

# Preparation of Whole Slide Images for usage in Neural Networks

Master Thesis

Submitted in partial fulfillment of the requirements for the degree  
of

**Master of Science (M.Sc.)**  
in Applied Computer Science

at the

Berlin University of Applied Sciences (HTW)



First Supervisor: Prof. Dr. Peter Hufnagl

Second Supervisor: Diplom Informatiker Benjaming Voigt

Submitted by:

**Sascha Nawrot (B.Sc.)**

Berlin, April 11, 2016

## **Abstract**

This is the abstract.

# Preface

Hello, this is the preface

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Motivation . . . . .	4
1.2	Research Objective . . . . .	4
1.3	About this thesis . . . . .	4
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	Definition of terms . . . . .	6
2.1.1	Deep Zoom Image Format . . . . .	6
2.1.2	Microservice . . . . .	7
2.1.3	Machine Learning . . . . .	7
2.1.4	Neural Networks . . . . .	7
2.2	Process chain . . . . .	7
2.2.1	Description . . . . .	7
2.2.2	Definition of Conversion Service . . . . .	7
2.2.3	Definition of Annotation Service . . . . .	7
2.2.4	Definition of Tessellation Service . . . . .	7
<b>3</b>	<b>Methodology</b>	<b>8</b>
3.1	Conversion Service . . . . .	8
3.1.1	Literature review . . . . .	8
3.1.2	Chosen methods . . . . .	8
3.2	Annotation Service . . . . .	8
3.2.1	Literature review . . . . .	8
3.2.2	Chosen methods . . . . .	8
3.3	Tessellation Service . . . . .	8
3.3.1	Literature review . . . . .	8
3.3.2	Chosen methods . . . . .	8
<b>4</b>	<b>Implementation</b>	<b>9</b>
4.1	Implementation of Conversion Service . . . . .	9
4.1.1	Used technologies and frameworks . . . . .	9
4.1.2	Documentation . . . . .	9
4.2	Implementation of Annotation Service . . . . .	9
4.2.1	Used technologies and frameworks . . . . .	9

4.2.2	Documentation . . . . .	9
4.3	Implementation of Tessellation Service . . . . .	9
4.3.1	Used technologies and frameworks . . . . .	9
4.3.2	Documentation . . . . .	9
<b>5</b>	<b>Discussion</b>	<b>10</b>
5.1	Conversion Service Test . . . . .	10
5.1.1	Setup . . . . .	10
5.1.2	Results . . . . .	10
5.2	Annotation Service Test . . . . .	10
5.2.1	Setup . . . . .	10
5.2.2	Results . . . . .	10
5.3	Tessellation Service Test . . . . .	10
5.3.1	Setup . . . . .	10
5.3.2	Results . . . . .	10
<b>6</b>	<b>Conclusion</b>	<b>11</b>
6.1	Results . . . . .	11
6.2	Conclusion . . . . .	11
6.3	Future tasks . . . . .	11
	<b>Bibliography</b>	<b>12</b>
	<b>List of Figures</b>	<b>13</b>
	<b>List of Tables</b>	<b>14</b>

# Chapter 1

## Introduction

### 1.1 Motivation

### 1.2 Research Objective

The objective of this thesis is the conceptualization and implementation of tools for whole slide images<sup>1</sup>, which allow for their annotation and a further usage in neural networks<sup>2</sup>. As a requirement, the tools have to be implemented in the form of microservices<sup>3</sup>, each one with their own short documentation, including instructions for installation, usage and some exemplary use cases. To achieve this goal, the implementation of 3 microservices is necessary.

The first microservice needs to be capable of converting a given set of image formats into the so called *Deep Zoom Image Format*<sup>4</sup>. The supported image formats are *.bif*, *.mrxs*, *.ndpi*, *.scn*, *.svs*, *.svslide*, *.tif*, *.tiff*, *.vms* and *.vmu*, in accordance with the capabilities of the Openslide framework [1].

### 1.3 About this thesis

Apart from the *Introduction*, there are 5 more chapters in this thesis.

*Chapter 2 - Background* defines some terminology and the general, required process chain which are all necessary to understand further chapters of this thesis. Furthermore, 3 microservices will be introduced in short.

*Chapter 3 - Methodology* gives an overview over the current state of research for each microservice, as well as best practices.

*Chapter 4 - Implementation* goes into further details about how each microservice is implemented and which software and frameworks were used for that.

---

<sup>1</sup>See chapter 2.1.x

<sup>2</sup>See chapter 2.1.x

<sup>3</sup>See chapter 2.1.x

<sup>4</sup>See chapter 2.1.x

*Chapter 5 - Discussion* will introduce a measurement for each microservice to measure its success. It will discuss the test setup as well as list the results.

*Chapter 6 - Conclusion* will interpret the Results from Chapter 5 and analyze them closer. Furthermore, it will give an idea of what steps are to be taken next in the future.

## Chapter 2

# Background

### 2.1 Definition of terms

#### 2.1.1 Deep Zoom Image Format

The Deep Zoom Image Format (.dzi) is an xml-based file format maintained by Microsoft to improve performance and quality in the handling of large image files. Therefore an image will be represented in a tiled pyramid scheme (see fig. 2.1).

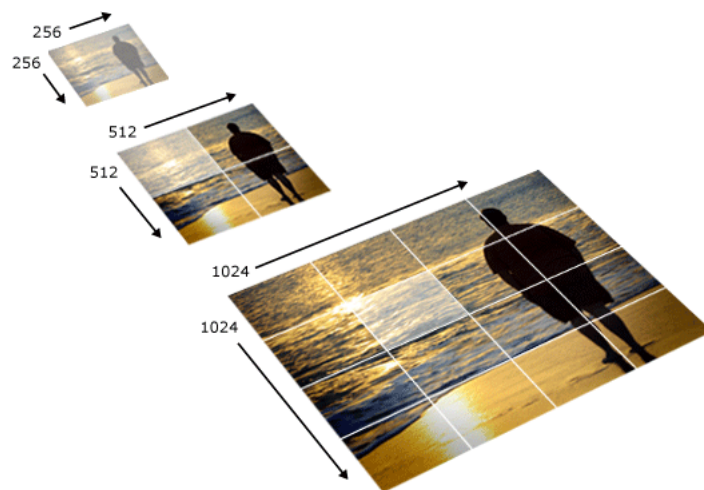


Figure 2.1: of the dzi pyramid image representation (source: <https://i-msdn.sec.s-msft.com/dynimg/IC141135.png>)

As seen in fig. 2.1 there are multiple versions of a single image in different resolutions. The idea behind this is, that if a user wants to see a whole picture



zoomed out or as a small thumbnail, it is not necessary to load the image file in its highest resolution. To save bandwidth a version with a smaller resolution is loaded. If the user wishes to zoom in on a specific area of the image, a version with a higher resolution is loaded. Once again, however, it is not necessary to load the whole image, since only a fraction of it will be visible. For this reason there are tiles of the image which are loaded instead (see highlighted tiles in fig. 2.1) [2].

Each resolution in the pyramid is called a *level*. At each level the image is scaled down by the factor 4 (2 in each dimension). In other words, a level can be defined as an image with the resolution  $2^{\text{level}}$  for height and width, resulting in a resolution of  $(2^{\text{level}})^2$ . Levels are counted from level 0 (1\*1 Pixel) [2]. E.g. the levels shown in fig. 2.1 are:

- level 8 ( $2^8 = 256$ ) for the  $256^2$  pixel image
- level 9 ( $2^9 = 512$ ) for the  $512^2$  pixel image
- level 10 ( $2^{10} = 1024$ ) for the  $1024^2$  pixel image

### **2.1.2    Microservice**

### **2.1.3    Machine Learning**

### **2.1.4    Neural Networks**

## **2.2    Process chain**

### **2.2.1    Description**

### **2.2.2    Definition of Conversion Service**

### **2.2.3    Definition of Annotation Service**

### **2.2.4    Definition of Tessellation Service**

## Chapter 3

# Methodology

### 3.1 Conversion Service

#### 3.1.1 Literature review

#### 3.1.2 Chosen methods

### 3.2 Annotation Service

#### 3.2.1 Literature review

#### 3.2.2 Chosen methods

### 3.3 Tessellation Service

#### 3.3.1 Literature review

#### 3.3.2 Chosen methods

## Chapter 4

# Implementation

### 4.1 Implementation of Conversion Service

#### 4.1.1 Used technologies and frameworks

#### 4.1.2 Documentation

### 4.2 Implementation of Annotation Service

#### 4.2.1 Used technologies and frameworks

#### 4.2.2 Documentation

### 4.3 Implementation of Tessellation Service

#### 4.3.1 Used technologies and frameworks

#### 4.3.2 Documentation

## Chapter 5

# Discussion

### 5.1 Conversion Service Test

#### 5.1.1 Setup

#### 5.1.2 Results

### 5.2 Annotation Service Test

#### 5.2.1 Setup

#### 5.2.2 Results

### 5.3 Tessellation Service Test

#### 5.3.1 Setup

#### 5.3.2 Results

## Chapter 6

# Conclusion

6.1 Results

6.2 Conclusion

6.3 Future tasks

# Bibliography

- [1] A. Goode et al. Openslide: A vendor-neutral software foundation for digital pathology. *J pathol inform*, 4(1), September 2013. [http://download.openslide.org/docs/JPatholInform\\_2013\\_4\\_1\\_27\\_119005.pdf](http://download.openslide.org/docs/JPatholInform_2013_4_1_27_119005.pdf).
- [2] Microsoft. Deep zoom file format overview. [https://msdn.microsoft.com/en-us/library/cc645077\(v=vs.95\).aspx](https://msdn.microsoft.com/en-us/library/cc645077(v=vs.95).aspx). Accessed: 11.04.2016.

# List of Figures

2.1	of the dzi pyramid image representation (source: <a href="https://i-msdn.sec.s-msft.com/dynimg/IC141135.png">https://i-msdn.sec.s-msft.com/dynimg/IC141135.png</a> ) . . . . .	6
-----	--	---

## List of Tables