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

Chapter 1

Introduction

This is an introduction [1].

1.1 Motivation

1.2 Objective

1.3 Research Question

1.4 About this thesis

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[1] 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).

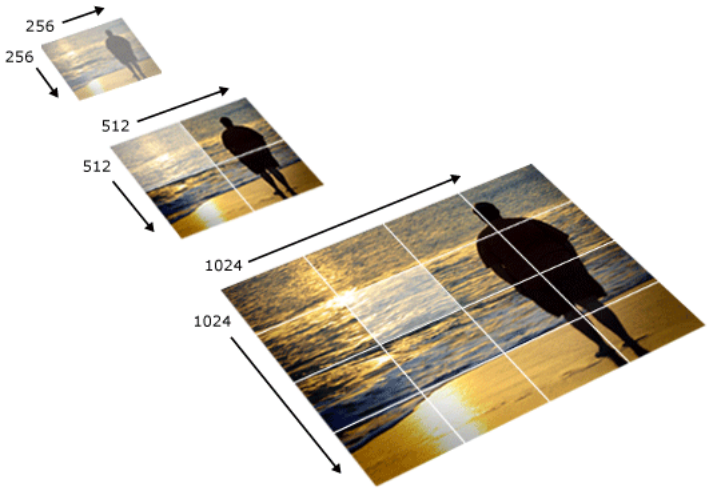

Figure 2.1: example of the dzi pyramid image representation (source: https://i-msdn.sec.s-msft.com/dynimg/IC141135.png)

[1] See https://msdn.microsoft.com/en-us/library/cc645077(v=vs.95).aspx for further details.

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 highlited tiles in fig. 2.1).

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*level for height and width, resulting in a resolution of (2*level)*(2*level). Levels are counted from level 0 (1*1 Pixel). 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 Tesselation 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] Microsoft. test01. `www.google.com`. Accessed 27.01.1989.

List of Figures

List of Tables