cancer.gov/csr/1975_2017/ (Accessed 2020-06-19).

[3] L. He, L. R. Long, S. Antani, and G. R. Thoma, "Histology image analysis for carcinoma detection and grading," *Computer Methods and Programs in Biomedicine*, vol. 107, no. 3, pp. 538–556, September 2012.

[4] P. J. Van Diest, C. H. Van Deurzen, and G. Cserni, "Pathology issues related to sn procedures and increased detection of micrometastases and isolated tumor cells," *Breast Disease*, vol. 31, no. 2, pp. 65–81, January 2010.

[5] G. Aresta, T. Araújo, S. Kwok, S. S. Chennamsetty, M. Safwan, V. Alex, B. Marami, M. Prastawa, M. Chan, M. Donovan, G. Fernandez, J. Zeineh, M. Kohl, C. Walz, F. Ludwig, S. Braunewell, M. Baust, Q. Dang Vu, M. Nguyen, N. To, E. Kim, J. T. Kwak, S. Galal, V. Sanchez-Freire, N. Brancati, M. Frucci, D. Riccio, Y. Wang, L. Sun, K. Ma, J. Fang, I. Kone, L. Boulmane, A. Campilho, C. Eloy, A. Polónia, and P. Aguiar, "BACH: grand challenge on breast cancer histology images," Tech. Rep., 2019. [Online]. Available: https://list.ku.dk/listinfo/sci-diku-imageworld

[6] M. Veta, Y. J. Heng, N. Stathonikos, B. E. Bejnordi, F. Beca, T. Wollmann, K. Rohr, M. A. Shah, D. Wang, M. Rousson, M. Hedlund, D. Tellez, F. Ciompi, E. Zerhouni, D. Lanyi, M. Viana, V. Kovalev, V. Liauchuk, H. A. Phoulady, T. Qaiser, S. Graham, N. Rajpoot, E. Sjöblom, J. Molin, K. Paeng, S. Hwang, S. Park, Z. Jia, E. I.-C. Chang, Y. Xu, A. H. Beck, P. J. Van Diest, and J. P. W. Pluim, "Predicting breast

tumor proliferation from whole-slide images: the TUPAC16 challenge," Tech. Rep.

[7] "Prostate cANcer graDe Assessment (PANDA) Challenge | Kaggle." [Online]. Available: https://www.kaggle.com/c/prostate-cancer-grade-assessment/overview/description (Accessed 2020-06-19).

[8] "Histopathologic Cancer Detection | Kaggle." [Online]. Available: https://www.kaggle.com/c/histopathologic-cancer-detection/overview (Accessed 2020-06-20).

[9] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling, "Rotation equivariant CNNs for digital pathology," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11071 LNCS. Springer Verlag, 2018, pp. 210–218.

[10] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. Van Der Laak, M. Hermsen, Q. F. Manson, M. Balkenhol, O. Geessink, N. Stathonikos, M. C. Van Dijk, P. Bult, F. Beca, A. H. Beck, D. Wang, A. Khosla, R. Gargeya, H. Irshad, A. Zhong, Q. Dou, Q. Li, H. Chen, H. J. Lin, P. A. Heng, C. Haß, E. Bruni, Q. Wong, U. Halici, M. Ü. Öner, R. Cetin-Atalay, M. Berseth, V. Khvatkov, A. Vylegzhanin, O. Kraus, M. Shaban, N. Rajpoot, R. Awan, K. Sirinukunwattana, T. Qaiser, Y. W. Tsang, D. Tellez, J. Annuscheit, P. Hufnagl, M. Valkonen, K. Kartasalo, L. Latonen, P. Ruusuvuori, K. Liimatainen, S. Albarqouni, B. Mungal, A. George, S. Demirci, N. Navab, S. Watanabe, S. Seno, Y. Takenaka, H. Matsuda, H. A. Phoulady, V. Kovalev, A. Kalinovsky, V. Liauchuk, G. Bueno, M. M. Fernandez-Carrobles, I. Serrano, O. Deniz, D. Racoceanu, and R. Venâncio, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *JAMA - Journal of the American Medical Association*, vol. 318, no. 22, pp. 2199–2210, 2017.

[11] P. Bándi, O. Geessink, Q. Manson, M. Van Dijk, M. Balkenhol, M. Hermsen, B. Ehteshami Bejnordi, B. Lee, K. Paeng, A. Zhong, Q. Li, F. G. Zanjani, S. Zinger, K. Fukuta, D. Komura, V. Ovtcharov, S. Cheng, S. Zeng, J. Thagaard, A. B. Dahl, H. Lin, H. Chen, L. Jacobsson, M. Hedlund, M. Çetin, E. Halici, H. Jackson, R. Chen, F. Both, J. Franke, H. Kusters-Vandevelde, W. Vreuls, P. Bult, B. Van Ginneken, J. Van Der Laak, and G. Litjens, "From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient Level: The CAMELYON17 Challenge," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 550–560, 2019.

[12] J. H. Medicine, "Breast Ultrasound - Johns Hopkins Medicine." [Online]. Available: https://www.hopkinsmedicine.org/

health/treatment-tests-and-therapies/breast-ultrasound (Accessed 2020-06-21).

[13] K. I. Bland, E. M. Copeland, V. S. Klimberg, and W. J. Gradishar, *The Breast: Comprehensive Management of Benign and Malignant Diseases*. Elsevier Inc., August 2017.

[14] M. S. K. C. Center, "Anatomy of the Breast - Memorial Sloan Kettering Cancer Center." [Online]. Available: https://www.mskcc.org/cancer-care/types/breast/anatomy-breast (Accessed 2020-06-22).

[15] A. Aydiner, A. Igci, and A. Soran, Eds., *Breast Disease: Diagnosis and Pathology*, volume 1 ed. Cham: Springer International Publishing, 2019. [Online]. Available: http://link.springer.com/10.1007/978-3-030-04606-4

[16] "Breast Anatomy - National Breast Cancer Foundation." [Online]. Available: https://www.nationalbreastcancer.org/breast-anatomy (Accessed 2020-06-22).

[17] E. N. Marieb and K. Hoehn, *Human Anatomy & Physiology, Global Edition*, 2015.

[18] M. Al-Azab, "Anatomy of the Immune & Lymphatic System," 2017. [Online]. Available: https://www.researchgate.net/publication/317645471_Anatomy_of_the_Immune_Lymphatic_System

[19] T. E. o. E. Britannica, "Lymphatic system," 2020. [Online]. Available: https://www.britannica.com/science/lymphatic-system/Bone-marrow

[20] P. Susan Standring DSc, *Gray's anatomy 41st edition: The anatomical basis of clinical practice*, 2015.

[21] "Lymph Node Anatomy In Detail." [Online]. Available: https://www.anatomynote.com/human-anatomy/cell-and-tissue/lymph-node-anatomy-in-detail/ (Accessed 2020-06-21).

[22] 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," *GigaScience*, vol. 7, no. 6, pp. 1–8, 2018.

[23] Y. Sucaet and W. Waelput, "Digital pathology," in *SpringerBriefs in Computer Science*. Springer, Cham, 2014.

[24] M. D. Zarella, D. Bowman, F. Aeffner, N. Farahani, A. Xthona, S. F. Absar, A. Parwani, M. Bui, and D. J. Hartman, "A practical guide to whole slide imaging a white paper from the digital pathology association," *Archives of Pathology and Laboratory Medicine*, vol. 143, no. 2, pp. 222–234, 2019.

[25] V. D.C. Shields and T. Heinbockel, “Introductory Chapter: Histological Microtechniques,” in *Histology*. IntechOpen, January 2019.

[26] J. K. Chan, “The wonderful colors of the hematoxylin-eosin stain in diagnostic surgical pathology,” pp. 12–32, February 2014. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1066896913517939

[27] UICC, “TNM Classification of Malignant Tumours 2017,” *Oncoline*, 2017.

[28] S. Otálora, M. Atzori, V. Andrearczyk, A. Khan, and H. Müller, “Staining invariant features for improving generalization of deep convolutional neural networks in computational pathology,” *Frontiers in Bioengineering and Biotechnology*, vol. 7, no. AUG, 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6716536/

[29] A. Balkan, “OpenSlide,” 2013. [Online]. Available: https://openslide.org/ (Accessed 2020-07-19).

[30] G. Litjens, “ASAP - Automated Slide Analysis Platform,” 2018. [Online]. Available: https://computationalpathologygroup.github.io/ASAP/ (Accessed 2020-07-18).

[31] B. S. Veeling, “GitHub - basveeling/pcam: The PatchCamelyon (PCam) deep learning classification benchmark.” 2018. [Online]. Available: https://github.com/basveeling/pcam (Accessed 2020-06-20).

[32] M. Babaie, S. Kalra, A. Sriram, C. Mitcheltree, S. Zhu, A. Khatami, S. Rahnamayan, and H. R. Tizhoosh, “Classification and Retrieval of Digital Pathology Scans: A New Dataset,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, pp. 760–768, May 2017. [Online]. Available: http://arxiv.org/abs/1705.07522

[33] A. Das, M. S. Nair, and S. D. Peter, “Computer-Aided Histopathological Image Analysis Techniques for Automated Nuclear Atypia Scoring of Breast Cancer: a Review,” pp. 1–31, January 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10278-019-00295-z

[34] M. Veta, J. P. Pluim, P. J. Van Diest, and M. A. Viergever, “Breast cancer histopathology image analysis: A review,” pp. 1400–1411, 2014. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/24759275/

[35] 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,” pp. 60–88, December 2017.

[36] S. C. B. Lo, S. L. A. Lou, M. V. Chien, and S. K. Mun, “Artificial Convolution Neural Network Techniques and Applications for Lung Nodule Detection,” *IEEE Transactions on Medical Imaging*, vol. 14, no. 4, pp. 711–718, 1995.

[37] K. Sirinukunwattana, S. E. Raza, Y. W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, “Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1196–1206, May 2016.

[38] J. Li, S. Yang, X. Huang, Q. Da, X. Yang, Z. Hu, Q. Duan, C. Wang, and H. Li, “Signet Ring Cell Detection With a Semi-supervised Learning Framework,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11492 LNCS, pp. 842–854, July 2019. [Online]. Available: http://arxiv.org/abs/1907.03954

[39] G. S. Omenn, “Grand challenges and great opportunities in science, technology, and public policy,” in *Science*, 2006.

[40] L. Maier-Hein, M. Eisenmann, A. Reinke, S. Onogur, M. Stankovic, P. Scholz, T. Arbel, H. Bogunovic, A. P. Bradley, A. Carass, C. Feldmann, A. F. Frangi, P. M. Full, B. van Ginneken, A. Hanbury, K. Honauer, M. Kozubek, B. A. Landman, K. März, O. Maier, K. Maier-Hein, B. H. Menze, H. Müller, P. F. Neher, W. Niessen, N. Rajpoot, G. C. Sharp, K. Sirinukunwattana, S. Speidel, C. Stock, D. Stoyanov, A. A. Taha, F. van der Sommen, C. W. Wang, M. A. Weber, G. Zheng, P. Jannin, and A. Kopp-Schneider, “Why rankings of biomedical image analysis competitions should be interpreted with care,” *Nature Communications*, vol. 9, no. 1, pp. 1–13, December 2018. [Online]. Available: www.nature.com/naturecommunications

[41] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, August 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem. IEEE Computer Society, December 2016, pp. 770–778. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June. IEEE Computer Society, October 2015, pp. 1–9. [Online]. Available: https://arxiv.org/abs/1409.4842v1

[44] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on*

Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.

[45] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep Learning for Identifying Metastatic Breast Cancer," pp. 1–6, 2016. [Online]. Available: http://arxiv.org/abs/1606.05718

[46] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," May 2016. [Online]. Available: http://arxiv.org/abs/1605.06211

[47] N. Otsu, "THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS." *IEEE Trans Syst Man Cybern*, vol. SMC-9, no. 1, pp. 62–66, 1979.

[48] B. E. Bejnordi, G. Litjens, N. Timofeeva, I. Otte-Höller, A. Homeyer, N. Karssemeijer, and J. A. Van Der Laak, "Stain specific standardization of whole-slide histopathological images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 2, pp. 404–415, February 2016.

[49] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, April 2017. [Online]. Available: http://liangchiehchen.com/projects/

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351. Springer Verlag, May 2015, pp. 234–241. [Online]. Available: http://lmb.informatik.uni-freiburg.de/

[51] M. Ester, M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," pp. 226—-231, 1996. [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.9220

[52] N. Burlutskiy, N. Pinchaud, F. Gu, D. Hägg, M. Andersson, L. Björk, K. Eurén, C. Svensson, L. K. Wilén, and M. Hedlund, "Segmenting Potentially Cancerous Areas in Prostate Biopsies using Semi-Automatically Annotated Data," pp. 1–13, 2019. [Online]. Available: http://arxiv.org/abs/1904.06969

[53] B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, 2004.

[54] "Residual Networks (ResNet) — Dive into Deep Learning 0.14.2 documentation." [Online]. Available: https://d2l.ai/chapter_convolutional-modern/resnet.html (Accessed 2020-08-09).

[55] “Stanford University CS231n: Convolutional Neural Networks for Visual Recognition.” [Online]. Available: http://cs231n.stanford.edu/ (Accessed 2020-08-10).

[56] A. C. Ian Goodfellow, Yoshua Bengio, “Deep Learning Book,” *Deep Learning*, 2015.

[57] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems*, 2014.

[58] L. N. Smith, “Cyclical Learning Rates for Training Neural Networks,” *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, pp. 464–472, June 2015. [Online]. Available: http://arxiv.org/abs/1506.01186

[59] D. Denisko and M. M. Hoffman, “Classification and interaction in random forests,” pp. 1690–1692, February 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5828645/

[60] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-Augu. Association for Computing Machinery, August 2016, pp. 785–794. [Online]. Available: https://github.com/dmlc/xgboost

[61] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, “Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10553 LNCS, pp. 240–248, July 2017. [Online]. Available: http://arxiv.org/abs/1707.03237http://dx.doi.org/10.1007/978-3-319-67558-9_28

[62] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, “Dice Loss for Data-imbalanced NLP Tasks,” November 2019. [Online]. Available: http://arxiv.org/abs/1911.02855

[63] Karl Rosaen, “K-fold cross-validation, hyperparameter tuning and improving my score on Kaggle’s Forest Cover Type Competition,” 2016. [Online]. Available: http://karlrosaen.com/ml/learning-log/2016-06-20/ (Accessed 2020-08-10).

[64] S. Raschka, “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning,” November 2018. [Online]. Available: http://arxiv.org/abs/1811.12808

[65] T. Yu and H. Zhu, “Hyper-Parameter Optimization: A Review of Algorithms and Applications,” March 2020. [Online]. Available: http://arxiv.org/abs/2003.05689

[66] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, “Methods for nuclei detection, segmentation, and classification in digital histopathology: A review-current status and future potential,” *IEEE Reviews in Biomedical Engineering*, vol. 7, pp. 97–114, 2014.

[67] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” March 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[68] “S cikit-image 0.17.2 docs — skimage v0.17.2 docs.” [Online]. Available: https://scikit-image.org/docs/stable/ (Accessed 2020-08-07).

[69] “JamesLuoau/LossAccPlotter: Plot loss and accuracy of neural networks over time for Python 3.” [Online]. Available: https://github.com/JamesLuoau/LossAccPlotter (Accessed 2020-08-12).

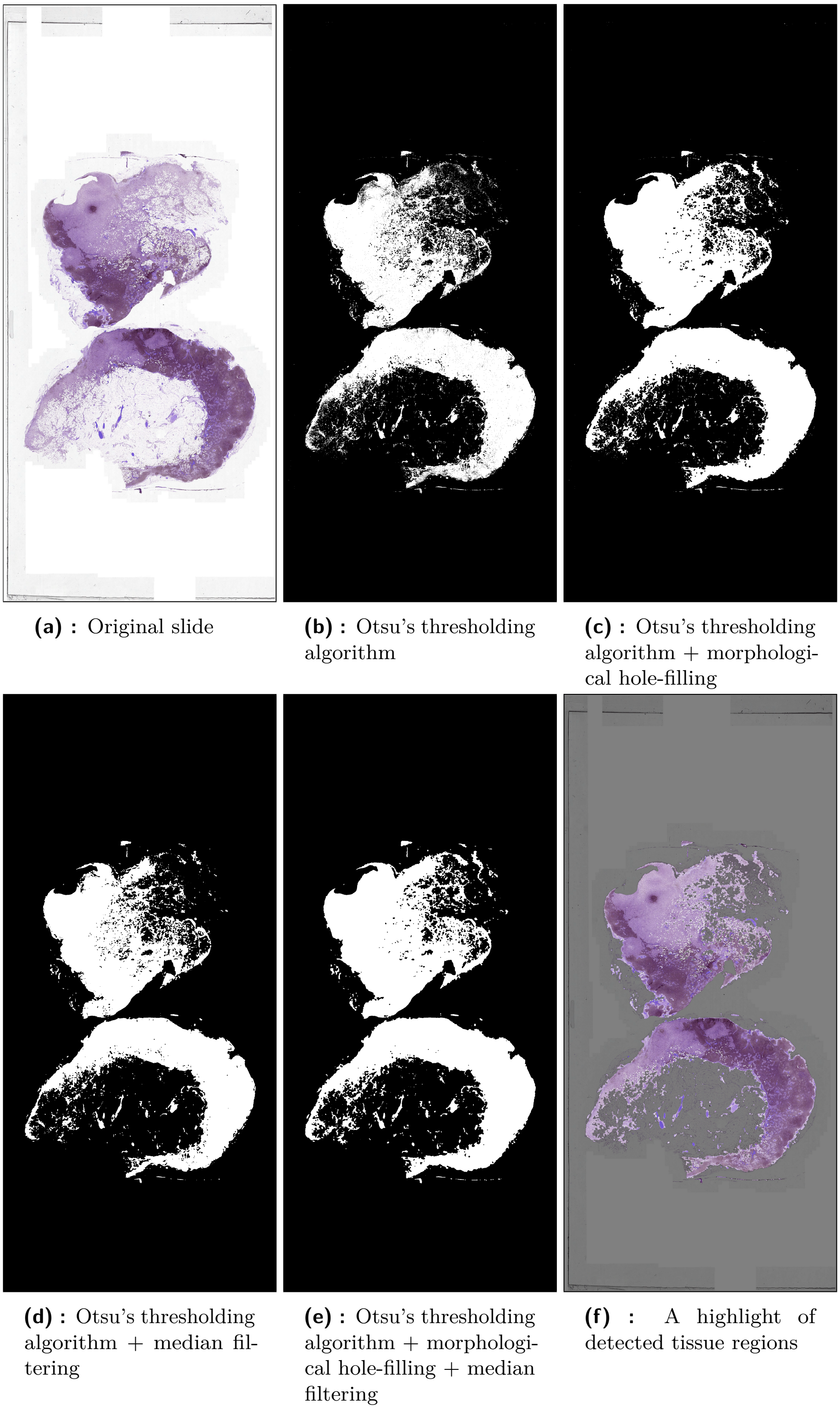
[70] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.

[71] K. Hajian-Tilaki, “Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation,” pp. 627–635, 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3755824/

[72] A. I. Bandos, H. E. Rockette, T. Song, and D. Gur, “Area under the free-response ROC curve (FROC) and a related summary index,” *Biometrics*, vol. 65, no. 1, pp. 247–256, March 2009. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2776072/

Appendix A

Tissue region detection visualization



(a) : Original slide

(b) : Otsu's thresholding algorithm

(c) : Otsu's thresholding algorithm + morphological hole-filling

(d) : Otsu's thresholding algorithm + median filtering

(e) : Otsu's thresholding algorithm + morphological hole-filling + median filtering

(f) : A highlight of detected tissue regions

Figure A.1: Visualization of the tissue detection algorithm. Each subplot represents a certain phase of the tissue detection process from the original image to the final highlight of the detected tissue.

Appendix B

Learning rate searching process logs

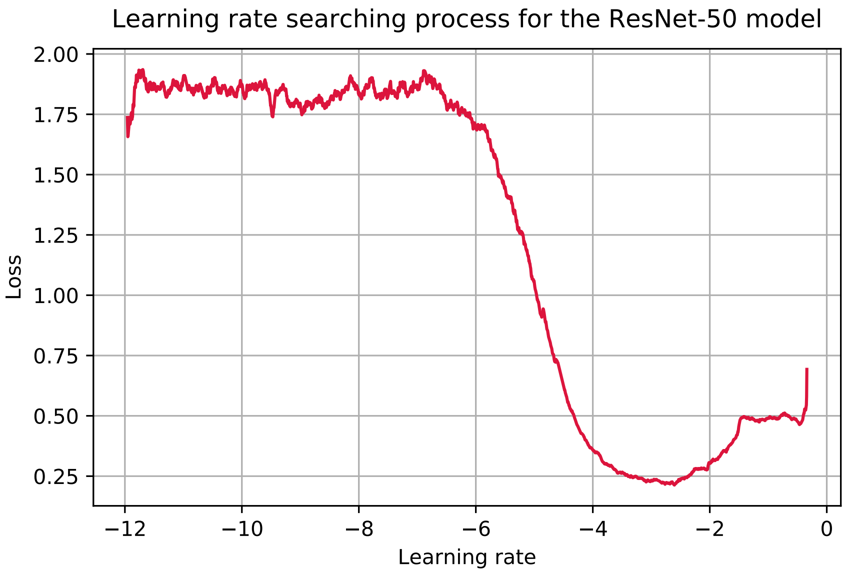


Figure B.1: Curve of the learning rate searching process for the ResNet-50 model.

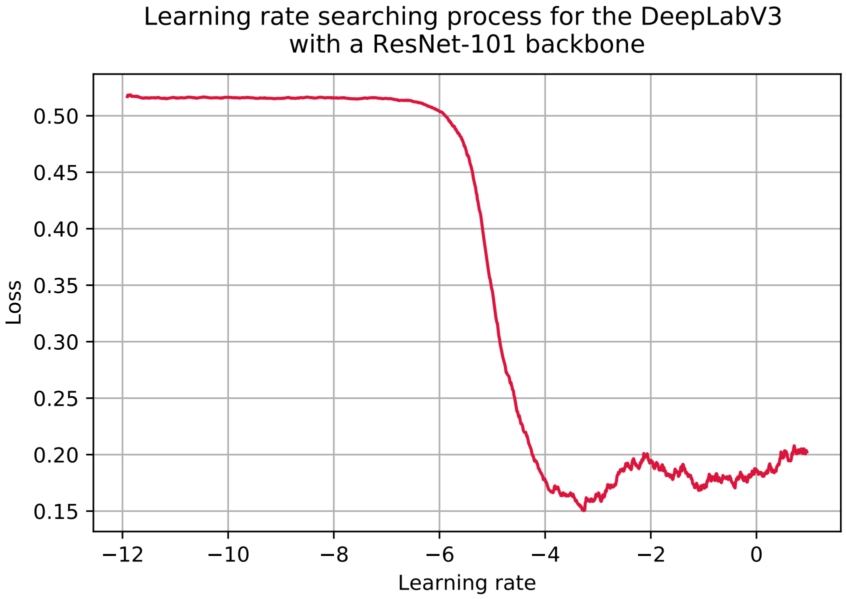


Figure B.2: Curve of the learning rate searching process for the DeepLabV3 model with a ResNet-101 backbone.

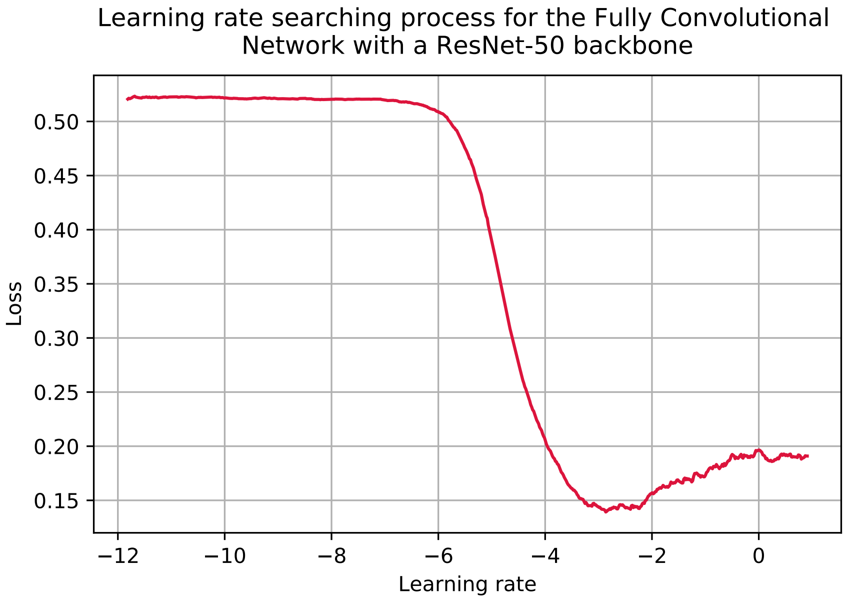


Figure B.3: Curve of the learning rate searching process for the FCN model with a ResNet-50 backbone.

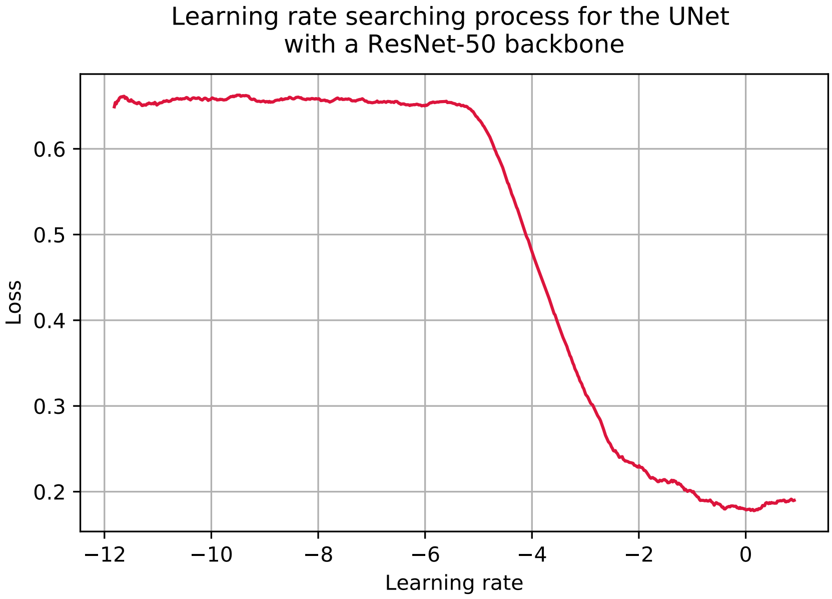
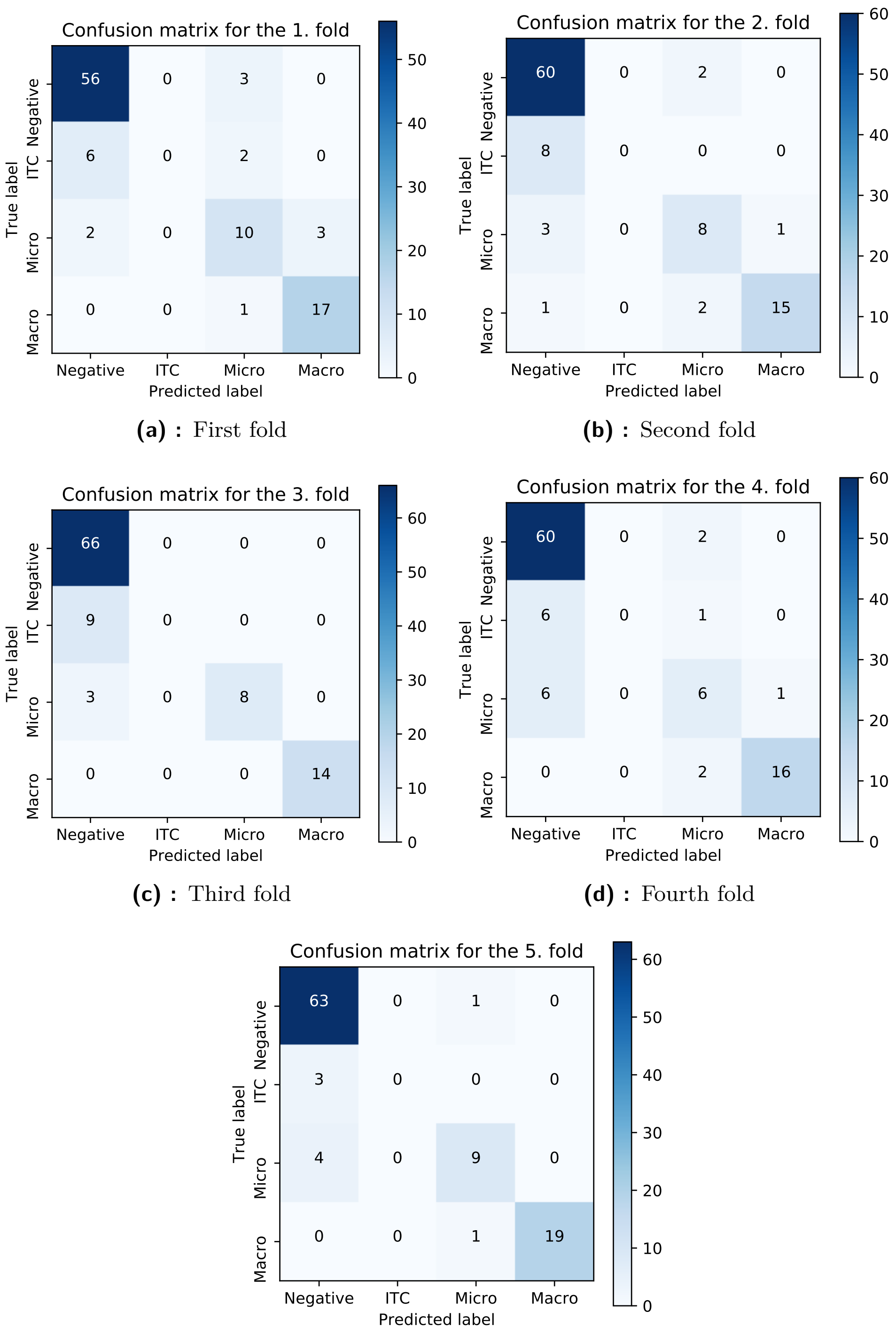


Figure B.4: Curve of the learning rate searching process for the UNet model with a ResNet-50 backbone.

Appendix C

5-fold cross-validation confusion matrices



(a) : First fold

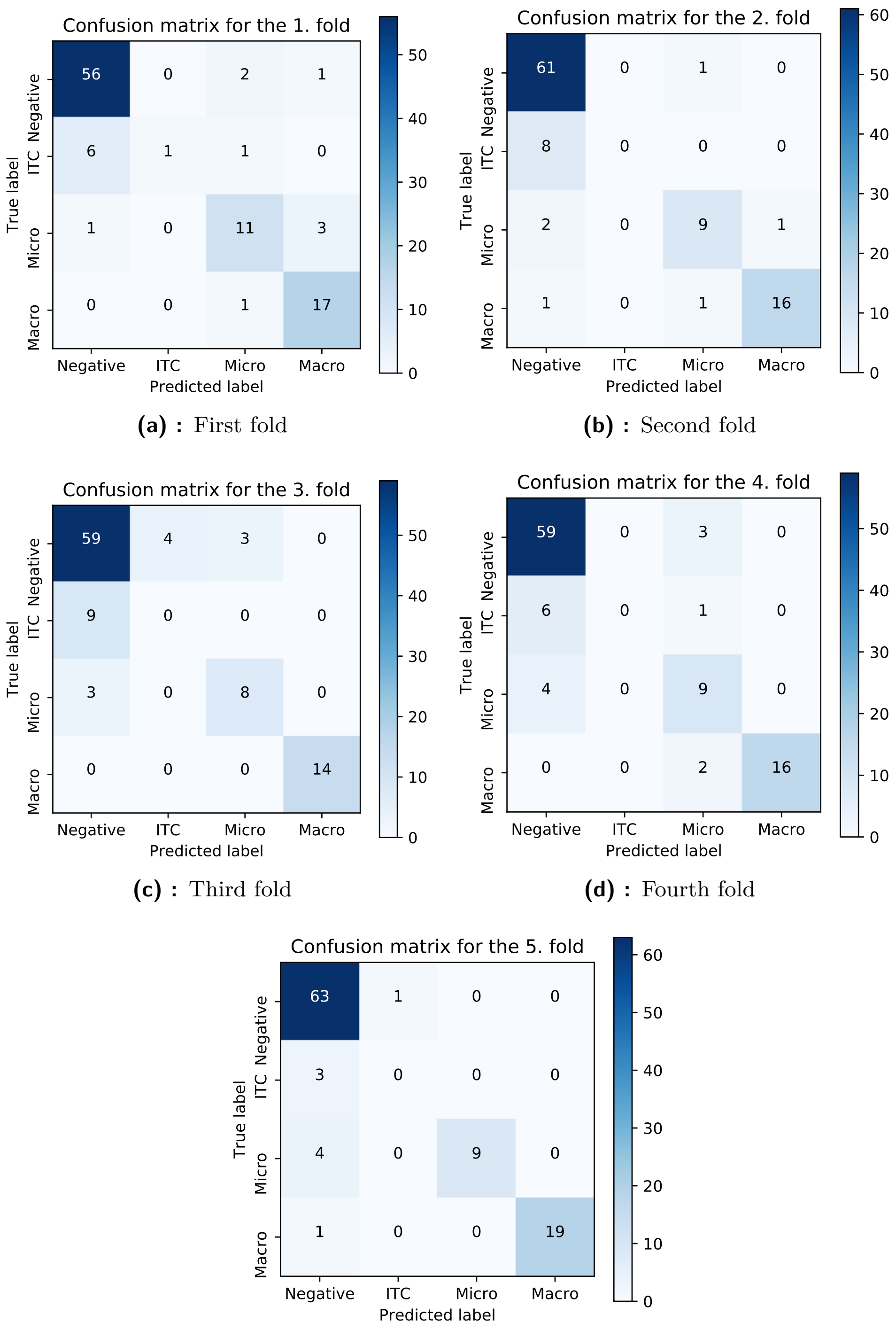
(b) : Second fold

(c) : Third fold

(d) : Fourth fold

(e) : Fifth fold

Figure C.1: Confusion matrices for the 5-fold cross-validation process of the Random forest classifier.



(a) : First fold

(b) : Second fold

(c) : Third fold

(d) : Fourth fold

(e) : Fifth fold

Figure C.2: Confusion matrices for the 5-fold cross-validation process of the XGBoost classifier.

Appendix D

Visualization of the DeepLabV3 performance

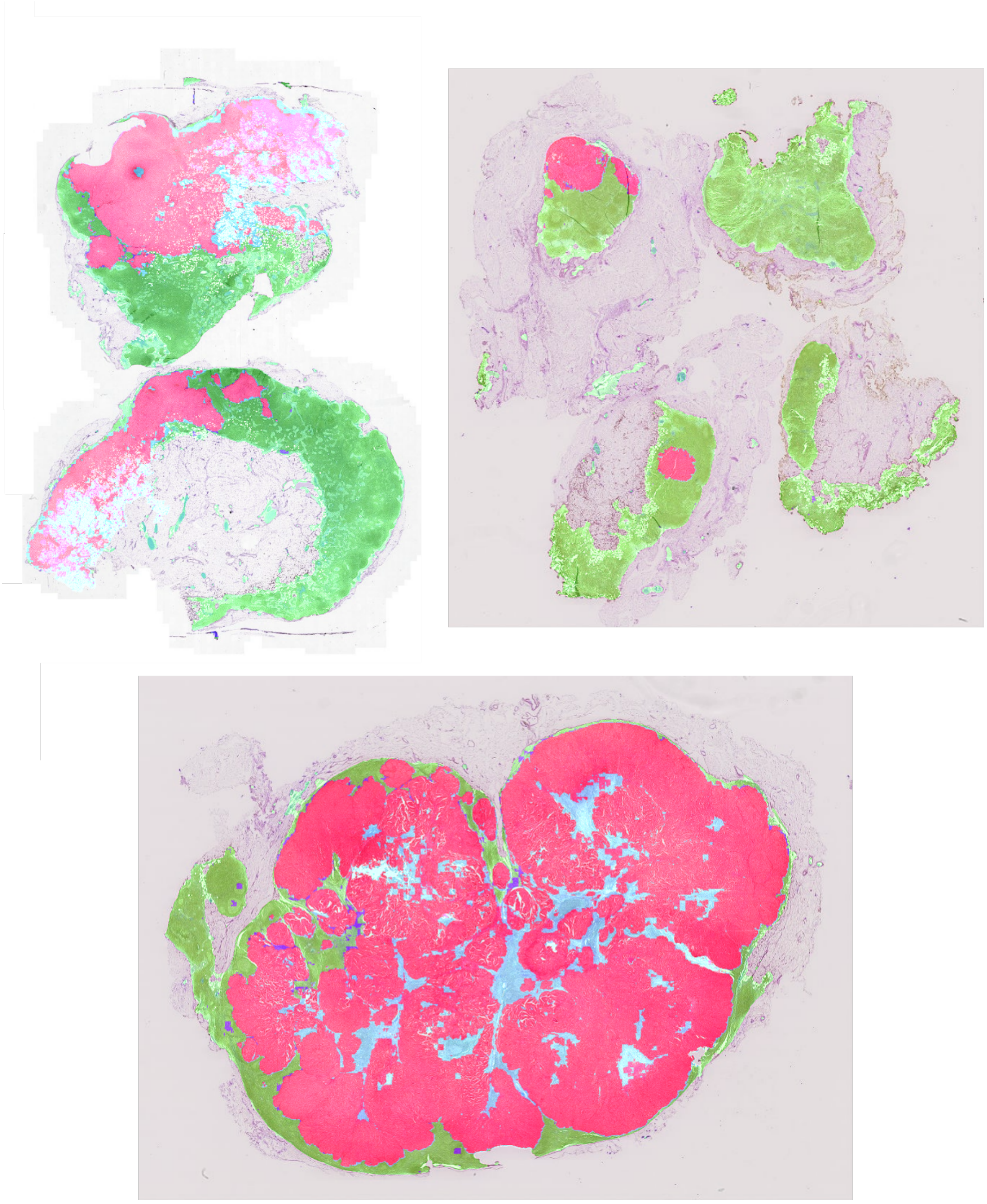
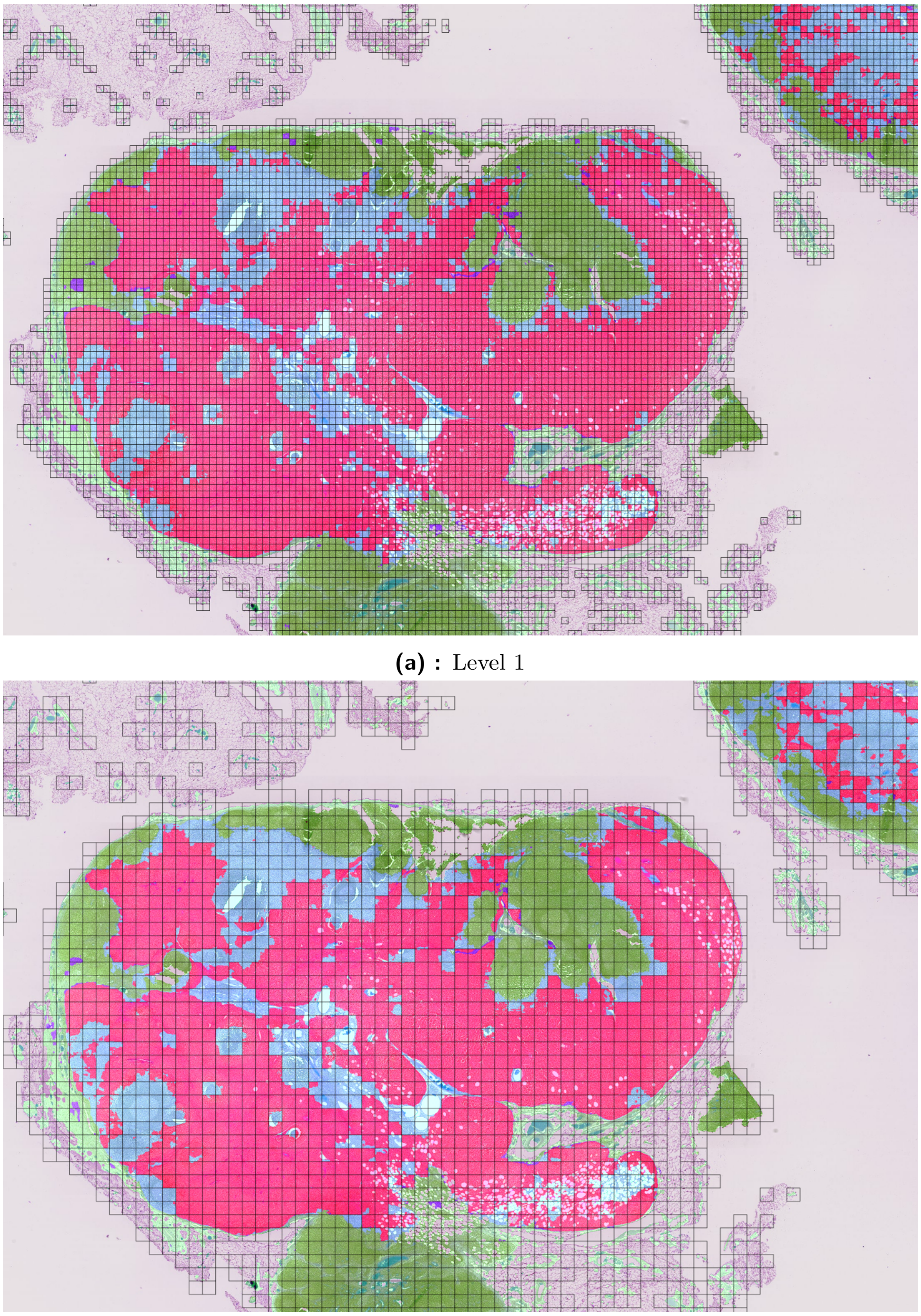


Figure D.1: DeepLabV3 performance visualisation generated for example slides. Dark blue stands for FP, red for TP, light blue for FN and green for TN.



(a) : Level 1

(b) : Level 2

Figure D.2: Comparison of the final evaluation for the predictions obtained from DeepLabV3 model at both resolutions. Dark blue stands for FP, red for TP, light blue for FN and green for TN. Grid, indicating from where the patches were sampled, is present.

Appendix E

Contents of the attachment

```
src
├─ 1_baseline_solution
│  ├─ evaluate_kaggle.py
│  └─ train_network.py
├─ 2_preprocessing_and_visualization
│  ├─ create_dataset.py
│  ├─ make_masks.py
│  ├─ make_patches.py
│  ├─ save_masks.py
│  └─ save_patches.py
├─ 3_patch_level_segmentation
│  └─ train_network.py
├─ 4_slide_level_classification
│  ├─ evaluate_c16.py
│  ├─ generate_features.py
│  ├─ generate_maps.py
│  ├─ official_evaluation.py
│  └─ train_classifier.py
├─ 5_patient_level_classification
│  ├─ evaluate_c17.py
│  └─ official_evaluation.py
├─ configuration.py
├─ laplotter.py
├─ plot_utils.py
├─ train_utils.py
└─ utils.py
```