Czech Technical University in Prague
Faculty of Civil Engineering
Department of Geomatics

Master's Thesis

Green Vegetation Classification in the Prague Region

Bc. Petr Poskočil

Supervisor: Prof. Ing. Lena Halounová, CSc.

Study Programme: Geodesy and Cartography

Field of Study: Geomatics

May 24, 2020

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta stavební
Thákurova 7, 166 29 Praha 6

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Bc. Poskočil Jméno: Petr Osobní číslo: 423782

Zadávající katedra: katedra geomatiky

Studijní program: Geodézie a kartografie

Studijní obor: geomatika

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce: Klasifikace zeleně na území Prahy

Název diplomové práce anglicky: Green Vegetation Classification in the Prague Region

Pokyny pro vypracování:
Proveďte klasifikaci ploch zeleně na území města.
Využijte letecké snímky s pásmy RGB a blízké inftračervené pásmo.
Využijte pravděpodobnostní klasifikační metodu i metody založené na strojovém učení.
Vyhodnoťte přesnosti klasifikací. Popište limity zvolených metod výpočtu.

Seznam doporučené literatury:
Halounová L., Pavelka K.: Dálkový průzkum Země. Vydavatelství ČVUT, Praha 2007. ISBN: 80-01-03124-1
Halounová, L.: Zpracování obrazových dat. ČVUT v Praze, 2008. ISBN: 978-80-01-04253-3
Active Learning Methods for Remote Sensing Image Classification
https://ieeexplore.ieee.org/abstract/document/4812037
https://pythontips.com/2017/11/11/introduction-to-machine-learning-and-its-usage-in-remote-sensing/

Jméno vedoucího diplomové práce: prof. Ing. Lena Halounová, CSc.

Datum zadání diplomové práce: 15. 2. 2020 Termín odevzdání diplomové práce: 24.5.2020
Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku

Podpis vedoucího práce Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

Datum převzetí zadání Podpis studenta(ky)

Declaration

I hereby declare that I have completed this thesis independently, and that I have listed all the literature and publications used.

In Prague on May 24, 2020 .....

Aknowledgements

I would like to express my appreciation to my supervisor, Prof. Ing. Lena Halounová, CSc., who guided me through the thesis and shared her valuable knowledge. My appreciation also extends to Mgr. Jiří Čtyroký Ph.D. and Mgr. Ondřej Míček of the Prague Institute of Planning and Development who initiated the topic and provided me with data. Thanks also go to Ing. Ondřej Pešek, who helped me with the difficulties I encountered. In addition, I would like to thank Prof. Dr. Ing. Karel Pavelka, who provided me with the hardware needed for the practical part of the thesis.

Abstract

Urban greenery is extremely important for healthy urban environment. For this reason, the greenery must be monitored. This thesis attempts to contribute to the geographic data of the Prague municipality by proposing a method for single-crown-detection and single-crown-delineation. Such data would provide a basis for vegetation-related studies on a single-tree-level. Two methods were designed and implemented to provide a more reliable result. The first method is based on rather traditional remote sensing techniques using Geographic Information System. The other one uses deep learning techniques based on Mask R-CNN neural network framework. Both models are compared using designed accuracy assessment. Using the proposed Mask R-CNN-based method, tree crowns can be delineated with an overall accuracy of 81%. It also proved to be more efficient than the other "traditional" remote sensing technique used in this study.

Keywords: remote sensing; deep learning; Mask R-CNN; single-crown-delineation

Abstrakt

Městská zeleň je nesmírně důležitá pro zdravé městské prostředí, a proto je důležité ji monitorovat. Tato práce se snaží přispět ke zpřesnění geografických dat města Prahy návrhem metody detekce jednotlivých korun stromů. Takováto data by poskytla základ pro různé studie související s vegetací v měřítku jednoho stromu. Byly navrženy a implementovány dvě metody, aby byl poskytnut spolehlivější výsledek. První metoda je založena na tradičnějších technikách dálkového průzkumu Země využívajících geografický informační systém. Druhá využívá techniky hlubokého učení založené na neuronové síti Mask R-CNN. Oba modely jsou porovnány pomocí navrženého posouzení přesnosti. Metodou založenou na Mask R-CNN mohou být koruny stromů detekovány s celkovou přesností 81%. Ukázalo se také, že metoda Mask R-CNN je účinnější než tradičnější metoda založená na dálkovém průzkumu Země použitá v této studii.

Klíčová slova: dálkový průzkum Země; hluboké učení; Mask R-CNN; detekce koruny

Contents

List of Abbreviations

AI Artifcial Intelligence
ALS Airborne Laser Scanning
ANN Artifcial Neural Networks
CHM Canopy Height Model
CIR Color-infrared
CNN Convolutional Neural Networks
DAS Distance Allocation Segmentation
DL Deep Learning
DSM Digital Surface Model
DTM Digital Terrain Model
FAIR Facebook AI Research
FCN Fully Convolutional Network
FPN Feature Pyramid Network
LiDAR Light Detection and Ranging
LMF Local-maxima-fnding
Mask R-CNN Region-based Convolutional Neural Network
ML Machine Learning
nCHM negative Canopy Height Model
NDVI Normalised Difeerential Vegetation Index
NIR Near-infrared
OBIA Object-based Image Analysis
R-CNN Region-based Convolutional Neural Network
ReLU Rectified Linear Unit
ResNet Residual Neural Network
RGB Red Green Blue
RoI Region of Interest
RoIAlign Region of Interest Align
RPAS Remotely Piloted Aircraft Systems
RPN Region Proposal Network
SAR Synthetic-aperture Radar
SGD Sustainable Development Goals
TIFF Tag Image File Format
TO True Orthophoto
VI2 Vegetation Index no 2

Chapter 1

Introduction

Urban greenery is an indispensable part of a city and servers as a natural resource. Greenery reduces several environmental impacts on the city. For instance, it improves air quality, retains water, and reduces overheating. Apart from the physical effects on the environment, greenery has a positive psychological impact on people's health, aesthetic, and overall life quality perception. The health of the greenery is crucial to future sustainable development of the city in future; therefore, it must be monitored to provide reliable data for urban management. This research aims to contribute to the United Nations' 17 Sustainable Development Goals (SDG) towards sustainability [56]. More precisely, the thesis tackles SGD 11, 13, and 15; regarding sustainable cities and communities, the life of the land, and climate action.

In the context of urban management, the city is rather a complex organism with quite a lot of laws, rules, or restrictions. Several professional services are needed to keep this complex organism running. Each of these professions rely on each other while planning, managing, or maintaining the city. Thus, there is a need to be as precise as possible for mismanagement to be eliminated. So far, in the current data set for the municipality of Prague, the vegetation is considered in the Digital technical map of Prague [32] in the form of a polygonal representation. Each vector polygon represents a class with a certain type of vegetation. There are types such as gardens, meadows, greenery in developed areas, and others. The current state might be suitable for the delimitation of vegetation as a whole. However, it does not provide any information about the vegetation itself. For instance the number of trees in a particular area, their attributes, species, or the percentage of each species are still missing. In other words, the current spatial definition of the vegetation in Prague is pretty vague. There is indeed a desire for a more detailed vegetation data since the city needs to keep up with other metropolitan cities. This is all the more true because the municipality aims to fulfil a responsibility in resolving climate issues. How can one achieve this data?

Geospatial information science, especially remote sensing, is solving such tasks on a daily basis. A lot within this field has been accomplished already. So, it is evident that a solution is available. We might generalize a bit and say, that in remote sensing there are per-pixel-based approaches, and object-based approaches (a group of similar pixels, which are merged

into objects). These two approaches are more traditional in remote sensing and they have been used for decades. Even though they are still relevant, there are some other newer approaches. Some of the new methods are slowly taking over newly developed techniques which make some solutions more feasible. This involves artificial intelligence; more precisely deep learning, and even more precisely Convolutional Neural Networks (CNN). The research question is whether these "novel" methods are ready to become the right tool, or whether the "traditional" methods can keep up with the current technology. The thesis presents findings in a case study of the city of Prague. Both approaches are tested on performing the single-crown-detection and single-crown-delineation task on top of high-resolution aerial imagery with NIR band and the LiDAR DSM raster data.

1.1 Literature

There are many research topics focused on the vegetation with the use of remote sensing techniques. One of them is vegetation health assessment, detection, classification, or simply just capturing its current state [24]. With reference to remote sensing, the data used in such studies are remotely-sensed either from various carriers (aircrafts, satellites, RPAS etc.) by measuring emitted or reflected radiation of the Earth's surface. The data is sensed in many wave bands of spectrum, recorded, and stored in multiple image bands. Another application of remote sensing in vegetation studies is the airborne laser scanning (ALS). ALS is mostly called light detection and ranging (LiDAR). LiDAR is a method that measures the distance from the laser to the earth's surface by illuminating the object by radiation and measuring the reflected radiation using a sensor attached to an aircraft. Remotely-sensed data dramatically reduces the load of fieldwork, likewise the time needed for data processing. The result of LiDAR measurement is incomparably more accurate than the ground measurement. Therefore, it is an inexpensive alternative to field-based measurements [47]. In addition, there is a plenty of open-source data already available and easily accessible sometimes even via web map service.

According to extensive review of studies on tree classification in [24], which reviewed more than a hundred studies. Approximately 30 percent studies used imaging spectroscopy or hyperspectral imagery, $\sim$ 25 percent used high and very-high spatial resolution sensors, $\sim$ 20 percent combined passive optical sensors and active sensor (LiDAR), $\sim$ 15 percent were only LiDAR-based, and the rest used either thermal sensors or synthetic-aperture radar (SAR). Approximately half of the studies used the object-based classification and the other half used the pixel-based classification or compared the field spectra. The majority of the studies were conducted on a single tree scale.

There are two fundamental types for using ALS to classify trees. The first is a cluster-based approach usually providing broader scale. The other one is a single-tree approach, which provides information on the scale of the tree as a unit, thus a more detailed classification [58]. The essential issue for the vegetation-related studies at the single tree scale is the spatial definition of an instance. Most of the papers refer to terms such as single-crown-detection and single-crown-delineation. Single-crown-detection is a process of detection of a single

unit, usually represented by coordinates of a point regardless of the shape or size. The single-crown-delineation, on the other hand, aims to describe the single unit as detailed as possible. By referring to single-crown-delineation, it usually means the polygonal representation of shape, size, and perimeter.

1.1.1 Per-pixel-based and object-based classification methods

As far as the data, a vast majority of models used the LiDAR data, or a combination of LiDAR data and optical or SAR imagery. The model can benefit from the use of LiDAR data because the data is less affected by shading and atmospheric conditions [60]. It has also been proved that models based on LiDAR data tend to derive more correctly-detected trees and more precisely-delineated crowns [15]. Method-wise, the single-crown-detection and single-crown-delineation field of research are dominated by the local-maxima-finding (LMF)-based models. This is true in using ALS data at least. According to [22], six out of eight reviewed papers based their models on the LMF approach, mostly in combination with other techniques. More precisely, LMF in combination with filtering, region growing, multi-scale canopy height model (CHM), or watershed segmentation. Another reviewed approach, which works directly with raw ALS data is a combination of segmentation and clustering. The last technique reviewed in this paper was a polynomial fitting method in combination with the watershed segmentation. This method fits a polynomial of the second-degree to a morphological profile of a potential crown. The majority of the papers used LMF, mostly in combination with other techniques which are described further.

In [47], the authors focused mainly on both single-crown-detection and single-crown-delineation in coniferous forest based on the high-resolution imagery with red (R), green (G), blue (B) and near-infrared (NIR) bands. It is the sole representative of a model using only aerial imagery. The paper describes the process of creating automated detection and delineation algorithms. The detection algorithm is divided into several major phases. Particularly, the preprocessing and the imagery refinement, the local maximum moving window for potential treetop detection, the transect sampling extraction from potential treetop for the tree edge detection, scaling the length of transects to a single crown size, analyzing the drops among transects signalizing the edge of a tree, fitting circular boundary to the most significant drops among transects, and finally computing centroid position representing the treetop. The delineation algorithm shares the same algorithm design, apart from a slightly modified input image. It also returns transects drop positions instead of a distance. The crown delineation is then represented by an enclosing polygon.

A given example given from [47] might serve as a role model. Leastways, it can work to some extent, since most of the researched models based on LMF follow a similar pattern of phases. Generally, a template for an LMF-based model might look like the following set of steps; (1) Pre-processing and refinement where, for instance, high-pass filtering or smoothing is done; (2) LMF which involves kernel of given values and size, and thresholding the maxima values; (3) Delineation of tree crown, usually based on treetop locations. Methods range from already mentioned transect sampling, region growing, watershed segmentation to Thiessen polygons; (4) Refinement, which is usually done for both crown detection and delineation.

The refinement phase is very important since the models do not derive directly reliable results. Such an issue is usually being resolved iteratively or on multiple scales for various tree sizes. Therefore, the following text is more about the differences among models rather than the similarities.

Model used in [16], in comparison with [47] works with masked vegetation, which is further segmented into objects. Unlike the per-pixel model from [47], the model from [16] is object-based. The pixels are merged into segments, or so-called superpixels based on similar spectral values. Local height maxima of these segments are represented by points/pixels creating potential treetops, which is just another way of LMF. The treetops are used as seed points for region growing. These points expand to the crown boundaries, which are identified as a positive difference between the current and the next pixel. The algorithm determines a value of threshold and new crown delineation is created. So instead of transect sampling, the region growing method is used. The model works iteratively. Delineated crowns from the first iteration are classified into single crowns or crown cluster. Iteration then continues on the clusters only, until split into single crowns reaches thresholds.

Another similar model is described in [52]. It is also based on the region-growing algorithm using LMF. Unlike the method described in [16], the method from [52] retain only the uppermost pixels in a grid which later affects the process of LMF. The crown delineation was done the same way. In [15], the authors also proposed a model using the region-growing algorithm. Novelty in this approach was a combination of both CHM and ALS data. Even though CHM originated from ALS, they both can provide extra differentiating features. For instance, the crown delineation is derived from the ALS data after the threshold was applied to the points and these points were enclosed by 2D convex hull. The paper also describes its-own way of algorithm functions and the use of thresholds. However, the main principle remains the same in general.

In [35], the authors took a model based on LMF. In this case, a self-defined kernel window was used for LMF. Delineation was based on the marker-controlled watershed segmentation. To make the model more reliable, the model was extended to predict the size of a crown based on the height of a particular pixel. The moving window differs from pixel to pixel meaning the higher the pixel, the bigger the moving window. In [63], the author also works with the relation between height and crown diameter. A simple local maxima searching window of the chosen size was used in this model. The crown diameter was defined by a circle of diameter dependent on the height of a tree. The same approach was used in [46], however, the moving window was defined by an average size of crowns computed form field data. The same value was then used as a representation of a crown diameter. An interesting twist added on top of the LMF method was reviewed in [35]. They proposed a minimum curvature-based model. The CHM was scaled by the curvature layer derived from a slope, where the curvature smaller than zero represents gaps among trees and the positive curvature represents treetops. Then the LMF is applied on the curvature instead and the delineation is then calculated by the watershed segmentation. In [7], the authors followed a similar model design, the only difference is the crown delineation, which is processed by Thiessen polygons which are later simplified to create a more natural look. It is necessary to emphasize, that

this method is suitable only for continuous data.

Several papers tackled the issue of different tree shapes, sizes, and heights, claiming, that different-sized trees should be extracted over different scales. The term scale has many meanings in the literature. In this context, scale refers to the level of the spatial detail. When referring to the scale in vegetation-related studies, scaling usually implies low pass filtering on to achieve a different level of detail. Scaling helps to reduce the amount of detail which might cause over-segmentation of an image. Well balanced segmentation is therefore crucial to such models. Mostly because of their object-based nature. The model proposed in [34] incorporated the scale analysis as a first phase to determine dominant crown sizes. Later, multiple Gaussian filters are applied to fit all crown sizes form the smallest to the largest. Next multiple watershed segmentation maps were generated with further refinement. Finally, the combination was done by integrating all scales to create one crown delineation map. This was done by assuming that the tree crowns are more circular than the tree clusters. Another model proposed in [62], is fairly similar, but more complex. Their model extracts three geometric properties from the segmented CHM on multiple scales, namely the size, convexity and circularity of a tree crown. The main idea is to approximate the crown with an ellipsoid. Then, the best approximation of a crown is selected over all the scale and combined together to create a tree crown map. Another representative is the multi-scale Laplacian of Gaussian method published in [35] which used space-scale-based selection combined into one layer. Multi-level scaling can be applied not only on the raster-based data but also directly on the raw ALS data. Such an example can be found in [45]. The authors use multiple-scale Gaussian filters on a 3D CHM created from ALS data. They segmented the points and the best one that fitted to the parabolic surface was selected.

In [35] the authors found in their benchmark that a simple LMF-based model has turned out to be the overall best method. There were 14 models compared among various categories. Most of the models used the same principles previously described. Additionally, the LMF-based models also have the most straightforward implementation among several software that makes them feasible for commercial purposes. The research for this thesis was data-driven to some extent since there are ALS data already available as well as the CIR imagery with NIR band. Therefore, studies which managed to combine both data sets together were the most suitable. Most of the models in research had been tested on a continuous forest. The result of the high value helped to clarify various principles which were combined together. The main purpose was to derive the most reliable delineated crowns for further elaboration.

1.1.2 Deep-learning-based methods

Deep learning (DL), known also as deep structured learning, is a subset of broad machine learning family. Machine learning (ML) is a subfield of even broader artificial intelligence (AI) family. The term “learning machine” was first introduced in 1950s, proposing that machines could develop into artificial intelligence [55]. DL was introduced in 1986, more precisely, the definition and terminology was formed in [49]. The basic principles and concepts of deep learning had already been around for decades. In 2000, the artificial neural networks (ANN) were made [3]. Since then, the field was growing rapidly taking advantage of technological

development. It became more feasible, thus more applicable in many fields. In 2013, MIT [41] listed DL as one of the ten biggest technological breakthroughs. The application areas vary a lot, for instance: image recognition, computer vision, automatic speech recognition, medicine, financial sector, etc. Fields of applications are those that have something to do with artificial intelligence or big data. Geospatial information science belongs to one of them. The amount of data that is being processed in this field makes it a perfect candidate for such technology.

As the remote sensing filed is dominated by raster data, there was a desire for technology that can extract information from such data. Over two hundred application papers were listed and categorised in [10] in their comprehensive survey. They analysed papers related to deep learning within remote sensing field. Therefore, there was a vast variety of different approaches and architectures. The DL-based approaches were namely: CNN, autoencoder neural network, deep belief network and deep Boltzmann machine, recurrent neural network, and deconvolutional neural network. The number of approaches and their possible implementations are endless. According to [65], CNN are highly effective in semantic segmentation and object detection. The CNN are widely applied in a computer vision field. Since the given task of this thesis fits well into these categories, the research is further narrowed and focused only on the CNN architecture. The CNN are a special kind of ANN. The fact that CNN are capable of classification and detection makes them quite convenient, especially, because the classification and detection are one of the major tasks of remote sensing.

In [38], the authors argue that CNN-based studies in the remote sensing field achieved finer performance than conventional methods. They supported the statement with several papers solving the current tasks of remote sensing such as object detection, scene classification, large scale land classification, or hyperspectral image classification. There are some studies relevant for this thesis in particular. In [11], the authors described the process of detecting vehicles from high-resolution aerial imagery. Likewise, the application for automated building detection was decribed in [57]. It might not seem as relevant at first, but since such an algorithm could learn how to detect vehicles and buildings from high-resolution aerial imagery, detecting trees should not be much different. One of such examples is in [38], which described a palm tree detection process. The main issue in the papers was the fact that the architectures used were not able to deliver the exact shape of the object. The object was commonly bounded by bounding box. A solution for this type of issue was proposed by the Facebook AI Research (FAIR) team in [28]. The authors proposed a method called Mask R-CNN, which is the fourth generation of region-based convolutional neural network (R-CNN) capable of instance segmentation. The Mask R-CNN method is therefore capable of deriving a precise shape of a detected object, frequently known as the mask extraction. Hence, the Mask R-CNN method has great potential for single-crown delineation.

Chapter 2

Theoretical Background

2.1 Study Area

The Czech Republic is a landlocked country located in the middle of the temperate zone of the northern hemisphere in the central part of Europe. The area of the country is 78 868 square kilometres. The existing districts are grouped into 14 regions, including the city of Prague as an independent region with an area of 496 square kilometres, and a population of 1 308 632 (2018). The Vltava River flows through Prague (433 km). The average altitude of the Czech Republic is 430 meters AGL. The height and relief shape have a great influence on the climate of the Czech Republic. The climate of the Czech Republic is characterized by mutual penetration and mixing of oceanic and continental influences. Intense cyclonic activity causes frequent changes of air masses and relatively abundant precipitation. The average annual temperature is approximately 9°C [13].

Figure 2.1: Prague region of the Czech Republic

The flora and fauna of the Czech Republic follows a spread pattern similar to that of the rest of Cental Europe. Forests are mostly coniferous, occupying approximately 34% of the total area of the Czech Republic [13].

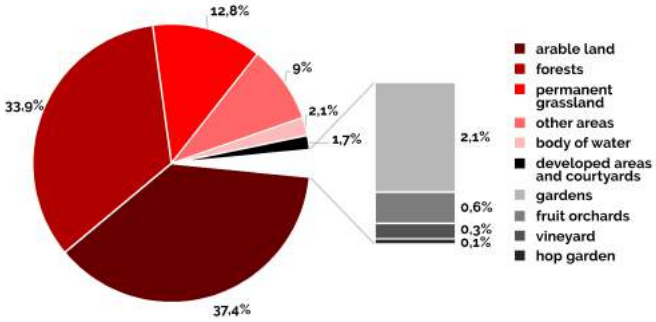


Figure 2.2: The overall land use percentage in the Czech Republic [14]

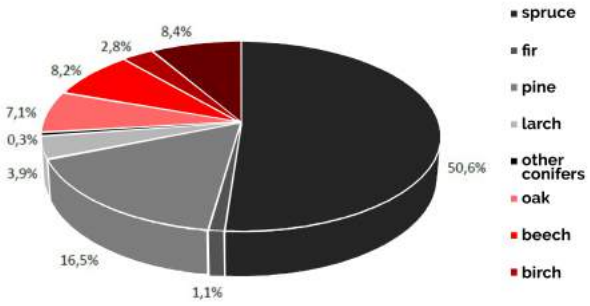


Figure 2.3: The overall tree species percentage in the Czech Republic [44]

According to [8], the last quantitative survey of vegetation in Prague was done in 1995. Total green area in Prague consists of natural parks (20%), forests (10%), protected areas (4%), street vegetation (NA%). The source could not be verified, thus this information must be taken with a pinch of salt, since there could not be found a newer quantitative study of such a character. Even though, there is a general trend of coniferous forests in the country. Cities are dominated by deciduous trees. According to the article written by the municipality's forest administrator [37], the species composition for afforestation is based on the natural composition of the original forest areas. Therefore, in 2015 there were planted approximately 180 000 trees, out of which 145 000 were deciduous and 35 000 coniferous. The species that occur are winter oak, European beech, cherry, linden, maple, hornbeam, elm, pine, larch, and fir. Assuming the initial trend, a rough estimation would be that 80% of the trees in Prague are coniferous trees. This trend is evident even from aerial imagery.

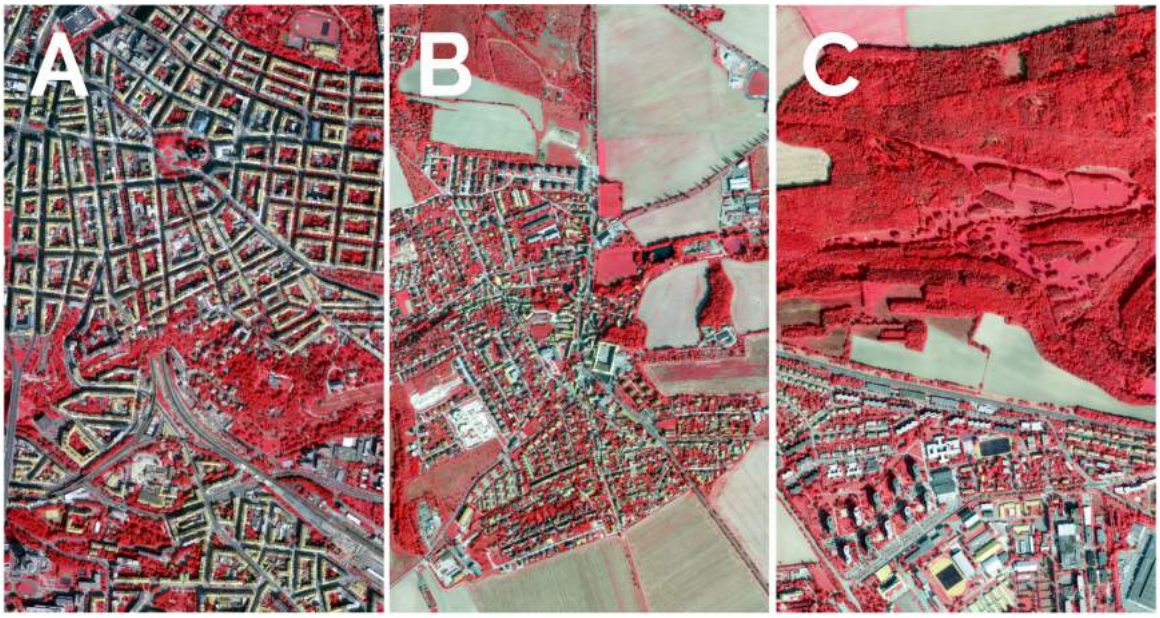


Figure 2.4: Different study areas: (A) Prague Vinohrady, (B) Kralupy nad Vltavou, (C) Beroun

2.2 Input

2.2.1 Color-infrared (CIR) imagery

The regular RGB image consists of three bands (channels). These are Red, Green, and Blue (RGB) bands. All these bands are from the visible part of the electromagnetic spectrum. Vegetation studies applying remote sensing use near-infrared (NIR) band with a wavelength longer the Red band, which is not detectable by human eyes. However, it can be measured by cameras and other instruments. Such extended information helps to distinguish the vegetation from other objects, especially, when we use vegetation indices. The NIR band in CIR imagery substitutes one of three colours from the visible spectrum to create a new representation of reality in the RGB color composite. In this thesis, the CIR imagery uses combination NIR, Red, Green. This means that Red is substituted by NIR, Green by Red, and Blue by Green. Therefore, the Blue band is not used, because it would not provide us with any piece of extra information. An example of the spectral behaviour of vegetation in individual bands is shown in the Figure 2.5.

2.2.2 Digital Terrain Model (DTM)

DTM is a digital topographic model of the Earth which can be digitally processed and visualised. The elevation information is stored either in a grid, or raster. The elevation is georeferenced and provides information of absolute height in a reference system. DTM does not include vegetation, buildings, or other manmade objects.

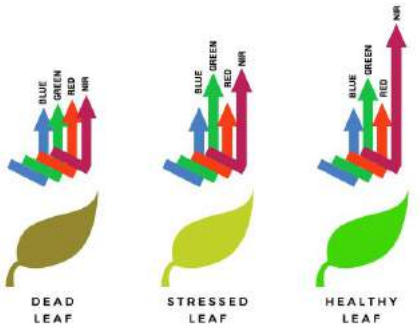


Figure 2.5: Reflectance in individual bands [4]

2.2.3 Digital Surface Model (DSM)

DSM is a digital topographic model of the Earth's surface. The definition is fairly similar to the DTM. Apart from DTM, DSM includes vegetation and manmade objects. Temporary objects such as cars are filtered out. A special kind of DSM is the Canopy Height Model (CHM). CHM is a digital topographic model of vegetation. Apart from DTM and DSM, the CHM stores relative heights, therefore CHM is useful for tree height determination. It is created by subtracting DSM and DTM. In urban areas, the DTM should include the buildings as well, however, the CHM contains vegetation only.

2.2.4 Acquisition

The data were provided by the Prague Institute of Planning and Development [33]. The acquisition was done by an external provider. The CIR imagery was acquired by an aircraft with mounted camera Vexcel Ultracam Eagle M; detailed specifications can be found in [1]. The layover was 60 meters/30 meters, and pixel size 0.1 meters. The DSM model was acquired from the same aircraft by LiDAR sensor and provided in the raster format with pixel size 0.25 meters. Both CIR and DSM were acquired on 31/8/2019. The month of August was chosen deliberately to detect the highest vegetation volume of the year. CIR imagery was later transformed into a true orthophoto (TO). The DTM was acquired in 2017 with 1 meter pixel size. The provider did not provide the exact approach.

Table 2.1: List of raster data and metadata

Name	**Type**	**Resolution**	**∼ Scale**
Color-infrared (CIR)	raster	0.1 x 0.1 m	1 : 1 000
Digital Surface Model (DSM)	raster	0.25 x 0.25 m	1 : 2 500
Digital Terrain Model (DTM)	raster	1 x 1 m	1 : 10 000
Name	**Data Type**	**Channel(s)**	**Acquisition**
Color-infrared (CIR)	8bit, uns. int	NIR, Red, Green	31/8/2019
Digital Surface Model (DSM)	32bit, float	Height	31/8/2019
Digital Terrain Model (DTM)	32bit, float	Height	2017

Figure 2.6: CIR

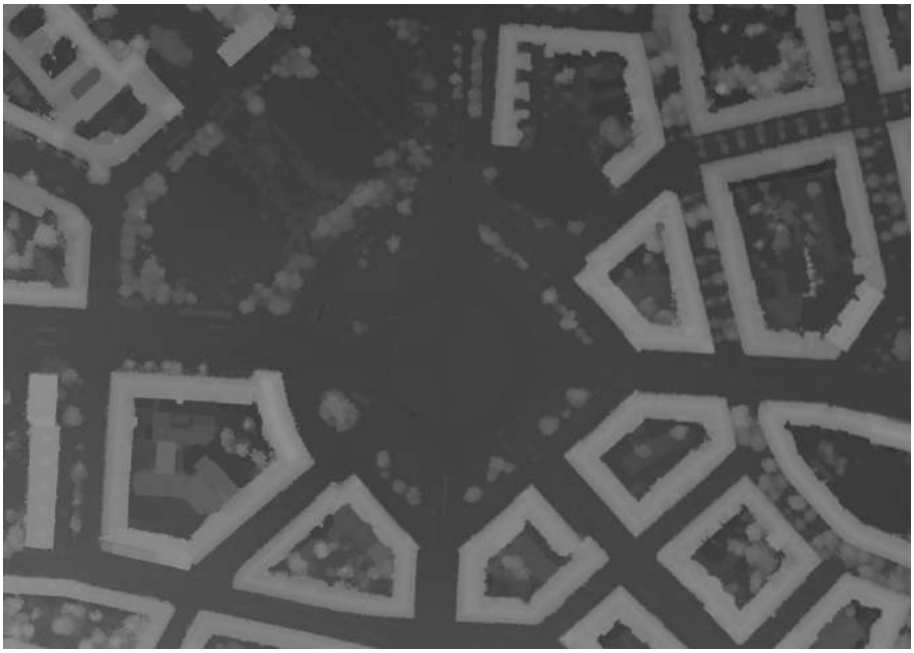


Figure 2.7: DSM

Figure 2.8: DTM and Buildings

2.3 Remote Sensing Techniques

Prior knowledge of image processing of the remote sensing data is assumed. This section does not attempt to replace the textbook, however, the following principles are explained on a certain level. This is needed to be able to understand the Methodology Chapter. The main source for this section was [27], therefore it is mentioned several times.

2.3.1 Spatial Filters

High-resolution imagery provides us with a high level of detail that is beneficial for such a study. However, the higher resolution does not provide us with better results in all cases. It is not just the resolution of the imagery that can be the reason for better or worse classification. Misclassification of pixels within an object can be caused by non-homogenous pixel values in homogenous areas. The high amount of detail causes noise and inconsistent objects. Another common issue raises the segmentation process. Too much detail in the imagery causes over-segmentation. Since the goal is to form segments containing objects we are looking for, the level of detail must be adjusted to a particular scale. By doing so, the objects should behave more consistently, and the image should be better segmented. There are several ways how to achieve such an adjustment. Some of them are pretty complex, some are fairly simple. The first one and probably the most straight forward is to simply resample the resolution of the raster data to lower pixel size. The resolution must be adapted to the size of the object.

A more sophisticated solution is the application of spatial filters. There are various reasons to filter the image; for instance noise reduction, image blur reduction, contrast

enhancement, or post-classification filtering. Filtering is a process by which the original pixel value is recalculation based on the values from its neighbouring pixels. This is performed by moving a window, also referred to as kernel, mask, or convolution matrix. Common kernel sizes are either 3x3, or 5x5 [27]. Accordingly, the new pixel value of a particular kernel is the value of the centre pixel from the kernel. Each type of kernel uses a specific equation. Generally, there are two filter groups. The first ones are low pass filters. Low pass filters are suppressing high spatial frequency values, which filters out outstanding details creating a smoother appearance. Widely applied examples are median filter, majority filter, or Gaussian filter. The other ones are high pass filters. On the contrary, the high pass filters are usually used for sharpening, often used for edge enhancement.

2.3.2 Fuzzy Membership

Fuzzy logic is based on the idea that there are not only logical TRUE or FALSE values, but that something can also be partially true. It quantifies how partially true it is. A fuzzy set is a set of elements, which belong to a set with a certain probability. Each element has a value which portrays the possibility of being a member of a specified set, i.e. membership value. The function that assigns the membership value is called membership function [27]. In remote sensing, the element is a pixel. A fuzzy set of pixels is a class. This means that every pixel belongs to every class to a certain degree. For instance, grey colour in greyscale is neither white (TRUE) nor black (FALSE). If there were just two classes, we would need to decide which one is correct - TRUE or FALSE. Fuzzy membership is the way how to distinguish uncertainty of the membership. Suppose the greyscale value of our grey is 127, white is 255, and black is 0. The value 127 is closer to the value 0 and therefore the final value would be equal to 0. The probability of grey colour belonging to the black class is slightly higher, assuming a linear membership function. Such an example is the most basic form of fuzzy classification of one class. In the first step, the fuzzy membership normalized data layer is created, i.e. $0 - 1$ scale. In the second step, the membership function is defined. When referring to the classification among the fuzzy system for several classes, the implementation is far more complex. Membership function used in the fuzzy classification defines the multi-dimensional space of image patterns. The dimensionality is given by the number of membership functions. This case is not analysed in this thesis. However, the concept of fuzziness is indeed interesting due to its nature, which is more realistic.

2.3.3 Thresholding

Thresholding is a process of transferring of the original set to a set with a lower number of elements. The threshold value splits the set into two intervals with new values. It is a form of a simple segmentation or the most basic classification. Since the elements are divided into two groups, or classes [27]. Thresholding is very often used to create a mask with values 0, and 1 in the map algebra. It helps to reduce the amount of redundant information. Accordingly, it is easier to predict the spectral behaviour of a specific class because the statistics are not affected by other classes. There is also a multiple thresholding called level slicing. The level slicing is equivalent to the thresholding, however, for more than two classes. The set is then divided into $n+1$ segments (classes).

2.3.4 Segmentation

In segmentation, the scene is partitioned into non-overlapping segments, also referred to as regions. The process has predetermined rules. The first example is the region growing. The principle of this method is relatively simple. This method requires an input of pixels (samples) of desired segments. These pixels or points are called seed points. The region growing method is driven by a control function and threshold value. The function compares neighbouring pixels with the value of a seed pixel and decides whether the new pixel belongs to the region or not. For instance, the function can be the difference of spectral values that is growing outwards until it reaches the threshold. Another widely used method in remote sensing is multiresolution segmentation. This method is far more complex. The segmentation is driven by spectral and spatial heterogeneity. The vital factor is a scale and the desired level of detail. Segments are also sometimes called superpixels. One example of segmentation is used in eCognition and is described in [9]. This method was revolutionary and fundamental for Object-based Image Analysis (OBIA). The object-based approach is essential for more advanced image analysis. The combination of vector representation with raster values is very advantageous because it combines the best from both worlds.

Even though the OBIA is still perceived as a product of software eCognition [54]. The OBIA principle can be applied in GIS as well. The term segmentation is quite loose, therefore all the previously mentioned remote sensing principles like fuzzy membership, or thresholding, can be turned into segmentation. There are many other ways how to perform segmentation in GIS. One of them is distance allocation even though it was not primarily intended as a segmentation tool. It uses pixel values to assess the neighbourhood and decides to which segment it belongs. In GIS is quite common to repurpose tools and functions an apply them to a different task, as long as the math behind can derive the desired result. Distance allocation, similarly to the region growing, uses seed points, or maybe more precisely; destinations. The segments are created according to the distance among the seed points (destinations). It can also take into account true surface distance, along with horizontal and vertical cost factors. In such a manner, we can achieve a segmented image by allocating distance over surface raster. Detailed description of how fuzzy membership, thresholding, or distance allocation can be used for segmentation can be found in Chapter 3.

2.3.5 Morphology

While working with the imagery at the object (segment) level, the morphological behaviour of an object is vital for a deeper understanding. Morphological properties like area, shape length, minimum bounding geometry, compactness, rectangularity and circularity, helps us to understand objects in a far broader perspective. Such properties can be easily calculated from elemental geometry properties. In GIS, every object has at least a shape and an area. Given a particular object, we might study its characteristic features to observe patterns in its behaviour. Based on the behavioural patterns, it is much easier to determine the class of a particular object.

2.4 Deep Learning

As mentioned earlier, DL is a subset of broad AI and ML family [51]. The hierarchy goes in this particular order: AI, ML, DL, ANN, CNN, R-CNN. The field of DL is extremely broad. Therefore, for this thesis, it is narrowed down to just CNN. More precisely, it is narrowed down to R-CNN, even more precisely to the Mask R-CNN.

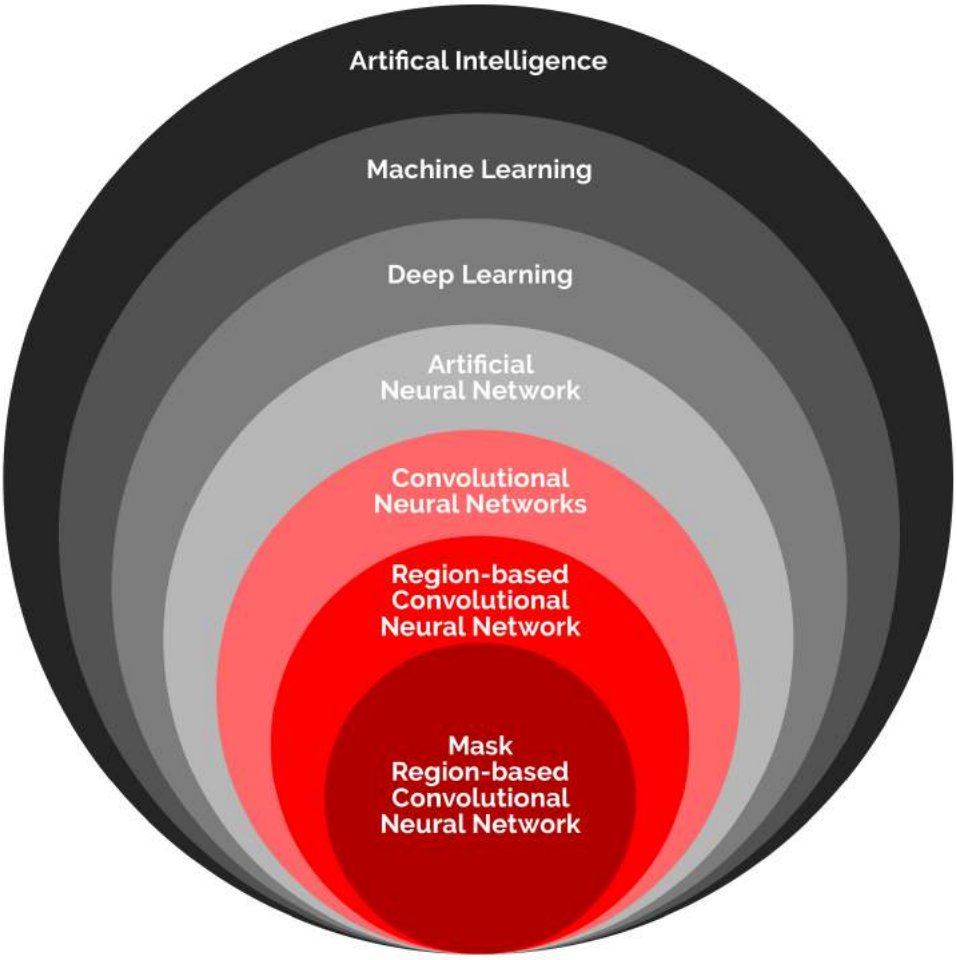


Figure 2.9: Hierarchy of Artificial Inteligence

2.4.1 Computer Vision and Convolutional Neural Networks

The following lines are devoted to the application of CNN on imagery in the computer vision field, although CNN are applied in other fields as well, for instance, natural language processing, automatic speech recognition, or other fields using data with grid-like topology.

This thesis is focused on the implementation in remote sensing. Computer vision deals with how the computer "sees" the information in imagery, unlike the human eye which transfers the visual information to the brain where the brain analyses and perceive the visual input as an outer environment with several categorised objects, the computer sees just an array of meaningless pixel values. Computer vision must, therefore, train the computers to understand and interpret the visual input. Using deep learning approaches, the computers are capable of identifying and classifying objects. First, the computer is looking for low-level features such as curves and edges, and afterwards it constructs higher abstract clue across a series of convolutional layers [2]. A gneneralised scheme of computer vision using neural network can be seen in Figure 2.11.

The fundamentals of CNN are biologically related. The inspiration comes from the visual cortex of the brain. Biological fundaments of CNN are based on Hubel and Wiesel's research on the vision of mammals [31]. The visual cortex has small groups of cells. Each group is sensitive to a particular part of an object. What is more, it could also react to the pattern of the visual input, i.e. orientation of the edges. They found that all neurons in the visual cortex are organised into columns and the combination of information from each neuron could produce visual interpretation. The principle of human vision can be seen in Figure 2.10.

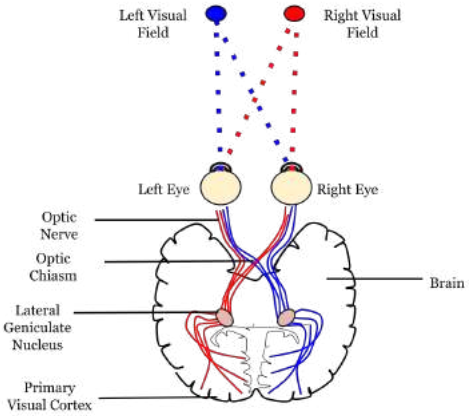


Figure 2.10: Human vision [40]

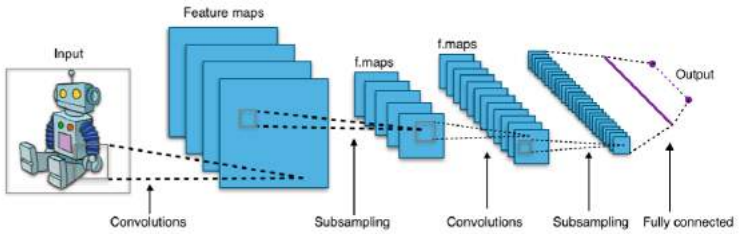


Figure 2.11: Computer vision example, illustrative scheme of neural network. [6]

The structure of CNN is organised into several layers. Each layer performs a different function. Layers are listed and briefly described in the following section, as well as their purpose and functions. Recommended source of more detailed literature can be found in [2].

2.4.1.1 Convolutional Layers

Convolution in geomatics is usually referred to as kernel which is a window of predefined size. The kernel is sliding or convolving, over the input raster. Pixels in the kernel are called the receptive field. The receptive field is multiplied by kernel filer values and returns one matrix element. This is done for every single location. The output array is called a feature map, or activation map. Dimensionality depends on the number of channels, which means that for the input with three channels, there are three feature maps [2]. If the original input was of size X by X, the feature map output would be *(X – kernel width + 1) by (X – kernel height + 1)* [43]. The reduced size also prevents the network from overfitting. This was the first convolutional layer.

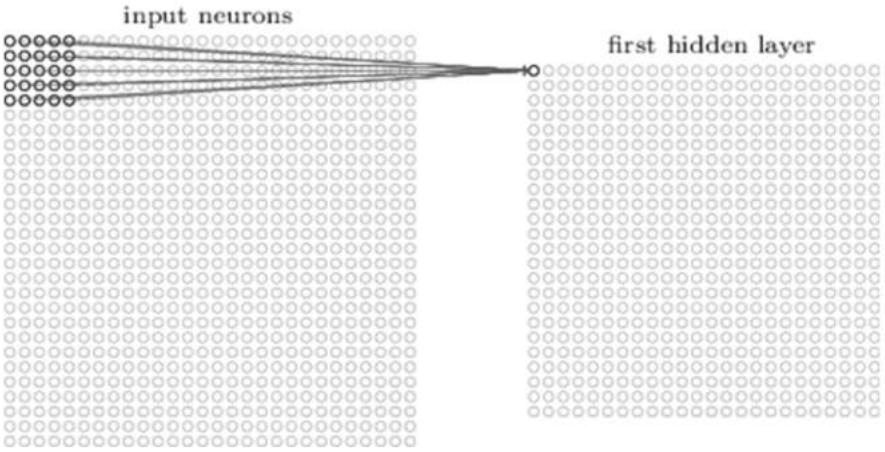


Figure 2.12: Kernel [43]

So far only one filter has been applied. Each filter is a feature identifier for a single purpose. The deeper we go, the more complex the filter is. The most basic filters are usually edge-detecting filers, etc. The output of the first layer is the feature map with low-level features. This serves as an input for another layer. The output low-level features from the first layer are connected through all channels. The output of the second layer is therefore connected to all previously detected features [2]. So, the output of the second layer provides us with higher-level features [25]. This goes on and on, depending on the number of different types of filters. Again, the deeper the layer, the higher the level of a feature.

2.4.1.2 Rectified Linear Unit (ReLU) Layers

The output of the ReLU layer is a rectified feature map. It applies a non-saturating activation function: $f(x) = max(0,x)$. It removes negative values from the feature map by replacing them with zero. It increases non-linear properties [42].

2.4.1.3 Pooling layers

The pooling layer has nothing to do with learning. Its main purpose is to either down sample or up sample the input size [2]. Therefore, they are also called subsampling layers. Again, the kernel is used for this purpose. The level of sampling is dependent on two parameters: the kernel size, and the stride (sliding step). The size reduction is usually done within the convolutional layers as well, although, the use of pooling layers is still beneficial because they do not cause the growth of parameters and since the detail is reduced, they lower the threat of overfitting [51]. A very common example of such a layer is the pooling with max-pooling function. The function extracts the maximum value from the kernel over the receptive field.

2.4.1.4 Normalization layers

These layers tackle an issue with sensitivity of deeper layers since they are highly dependent on the lower layers. Such an issue is called covariate shift. It could be resolved by lowering the learning rate, but that would make the training drastically slower. The computation in normalisation layers is done by using batches of training samples. This means that instead of one sample, several samples are processed at the same time. Such improvement reduces the computational steps and makes the training faster [43] because the learning rate can be higher.

2.4.1.5 Fully connected layers

All the neuron layers are connected to the neurons from previous layers. The main purpose of fully connected layers is to output classification vector from a high-level feature map. The classification vector represents a level of belonging to every single class [2]. The fully connected layer then backtracks every single neuron layer related to a particular high-level feature map and looks for the highest correlations to classes. Based on the highest correlation, the feature is assigned to a particular class.

2.4.2 Mask R-CNN

Mask R-CNN was proposed in 2017 by the FAIR team [28]. As previously mentioned, Mask R-CNN is the fourth generation of region-based convolutional neural network (R-CNN). It was built on top of the Faster R-CNN, which was the previous generation of R-CNN. The latest version is capable of instance segmentation. The Mask R-CNN method is therefore capable of deriving precise shapes of detected objects. It is also frequently known as mask prediction. Hence, the Mask R-CNN method is a great candidate for remote sensing application. The general architecture of CNN was described in the previous section. This section aims to describe the architecture of Mask R-CNN in particular. The architecture schema can be seen in the Figure 2.13. Further, the process within the network is described, as well as the key components. This section is using FAIR's paper as main reference [28].

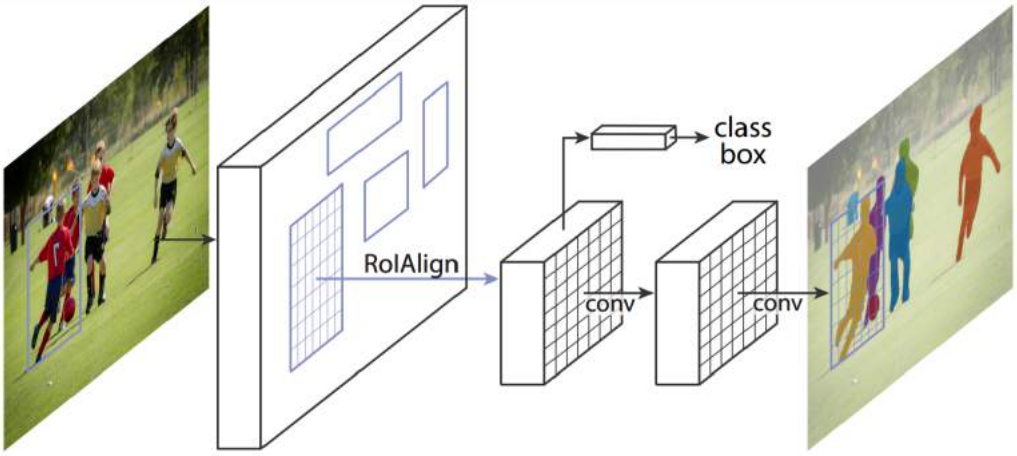


Figure 2.13: Mask R-CNN scheme [28]

Before we dive deeper into the architecture, we need to understand the underlying processes. Each such process is described, and the hierarchy is shown in Figure 2.14. Semantic segmentation is probably the most important one. It was already mentioned earlier as the biggest advantage of Mask R-CNN. It is a process composed of two different phases. The first phase is object detection. This means finding and classifying objects in an image within labelled bounding boxes. The other phase is a semantic segmentation. It is a classification approach on a pixel-to-pixel level. Thanks to the semantic segmentation, we can delineate a precise boundary of each object [28]. Combing both phases, we get the instance segmentation. The semantic segmentation is detection, classification, and delination simultaneously.

The object detection comes first and the instance segmentation afterwards. Region proposal network (RPN) is the part of the network responsible for the bounding box of an object. Since we have the bounding box of an object, we need to assign the class to the object, otherwise, we would have a detected object without knowing what kind of object it is. That is where the Region of interest Align (RoIAlign) takes place. Region of interest (RoI) is the detected object with a bounding box and class label on it. The RoIAlign is a pooling layer which takes RoIs from the feature map and down samples them into fixed size feature map. Align in the name means that unlike other pooling layers, the RoIAlign does not quantise the stride number. What usually happens is that the stride does have a remainder after division by kernel size. In that case, the number would be rounded off, which would mean loss of an information [28]. RoIAlign simply does not do that.

The object is now detected and the label is assigned. What is left is the delineation of the mask. A component used for that is the Fully Convolutional Network (FCN) described in [39]. The FCN is responsible for the semantic segmentation of every single RoI. The FCN uses CNN to transform image pixels into pixel categories. The key feature of the convolutional layer in FCN is the ability to retain spatial information. It is a substitution of

the fully connected layers.

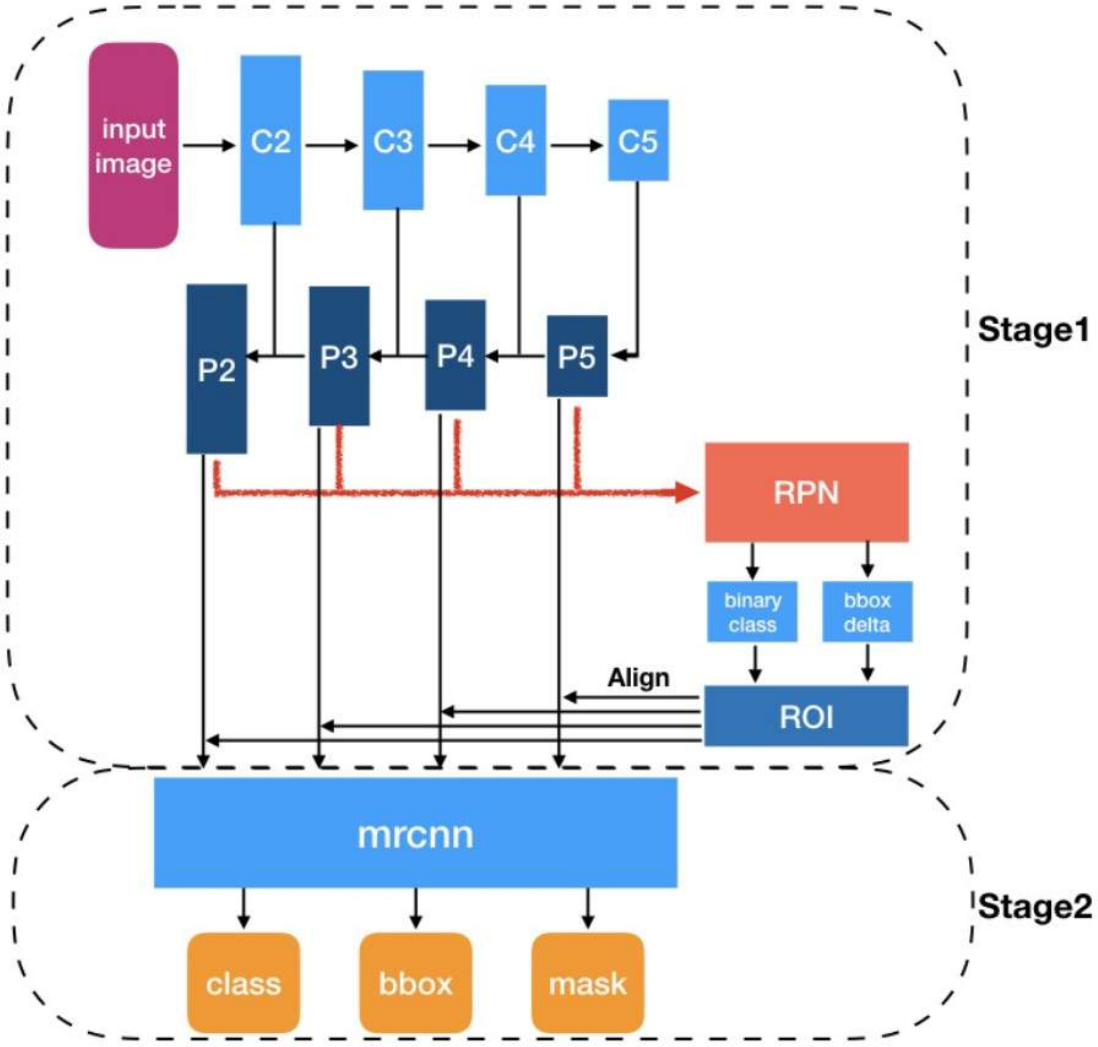


Figure 2.14: Hierarchy of processes of Mask R-CNN. Stage1 is a backbone architecture, and Stage2 is a head architecture [64]

In the network architecture, we differentiate between the convolutional backbone architecture and network head. Because the backbone architecture is not given to the Mask-RCNN, there are several possibilities on how to build up Mask-RCNN-based model. The backbone architecture is used for feature extraction, example in Figure 2.14 (Stage 1). The head is what makes the Mask R-CNN. The function of the head is to recognise the bounding box, assign a label, and predict mask for each RoI. The head architecture can be seen in 2.15 or in 2.14 (Stage 2).

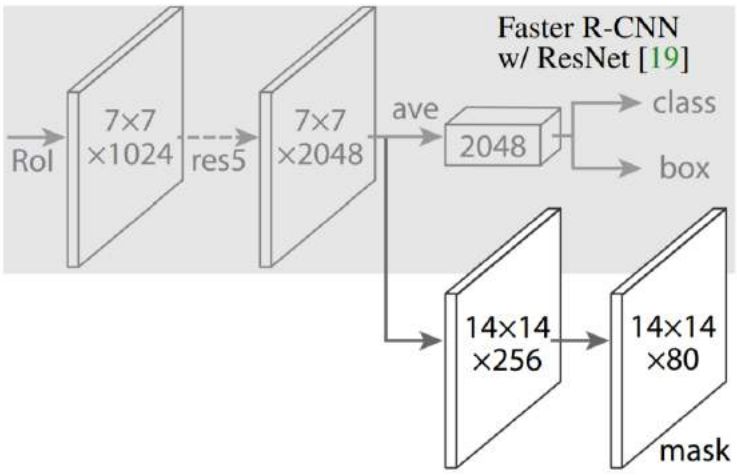


Figure 2.15: Head architecture of Mask R-CNN[28]

A backbone model in this thesis is the residual neural network (ResNet) described in [29]. More precisely, it is 34 layer deep model that was pre-trained on the ImageNET dataset with more than a million images [21]. Therefore, it is a preconfigured neural network. The process where the pre-trained model is used for training the new model is called transfer learning.

2.5 Software

The thesis proposes two different models for single-crown-detection and single-crown-delineation. The word model is used because both are using an input which is automatically processed and the output is returned. Such models are widely used in GIS. You can think of the model as a series of processes or self-standing tools combined in a hierarchical order. Since the model does not have its own graphical interface, it requires a graphical interface form GIS. This is in the form of either a code written in Python programming language, or diagram-alike form editable in the model builder environment.

Although, the tasks are mainly remote-sensing-related, the models were created in GIS. There is a variety of software specialising in remote sensing tasks. Some of these can deliver high-quality results. It is the versatility of the tools and fluency of transfer between raster and vector representation that make the GIS software advantageous. Plus, each year the GIS tools are becoming more and more specialised in the remote sensing field, regardless of the fact that strictly remote-sensing software provides "single" functionality. What is more, the high-end remote sensing software is usually quite expensive. GIS can cover most of the tasks of geomatics at once. Of course, this statement is greatly generalised and it depends on each user. This thesis used an ArcGIS pro platform for both models, which in terms of cost does not make much of a difference, but it is probably the most commonly used software in geomatics out there. There are several third-party dependencies listed below.

- **ArcGIS pro 2.5 [17]**
  ArcGIS pro is Geographic Information System by Esri.
- **Python 3 [48]**
  Python is a programming language.
- **Conda [5]**
  Conda is an open-source package management system and environment management system.
- **Tensorflow [53]**
  Tensorflow is an end-to-end open-source machine learning platform.
- **Keras [36]**
  Keras is a deep learning application programming interface.
- **PyTorch [23]**
  PyTorch is an open-source machine learning framework.
- **fastai [30]**
  Fastai is deep learning library.
- **scikit-image [59]**
  Scikit-image is a collection of algorithms for image processing.
- **Pillow [12]**
  Pillow is a Python imaging library.
- **LibTIFF [61]**
  LibTIFF software provides support for the Tag Image File Format (TIFF).

Chapter 3

Methods

3.1 GIS-based Model

There are three general tree classes used in this section. The first one represents the self-standing trees. The other two classes represent three clusters which were processed differently than the first class. The cluster classes were dense vegetation, and inner yards and street vegetation. Such a division was based on the most common forms of vegetation in the urban environment. The GIS model is composed of six phases: pre-phase, delineation of self-standing trees, delineation of dense vegetation and parks, delineation of inner yards and street vegetation, refinement, and classification. Each phase is described in a detail in the following section.

3.1.1 Pre-phase

Canopy Height Model
CHM was used in most studies described in the Literature section. This thesis is no exception. First, the DTM with buildings was resampled from 1 meter pixel size to 0.25 meters pixels size. The reason was to match the resolution of the DSM. Second, the DTM was subtracted from DSM (with buildings) using map algebra, see Figure 3.1.

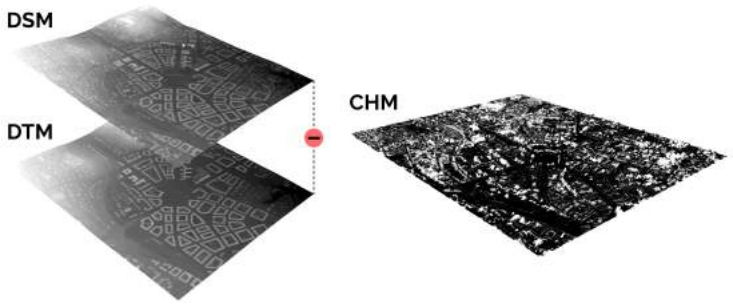


Figure 3.1: Map algebra - Canopy Height Model

Vegetation Mask
The vegetation mask took an advantage from the NIR band. It was extracted using a combination of two vegetation indices. The first was normalised differential vegetation index (NDVI), and the second was a modification of Red-Edge Triangulated Vegetation Index [26]. Therefore, it is referred to just as vegetation index no 2 (VI2).

$$NDVI = \frac{NIR - Red}{NIR + Red} \text{ [50]}$$

$$VI2 = \frac{100 * (NIR - Red)}{10 * (NIR - Green)}$$

Both indices complement each other. The VI2 is more efficient in areas where NDVI lags behind, and vice versa. For instance, VI2 is generally less sensitive to shadows. Application of both indices yielded in better results. Both indices combined performed better than just one of them.

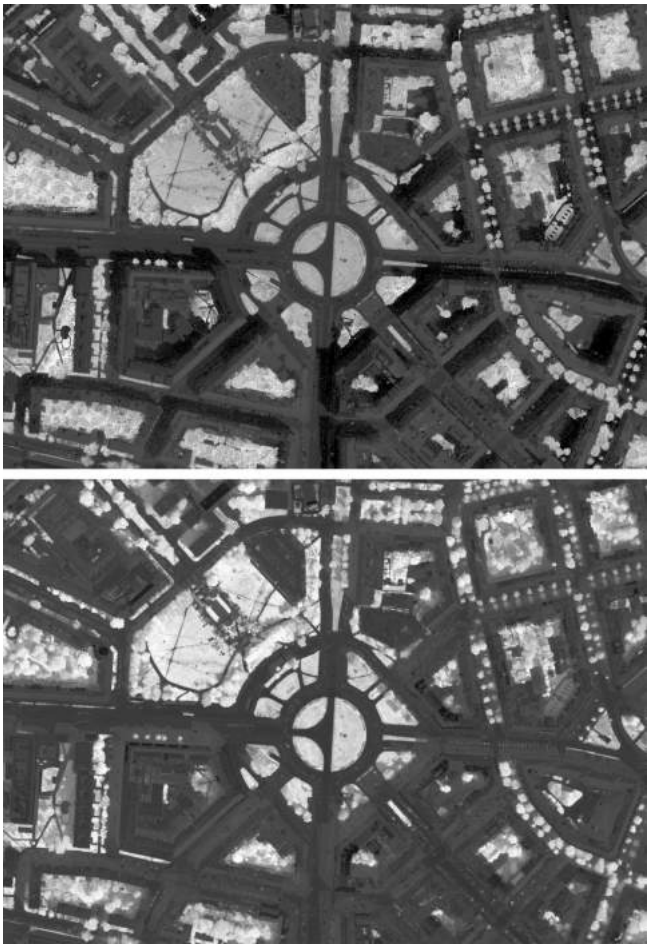


Figure 3.2: Comparison of vegetation indices. NDVI (up), VI2 (down)

Both index rasters were transformed into 0 to 1 scale. Each raster represented fuzzy membership, i.e. strength of membership in vegetation class. To achieve that, MSLarge fuzzification algorithm driven by *u(x)* function was used. This algorithm is useful when the large input values have a higher membership [20]. This was the case in both rasters.

$$u(x) = 1 - \frac{b * s}{x - (a * m) + (b * s)} \text{ [20]}$$

Where: x is a pixel value; m is the mean pixel value of the raster; s is the standard deviation of all pixel values from the raster; a is a multiplier (parameter) of the mean; b is a multiplier (parameter) of the standard deviation. Both multipliers were assigned to 1, so the weights were equal

As the next step, the fuzzy overlay was used. Fuzzy overlay analyses the memberships of multiple sets. This tool combines input data based on the selected fuzzy type. Fuzzy type OR returns the maximum value of a particular cell from both fuzzy rasters. Setting the threshold of a greater value or equal to 0 allowed to create a binary mask.

The mask was filtered by the majority filter to remove noisy pixels according to the majority of values in their neighbourhood. The boundaries of the mask were cleaned by the boundary clean tool. The result was a filtered binary mask. The last step of the pre-phase was to convert the filtered binary mask raster into a polygon. Morphological closing was performed to create a more circle-like shape representation. By simplifying the polygon, we achieved a hole-less and less morphologically opened polygonal representation of the vegetation.

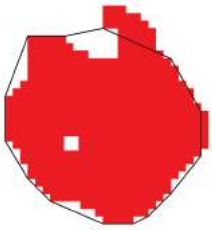


Figure 3.3: Morphological closing

3.1.2 Self-standing Trees

First Delineation
When the vegetation mask was created, there were many trees already delineated, mostly in the area of artificial surface. They were excluded from further processing. Such a polygon was considered either a self-standing tree or a cluster of trees. The decision was based on three morphological criteria: (a) compactness [27], (b) area, (c) average zonal height. In the first step, the compactness attribute of each polygon was calculated.

$$Compactness = \frac{shape\ length}{\sqrt{shape\ area}}$$

Threshold values were observed and set in the next step. Polygons below the threshold value of compactness and area were excluded. In the other step, the average zonal height was calculated for each polygon. Again, the threshold value was set and only polygons above were considered. The following thesholds were used: compactness ≤ 4, area < 200 m^2, zonal height ≥ 1.5 m.

Second Delineation
The heights greater than 2 meters were extracted from the vegetation mask in this subphase. It allowed to create the high vegetation mask. From this point, everything within the high vegetation mask was considered the tree class. Further splitting was always done on top of this mask. By excluding grassy areas and small shrubs, new potential self-standing trees were delineated because the trees growing on the grassy area could not be detected before. The second delineation was performed similarly to the first one except for a couple of changes. The selection was based the minimum bounding rectangular geometry of each polygon. As long as the side ratio of the bounding rectangle was within the threshold (≥ 0.8, ≤ 1.2), the polygons passed through. The area was also then taken into account (< 200 m^2). The rectangularity attribute was a bit milder in terms of compactness, therefore it detected even the crowns which did not go through the first subphase. The compactness with slightly milder threshold (≤ 6) was used to exclude objects with inappropriate shapes.

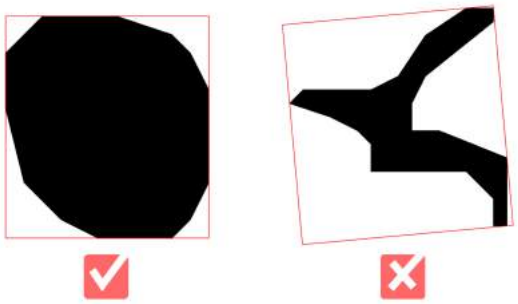


Figure 3.4: Morphologically correct shape

Third Delineation
The third subphase was just an extension of the first and the second subphase. Its main purpose was to detect remaining single trees which did not go through the previous subphases. It was mostly the circle-like shapes, but with rough edges. Therefore, the attribute of circularity was used in the third subphase. The circularity was defined as the percentage ratio of the polygon feature and the area of a minimum bounding circle. Polygons with the

circularity value lower than 65% were excluded, as well as the polygons with a greater area than 200 square meters.

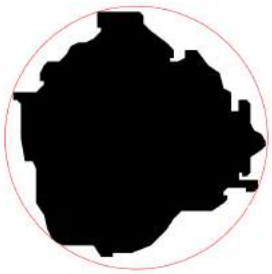


Figure 3.5: Circular shape with rough edges

3.1.3 Dense Vegetation

So far, only self-standing trees were delineated. This was the first phase focused on tree clusters. It was focused on large areas with dense vegetation such as parks and urban forests. The threshold value for a polygon to be treated as a dense vegetation cluster was a minimum area of 3000 square meters. These were probably the most complicated areas because there were various kinds of trees of various shapes and heights. The tree crowns were often interconnected, what made the solution more complicated. The main idea behind the GIS model in this thesis was to treat the single-crown-detection as a hydrological analysis and single-crown-delineation as a segmentation based on distance allocation.

The treetops had to be detected first, therefore the CHM had to be converted into a hydrological model where water cumulates in "pits". But first, the CHM had to be down sampled because the resolution was too high and there was too much detail, otherwise it would have caused over-segmentation in the next steps. Thus, it had to be resampled to reduce the complexity of trees. Different pixel sizes were tested. The best-fitting pixel size was observed from statistics and it was empirically verified. A pixel size of 0.75 meters was selected for dense vegetation. The CHM model was then multiplied by (-1) to create negative CHM (nCHM), i.e. hydrologically alike surface. Using the focal flow tool, a raster with flow accumulation was created. The highest values (255) represented potential treetops. In some places, the treetops were too close to each other, which would have caused over-segmentation, therefore the points within a distance tolerance of 1.5 meters were removed. Only one point for each potential crown was left.

The treetops were one of three inputs for the distance allocation segmentation (DAS). The other two inputs were cluster polygons serving as a boundary for the segmentation, and the CHM as a surface raster. The inclusion of CHM to DAS proved to be beneficial as it derived a more reliable representation of a crown shape.

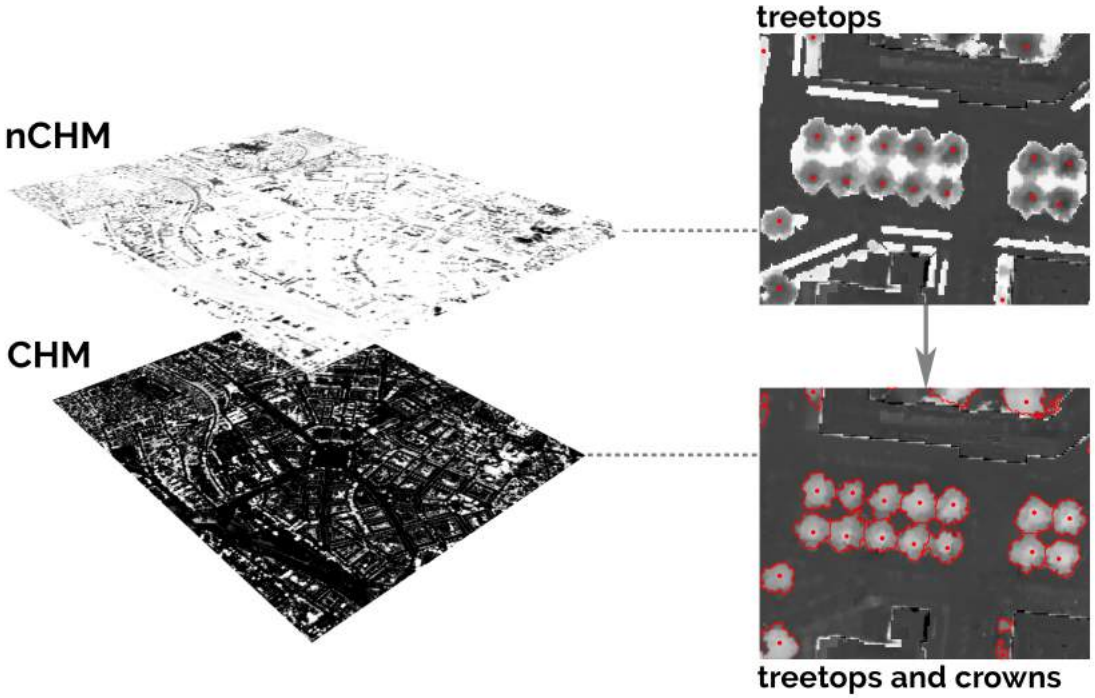


Figure 3.6: nCHM and the process of DAS

3.1.4 Inner Yards and Street Vegetation

The third phase was divided into two subcategories. Using zonal statistics, the standard deviation was calculated for each cluster left. Based on the assumption that the street vegetation (trees along streets) has a lower standard deviation because it is usually planted at once, which means the trees tend to have similar heights and are more likely of similar shape. On the other hand, trees within inner yards, differed from each other much more, thus the standard deviation tended to be higher. Thanks to such division, each category could be treated at a different scale. For the street vegetation, the size of the resampled pixel was set to 0.75 meters, and for the inner yards, the pixel size was set to 1.5 meters. Both values were assigned empirically and tested afterwards. Apart from the different scales, the method follows the same design as the previous phase, i.e. creation of the resampled nCHM, application of the focal flow, application of the point distance tolerance of 1.5 meters, and application of the DAS.

3.1.5 Refinement

All the delineated crowns were merged and the polygons were smoothed. The morphological attributes were verified again and polygons that did not meet the conditions were removed. For each crown, attributes of height and crown diameter were calculated. Each crown also carries coordinates of the tree trunk. The height was calculated as the maximum height value of each polygon using zonal statistics on top of the CHM. For the crown diameter, a circular

shape was assumed and was calculated from the crown area. The tree trunk coordinates were computed as centroids of each crown.

3.1.6 Classification

In the last phase, classification was performed. The classes were divided into Tree Crowns, which were already classified from the previous phases. Other classes were Shrubs, Grass Areas and Non-vegetation areas. Non-vegetation areas were FALSE values of the vegetation mask. The remaining classes were extracted from the vegetation mask using height threshold values: ≤ 0.5 m (Grass Areas), < 0.5 m ≤ 2 m (Shrubs). The result was a classified polygonal layer.

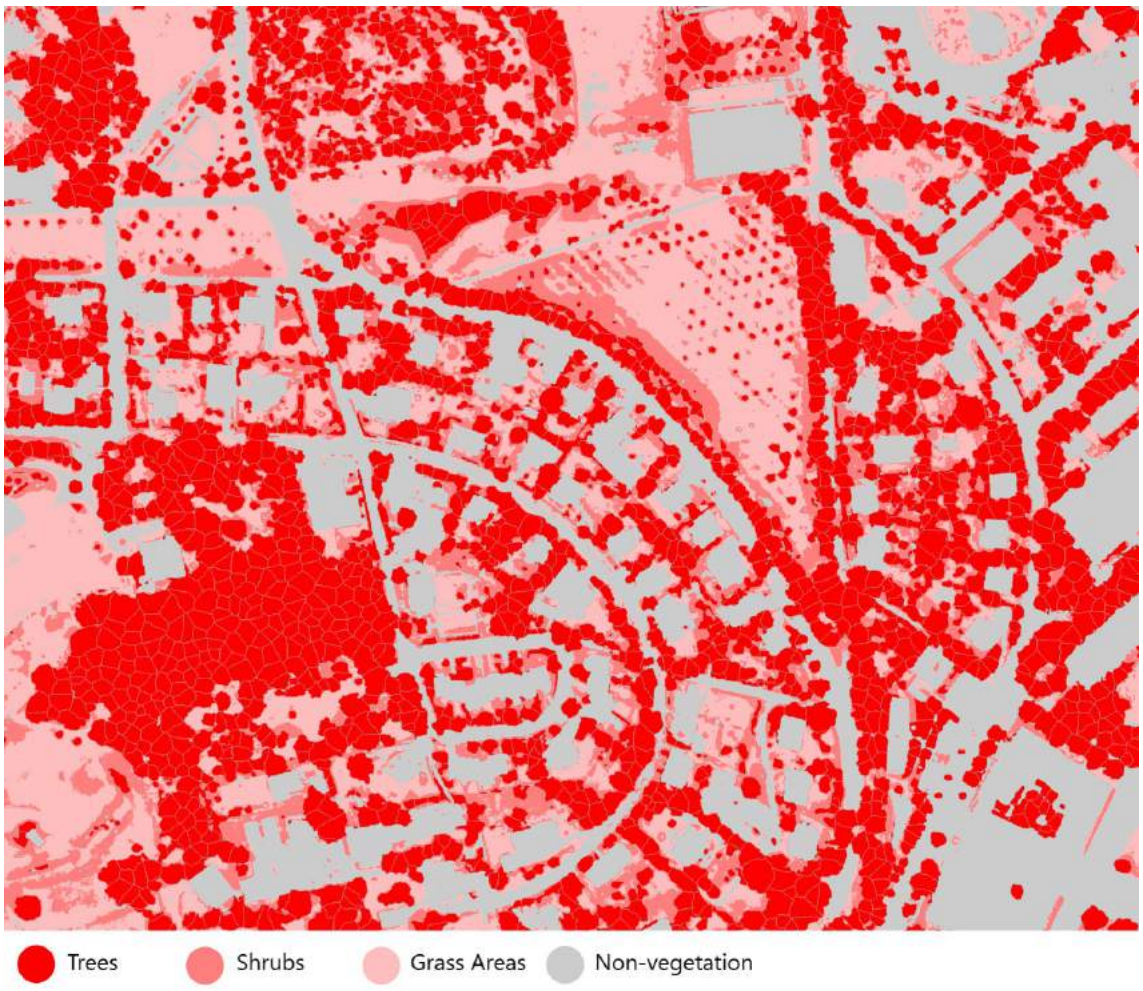


Figure 3.7: Classification

3.1.7 Model scheme

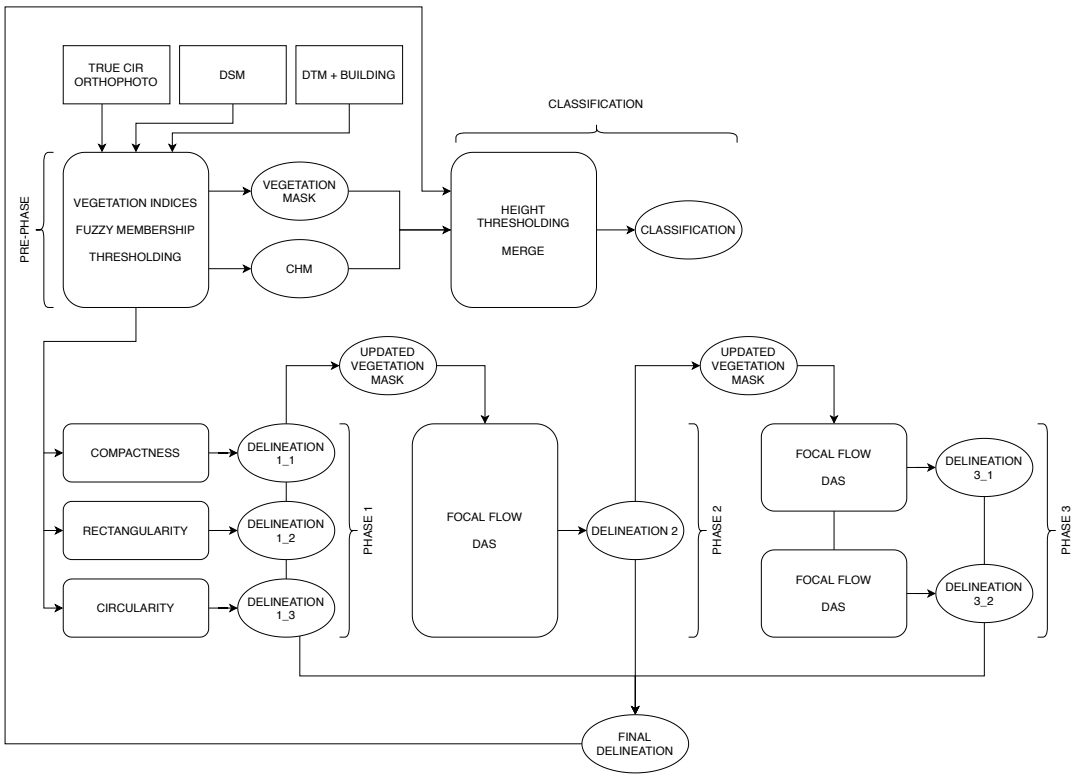


Figure 3.8: GIS Model scheme

3.2 Mask-RCNN-based Model

The problem with the training data set for the deep learning was that it did not exist. No available image dataset contains masked tree crowns. That was an issue for the training of the Mask R-CNN model, therefore the masks had to be created from scratch. One option was to create tree masks manually and spend days or even weeks by doing so. The other option was to create an automated model for tree-crown-delineation. This was a more reasonable choice, since the primary functionality of this model was to delineate crowns for the whole city. Masks from the GIS model were used as an input for the training of the Mask-RCNN model. In addition, the GIS model and the Mask R-CNN mask could have been compared to each other.

3.2.1 Pre-requirements

Deep learning, in general, is very demanding on computing power, therefore at least 6 GB of graphics memory is highly recommended. Mask-RCNN model was trained on NVIDIA

Quadro P1000 with 4 GB of graphics memory, 640 cuda cores, and compute capability 6.1. Even though the graphics memory was only 4GB, 640 cuda cores were sufficient enough. Cuda cores are capable of parallel processing, therefore GPU processing is much faster than the CPU, which has significantly fewer cores. The GPU processing is compatible only with CUDA-enabled GPUs, at least, in the ArcGIS pro (version 2.5) environment. Apart from the powerful GPU, the training requires several third-party dependencies. The dependencies were already listed and described in the Software section. The training also must have a working environment. It is a cloned python environment with all the dependencies in it.

3.2.2 Training data set

Training of the Mask R-CNN requires a lot of data, usually the more the better. The training data set in the case of Mask R-CNN consists of class training samples, called image chips. Each chip can contain one or more objects. Each chip has a raster label of the same extent that contain mask on existing object/objects. The format of the output metadata RCNN_Masks is based on Feature Pyramid Network (FPN) and a ResNet backbone [18].

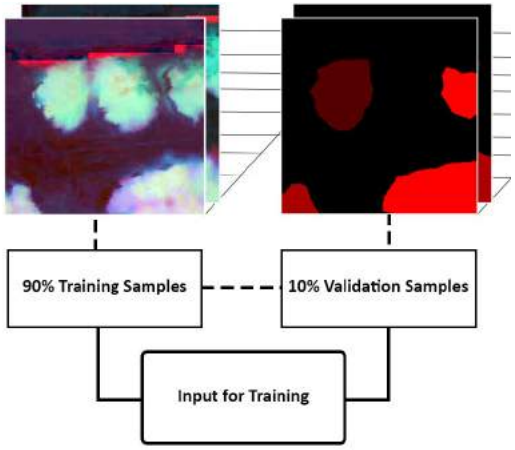


Figure 3.9: Training chips (Left) and masks (Right), each object within one chip has its mask and the different shades of red represent different objects. The training dataset consists of training and validation samples.

The input raster for training set must be an 8bit raster with 3 bands and a polygon or raster with masks representing the particular class [21]. In this thesis, the raster was composed of three 8bit int unsigned bands: NDVI, VI2, and CHM (Figure 3.10). Such a combination was chosen because it contained more information than a regular CIR. The NDVI and VI2 were already created from three bands and on top of that, the CHM could have been included as well. The polygon was derived from the GIS-based model. It is based on the following sentence: "What you see is what you get". This means band combinations where the objects are easily distinguishable for the human eye, will more likely be distinguishable for computer vision as well.

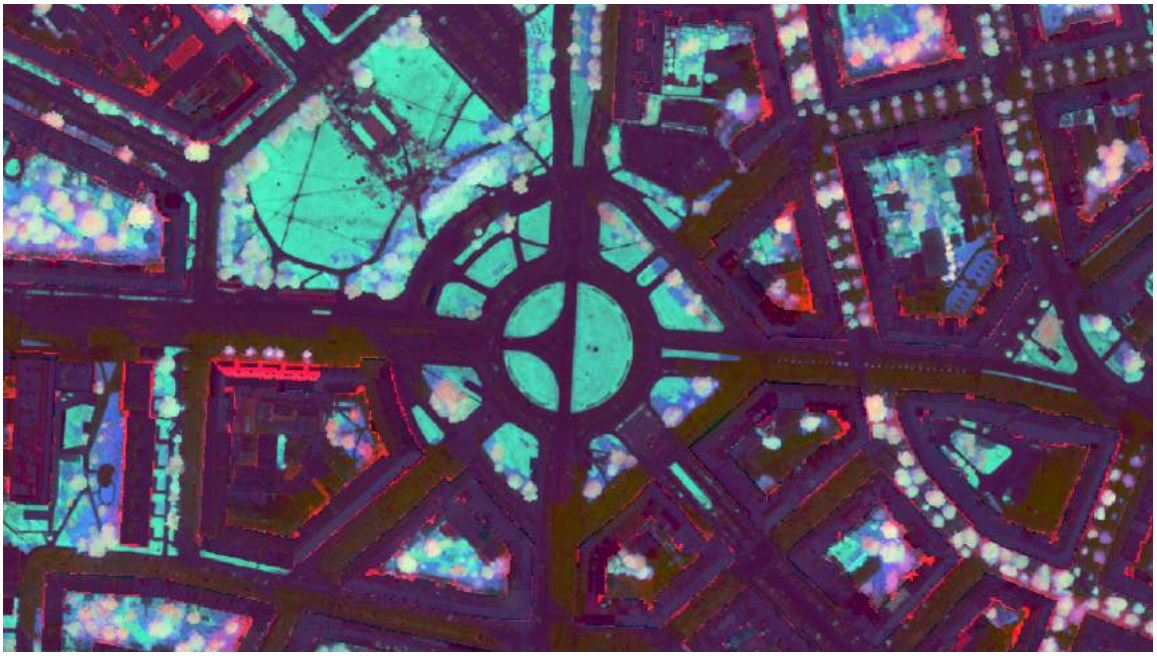


Figure 3.10: Composite of NDVI, VI2, and CHM

The training data set consists of three areas from the entire city, see Figure 2.4. Each area had a specific type of urban vegetation. The tree masks were delineated using the GIS model. For each area, the training data sets were exported separately and then merged. The reason was that the data had to be continuous within each data set. There were 38 510 of trees on 68 780 image chips in the *.TIFF* format at the end. The parameters used were tile size X = 256 pix, tile size Y = 256 pix, stride X = 128 pix, stride Y = 128 pix, rotation angle = 0 degrees.

3.2.3 Training Mask-RCNN model

The training uses PyTorch deep learning framework [18]. The input is the training data set from the previous section. There are several parameters which affect the training phase. The maximum number of epochs affects how many times the dataset passes forward and backwards through the neural network. The number of epochs was set to 20. Another parameter which drastically affects the duration of training is a batch size. The batch size is the number of training samples processed at the same time. The graphic card used for the training could not apply a higher value than 2. With a more powerful GPU, the training would have been much faster. The time needed for training was almost four days with the ResNet 34 preconfigured backbone model. The other possible option for backbone model was ResNet 50 which is 50 layers deep. The ResNet 34 is 34 layers deep [21]. Another parameter is the learning rate, which is a rate of overwriting the old information with the newly acquired information. The optimal learning parameter was extracted automatically from the learning curve during training. The last parameter is the validation percentage.

Based on the set value, the percentage of the training sample is used for validation. The value was set to 10%. The output folder contains training statistics and model definition JSON file. A path to the trained binary deep learning model and a path to the Python raster function for object processing is stored in the model definition file. The parameters used are listed below.

- Max epochs = 20
- Batch size = 2
- Backbone model = Resnet 34
- Validation = 10%

The model is considered trained usually after the training stops improving. The model is gradually improving with the increase epochs to a certain point where the improvement is no longer perceptible. It is also common that the improvement starts decreasing after some time, which is not in our interest. In ArcGIS pro, the model is considered trained after the validation loss stops improving for 5 epochs [18]. The validation loss is based on the 10% of training samples which were set aside for validation. Validation loss is a function comparing the training and validation set. ArcGIS pro documentation does not provide the exact loss function used. The reached loss is drawn into a graph with processed batches, see Figure 3.11. The decision whether the model was properly trained or not had to be considered on two levels since the training data is derived from the GIS model, i.e. not 100% accurate. So the assessement form ArcGIS pro's graph had to be verified on ground truth data. The accuracy assessment is described in Section 3.3.

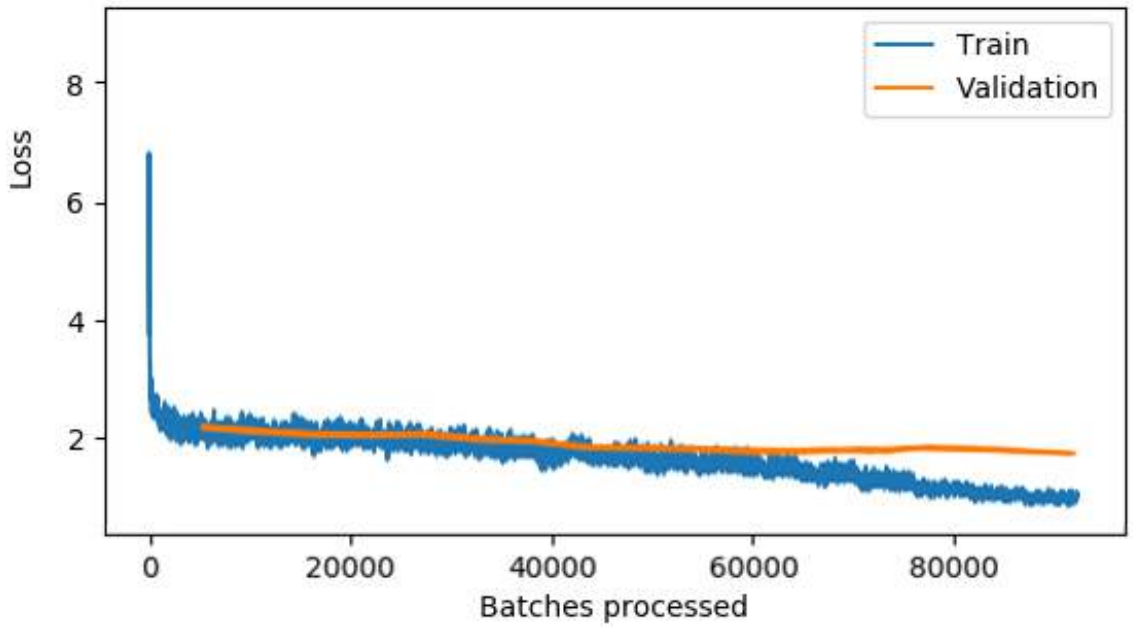


Figure 3.11: Output model statistics from ArcGIS pro's framework

3.2.4 Running Mask-RCNN model

Since we have the trained model, it can be used for object detection. The input parameters are input raster, a path to the file where to store detected objects, model definition file, model arguments, and the non-maximum suppression. The output layer is a polygonal representation of delineated tree crowns. Each crown has a confidence attribute, i.e. from 0 to 1 of the membership probability. The non-maximum suppression chooses the object with higher accuracy when two or more detected object overlap, the maximum overlap ratio can be specified. The model arguments are then padding size, batch size, confidence threshold, and return bounding box. The padding is the pixel window size added to an image chip when processing by the kernel. The batch size is how many chips are being processed at once. The confidence threshold is the threshold value for each potential detected object. The objects of lower confidence than a threshold are not in the output. The bounding box parameter (TRUE/FALSE) decides whether we want the output of bounding boxes or the precise shape of an object [19]. The parameters used are listed below.

- Padding = 56
- Batch size = 4
- Confidence threshold = 0.9
- Return bounding box = FALSE
- Non-maximum suppression = 0.2

3.3 Accuracy Assessment

The accuracy assessment was processed by the method from [38]. The methodology presented by the authors used a combination of *Precision* and *Recall* assessment criteria. The assessment in this thesis was extended for *Quantity match* criterion. The overall accuracy is then calculated as an average of all three assessment criteria. The criteria were computed from the following variables: (1) The number of crowns in the ground truth was the total number of ground truth samples; (2) The number of all detected crowns was the total number of detected crowns in a particular area; (3) The number of correctly detected crowns was based on the centroid position of the ground truth crowns. If the detected crow was intersected by just one centroid for the ground truth, the detected crown was considered correct.

$$Quantity\ Match = \frac{(2)\ The\ number\ of\ all\ detected\ crowns}{(1)\ The\ number\ of\ crowns\ in\ the\ ground\ truth}$$

$$Precision = \frac{(3)\ The\ number\ of\ correctly\ detected\ crowns}{(2)\ The\ number\ of\ all\ detected\ crowns}$$

$$Recall = \frac{(3)\ The\ number\ of\ correctly\ detected\ crowns}{(1)\ The\ number\ of\ crowns\ in\ the\ ground\ truth}$$

$$Overall\ Accuracy = \frac{Quantity\ Match + Precision + Recall}{3}$$

Because the accuracy assessment was based on validation with ground truth data, the validation data was manually delineated from three representative areas. Each area represented different kind of urban vegetation: (A) Dense vegetation, (B) Parks, and (C) Street vegetation and inner yard vegetation. There were over 1 300 manually delineated crowns.

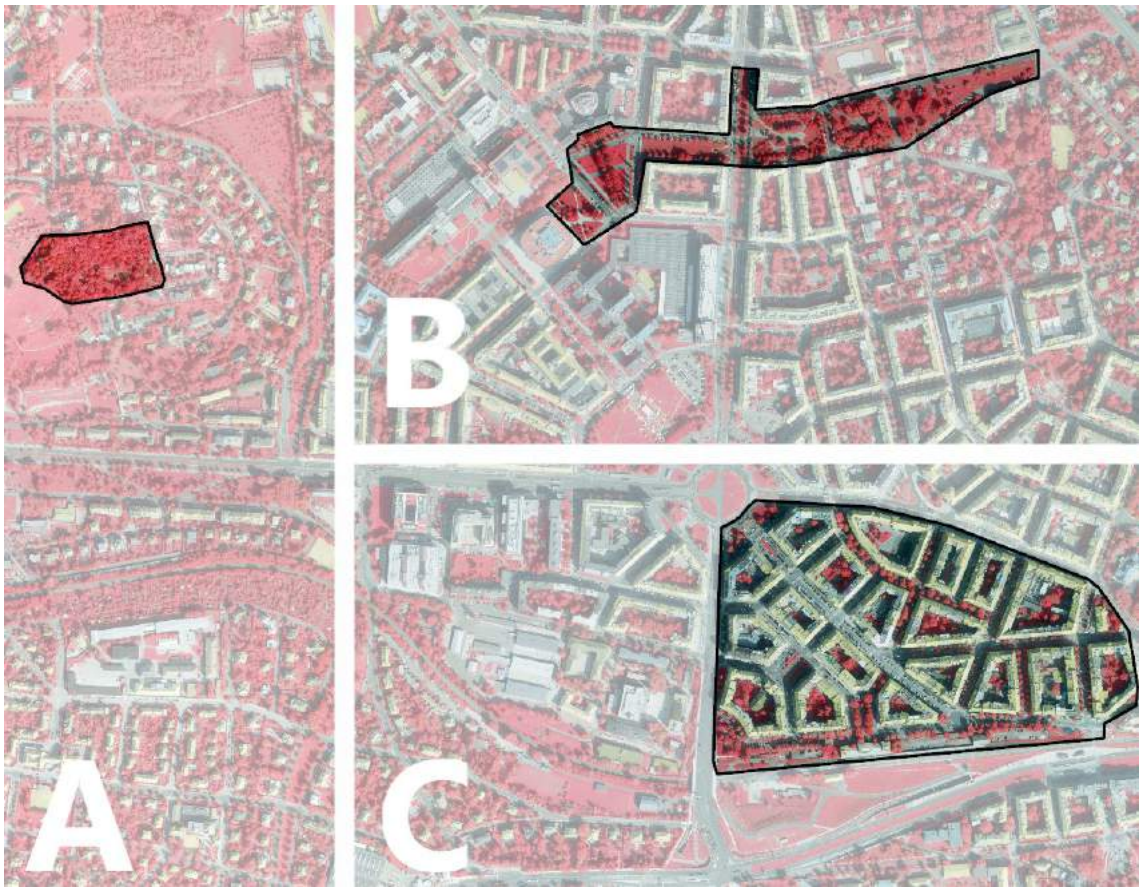


Figure 3.12: Ground truth validation areas; (A) Dense vegetation, (B) Parks, and (C) Street vegetation and inner yard vegetation

Chapter 4

Results

Delineated crowns from both models were compared with the ground truth data. The comparison was based on the results from the accuracy assessment. Mask R-CNN Model with crowns of confidence higher than 80% was the best one. The overall accuracy reached 81%. The same model with crowns of confidence higher than 90% finished in the second place. The overall accuracy reached 77%. GIS model ended up with 72% overall accuracy. These values are an average value from the overall accuracies from all three validation areas, see Table 4.3. The total number of detected crowns is in Table 4.1 and the overall accuracy from each area in Table 4.2. Comparison of both models in all three areas can show Figures 4.1, 4.2, and 4.3. The results are further discussed in Chapter 5.

Table 4.1: Accuracy assessment: Number of detected trees

Dense Vegetation			
Model	**Ground Truth**	**Detected**	**Corectly Detected**
GIS	291	247	175
Mask R-CNN 90%	291	240	202
Mask R-CNN 80%	291	264	218
Park			
Model	**Ground Truth**	**Detected**	**Corectly Detected**
GIS	389	322	248
Mask R-CNN 90%	389	336	280
Mask R-CNN 80%	389	361	298
Inner Yards and Street Trees			
Model	**Ground Truth**	**Detected**	**Corectly Detected**
GIS	678	562	395
Mask R-CNN 90%	678	569	413
Mask R-CNN 80%	678	648	446

Table 4.2: Accuracy assessment: Accuracy percentage

Dense Vegetation				
Model	**Quantity Match**	**Precission**	**Recall**	**Overall Accuracy [%]**
GIS	0.85	0.71	0.60	72
Mask R-CNN 90%	0.82	0.84	0.69	79
Mask R-CNN 80%	0.91	0.83	0.75	83
Park				
Model	**Quantity Match**	**Precission**	**Recall**	**Overall Accuracy [%]**
GIS	0.83	0.77	0.64	75
Mask R-CNN 90%	0.86	0.83	0.72	81
Mask R-CNN 80%	0.93	0.83	0.77	84
Inner Yards and Street Trees				
Model	**Quantity Match**	**Precission**	**Recall**	**Overall Accuracy [%]**
GIS	0.83	0.70	0.58	70
Mask R-CNN 90%	0.84	0.73	0.61	72
Mask R-CNN 80%	0.96	0.69	0.66	77

Table 4.3: The average overall accuracy

Model	**Overall Accuracy [%]**
GIS	72
Mask R-CNN 90%	77
Mask R-CNN 80%	81

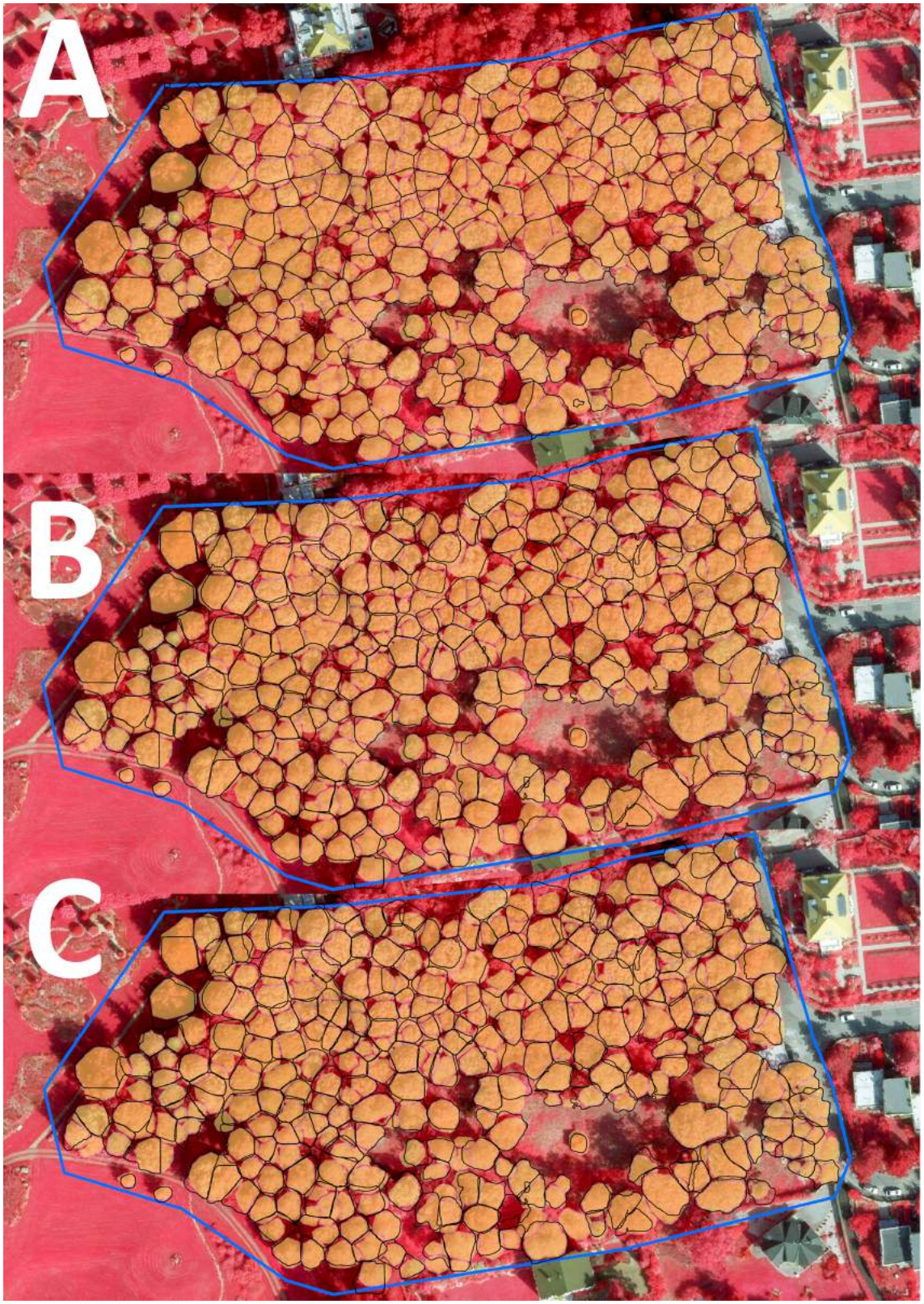


Figure 4.1: Ground truth (Orange) validation in the dense areas with the delineated cronws (Black) from: (A) GIS Model, (B) Mask R-CNN 90%, and (C) Mask R-CNN 80%

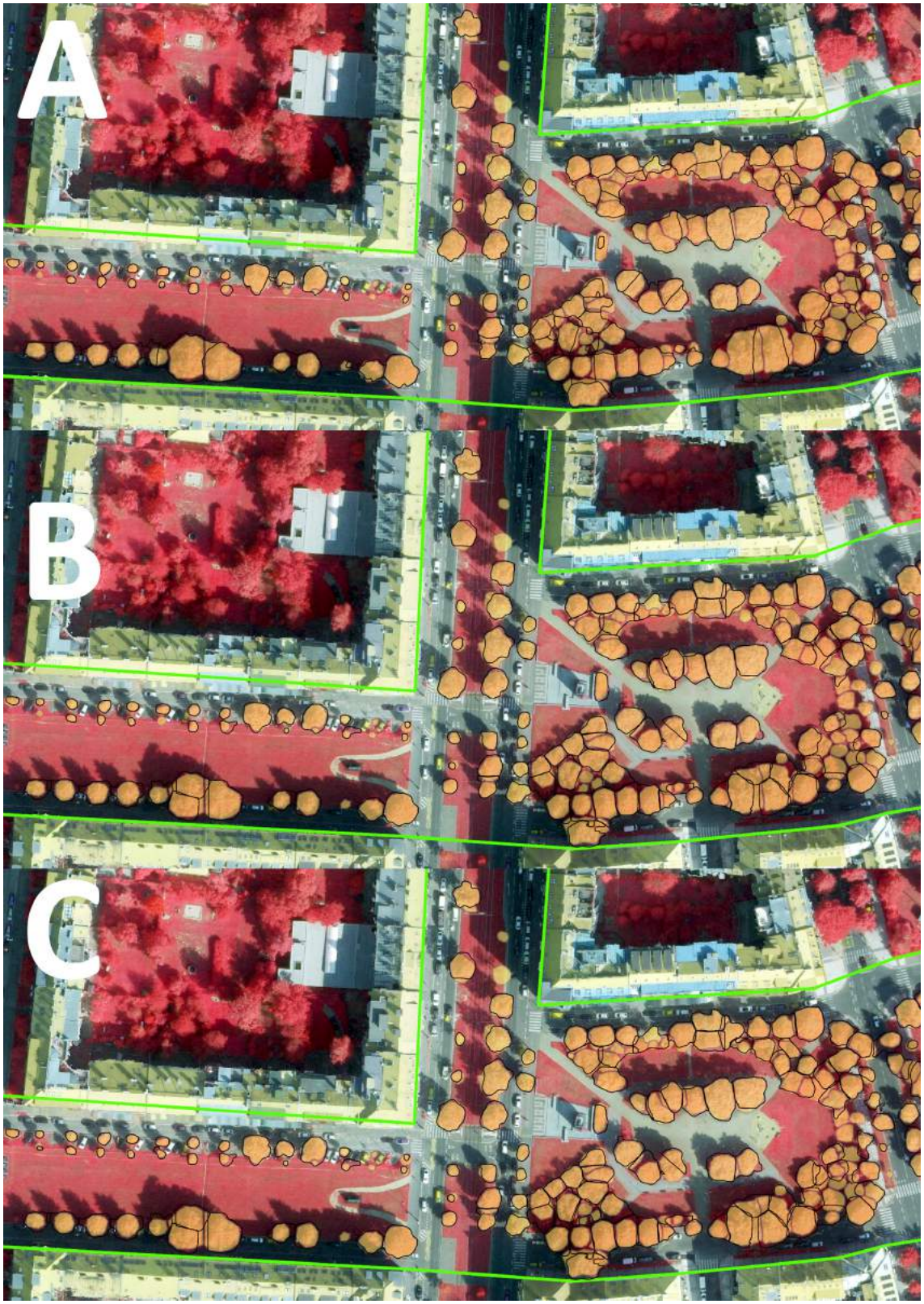


Figure 4.2: Ground truth (Orange) validation in the park areas with the delineated cronws (Black) from: (A) GIS Model, (B) Mask R-CNN 90%, and (C) Mask R-CNN 80%

Figure 4.3: Ground truth (Orange) validation in the street and inner yard areas with the delineated cronws (Black) from: (A) GIS Model, (B) Mask R-CNN 90%, and (C) Mask R-CNN 80%

Chapter 5

Discussion

The highest overall accuracy of 81% may not seem like the best at first sight. However, it is especially important to mention factors that contributed to this result. First of all, the Mask R-CNN model is based on training data that are already loaded with a certain error of the GIS model. So, instead of 100% correct data, the training is based on a prediction of correct data. This fact is not ideal, however, it is inevitable if we do not have a large data set with 100% correctly masked trees. This approach seemed to be the most feasible and confirmed the assumption that this method can produce more accurate results than the original data that even if the training does not use 100% correct data. This is a very valuable finding.

In general, if the data represents distorted reality, then the result will also be a partially distorted representation of reality. An example of this phenomenon, and at the same time another factor, were the input TO data. Because the process of their creation is automated, sometimes the quality of the original input is significantly reduced. The buildings that were transferred to the TO left their original parts in their original locations, therefore a false image parts and height data appeared. In some cases, some trees were partially or completely filtered out due to automated processing, i.e. Figure 5.1. As a result, the mentioned places did not behave like trees anymore, thus, could not be detected. Sometimes, the building "leftovers" cause a wrong representation of tree height as an attribute.

Another factor of assessing the accuracy was the human factor. The collection of validation data was not based on field measurements. Regardless of the effort, it is theoretically possible that in some cases the ground truth data was also burdened by human error. This factor was not considered to make the assessment possible.

Understanding external influences have a major impact on understanding overall accuracy as such. So it's not always just the fine-tuned parameters and attributes that lead to the correct result. Since the model was designed for detection of the trees of such a complex and diverse nature, regardless of all the negative influences, the final score of 81% is indeed success. In addition, it is a great starting point for further research.

Figure 5.1: Mistakes in the DSM. Clearly visible trees from the CIR imagery (Left), completely filtered out trees form the DSM (Midle), and the image from the street (Right).

5.1 Improvements and further research

The main purpose of the models in this work was to be able to detect individual trees throughout the city of Prague. It will be possible to train the model on more data and deterrmined crowns with the incresing data volume, therefore the Mask R-CNN model can be even more robust. It pays for the cases when a deep-learning-based model for a particular purpose is processed. It can also be used as a base model for training of a model with similar purpose.

Furthermore, a certain area could be mapped in more detail for better verification. The models' accuracy could then be judged more accurately.

The obtained tree crowns have great potential in long-term studies of vegetation condition and development. With the help of detected crowns, it is possible to monitor, for example, the state of vegetation during drought and its development in recent years. This study can also be done retrospectively thanks to a huge amount of freely available satellite data. For example, it would be possible to map Prague's vegetation and its development over the last ten years in individual months. From the imagery, it would then be possible to observe trend development and predict the future state.

Chapter 6

Conclusion

Author of this thesis designed and implemented two methods for single-crown-detection and single-crown-delineation. The first method used GIS-based tools and remote sensing techniques composed into an automated process. The other method was based on deep learning techniques using Mask R-CNN neural network framework. The process of implementation and training was described. Both methods were tested in a case study delineating crowns in the city of Prague, in the Dejvice district. The accuracy assessment using ground truth data was designed and described. The Mask R-CNN based method achieved a higher overall accuracy of a decent 81%. The GIS-base method achieved a 72% overall accuracy. The Mask R-CNN based method proved its potential in vegetation-related studies and remote sensing field in general because it outperformed the GIS Method, which represented a more traditional approach in current remote sensing.

Bibliography

[1] Aerial-survey-base. UltraCamEagle - Technical Specifications. *Microsoft*, 2011, retrieved 2020. URL https://aerial-survey-base.com/wp-content/uploads/2018/07/UltraCamEagle-M1-Specs.pdf.

[2] C. C. Aggarwal. Neural Networks and Deep Learning. *Neural Networks and Deep Learning*, 2018. doi:10.1007/978-3-319-94463-0.

[3] I. N. Aizenberg, N. N. Aizenberg, and J. Vandewalle. Multi-Valued and Universal Binary Neurons. *Multi-Valued and Universal Binary Neurons*, 2000. doi:10.1007/978-1-4757-3115-6.

[4] ALTAVIAN. What is CIR Imagery and what is it used for? *ALTAVIAN*, 2016, retrieved 2020. URL https://www.altavian.com/blog/2016/8/cir-imagery.

[5] Anaconda. Conda. retrieved 2020. URL https://docs.conda.io/en/latest/.

[6] Aphex34. Typical CNN architecture. *CC BY-SA 4.0*, retrieved 2020. URL https://commons.wikimedia.org/w/index.php?curid=45679374.

[7] R. J. Argamosa, E. C. Paringit, K. R. Quinton, F. A. Tandoc, R. A. Faelga, C. A. Ibañez, M. A. Posilero, and G. P. Zaragosa. Fully automated GIS-based individual tree crown delineation based on curvature values from a LiDAR derived canopy height model in a coniferous plantation. 2016. doi:10.5194/isprsarchives-XLI-B8-563-2016.

[8] ARNIKA. Kolik stromů je v Praze? *ARNIKA*, retrieved 2020. URL https://arnika.org/kolik-stromu-je-v-praze.

[9] M. Baatz and A. Schape. Multiresolution segmentation - An optimization approach for high quality multi-scale image segmentation angewandte geographische informationsverarbeitung XII. *AGIT Symposium*, 2000.

[10] J. E. Ball, D. T. Anderson, and C. S. Chan. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, 2017. doi:10.1117/1.jrs.11.042609.

[11] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 2014. doi:10.1109/LGRS.2014.2309695.

[12] A. Clark and Contributors. Pillow (PIL Fork). retrieved 2020. URL https://pillow.readthedocs.io/en/stable/.

[13] Czech Environmental Information Agency. Statistická ročenka životního prostředí ČR 2018. *Ministerstvo životního prostředí*, 2018. URL https://www.cenia.cz/wp-content/uploads/2020/01/Statisticka_Rocenka_ZP_CR-2018.pdf.

[14] Czech Environmental Information Agency. Zpráva o životním prostředí České republiky 2018. *Ministerstvo životního prostředí*, 2018. URL https://www.komora.cz/files/uploads/2019/09/ma_ALBSBGAJ8984.pdf.

[15] M. Dalponte, L. Frizzera, and D. Gianelle. Estimation of forest attributes at single tree level using hyperspectral and als data. *In Proceedings of the ForestSAT*, 2014.

[16] Definiens AG. Definiens professional 5 user guide. *Definiens Cognition Network Technology*, 2006.

[17] Esri. ArcGIS pro 2.5. retrieved 2020. URL https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview.

[18] Esri. arcgis.learn module. retrieved 2020. URL https://developers.arcgis.com/python/api-reference/arcgis.learn.html.

[19] Esri. Detect Objects Using Deep Learning. retrieved 2020. URL https://pro.arcgis.com/en/pro-app/tool-reference/image-analyst/detect-objects-using-deep-learning.htm.

[20] Esri. FuzzyMSLarge documentation. retrieved 2020. URL https://pro.arcgis.com/en/pro-app/arcpy/spatial-analyst/fuzzymslarge-class.htm.

[21] Esri. Train Deep Learning Model. retrieved 2020. URL https://pro.arcgis.com/en/pro-app/tool-reference/image-analyst/train-deep-learning-model.htm.

[22] L. Eysn, M. Hollaus, E. Lindberg, F. Berger, J. M. Monnet, M. Dalponte, M. Kobal, M. Pellegrini, E. Lingua, D. Mongus, and N. Pfeifer. A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the Alpine Space. *Forests*, 2015. doi:10.3390/f6051721.

[23] Facebook. PyTorch. retrieved 2020. URL https://pytorch.org/.

[24] F. E. Fassnacht, H. Latifi, K. Stereńczak, A. Modzelewska, M. Lefsky, L. T. Waser, C. Straub, and A. Ghosh. Review of studies on tree species classification from remotely sensed data. *Remote Sensing of Environment*, 2016. doi:10.1016/j.rse.2016.08.013.

[25] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

[26] D. Haboudane, J. R. Miller, E. Pattey, P. J. Zarco-Tejada, and I. B. Strachan. Hyperspectral vegetation indices and novel algorithms for predicting green LAI of crop canopies: Modeling and validation in the context of precision agriculture. *Remote Sensing of Environment*, 2004. doi:10.1016/j.rse.2003.12.013.

[27] L. Halounová. ZPRACOVÁNÍ OBRAZOVÝCH DAT. *České vysoké učení technické v Praze*, 2009. ISBN 978-80-01-04253-3.

[28] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. 2017. doi:10.1109/ICCV.2017.322.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi:10.1109/CVPR.2016.90.

[30] J. Howard, R. Thomas, and S. Gugger. fast.ai. retrieved 2020. URL https://www.fast.ai/.

[31] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 1959. doi:10.1113/jphysiol.1959.sp006308.

[32] Institut plánování a rozvoje hlavního města Prahy. Digitální technická mapa Prahy - technické využití území. (Czech). [Digital technical map of Prague - technical use of the territory]. *DIGITÁLNÍ TECHNICKÁ MAPA PRAHY*, april 2020. URL https://app.iprpraha.cz/apl/app/dtmp.

[33] IPR. Prague Institute of Planning and Development. *IPR*, 2020. URL http://en.iprpraha.cz/.

[34] L. Jing, B. Hu, T. Noland, and J. Li. An individual tree crown delineation method based on multi-scale segmentation of imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2012. doi:10.1016/j.isprsjprs.2012.04.003.

[35] H. Kaartinen, J. Hyyppä, X. Yu, M. Vastaranta, H. Hyyppä, A. Kukko, M. Holopainen, C. Heipke, M. Hirschmugl, F. Morsdorf, E. Næsset, J. Pitkänen, S. Popescu, S. Solberg, B. M. Wolf, and J. C. Wu. An international comparison of individual tree detection and extraction using airborne laser scanning. *Remote Sensing*, 2012. doi:10.3390/rs4040950.

[36] Keras. Keras. retrieved 2020. URL https://keras.io/.

[37] LESY HL. M. PRAHY. JE U NÁS VÍC LISTNATÝCH, NEBO JEHLIČNATÝCH STROMŮ? *LESY HL. M. PRAHY*, 2016, retrieved 2020. URL https://www.lhmp.cz/lesy/2016/04/letos-na-jare-v-praze-vysadime-vice-nez-180-tisic-novych-stromu/.

[38] W. Li, H. Fu, L. Yu, and A. Cracknell. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, 2016. doi:10.3390/rs9010022.

[39] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi:10.1109/CVPR.2015.7298965.

[40] Mads00. Visual cortex. *CC BY-SA 4.0*, retrieved 2020. URL https://commons.wikimedia.org/w/index.php?curid=49282011.

[41] MIT. 10 Breakthrough Technologies. *MIT Technology Review*, 2013. URL technologyreview.com.

[42] V. Nair and G. E. Hinton. Rectified linear units improve Restricted Boltzmann machines. 2010. ISBN 9781605589077.

[43] O. Pešek. *MASK R-CNN V PROSTŘEDÍ GRASS GIS*. České vysoké učení technické v Praze, 2018. URL https://github.com/ctu-geoforall-lab-projects/dp-pesek-2018.

[44] PEFC. JE U NÁS VÍC LISTNATÝCH, NEBO JEHLIČNATÝCH STROMŮ? *ARNIKA*, 2017, retrieved 2020. URL http://letemlesem.cz/zajimavosti/je-u-nas-vic-listnatych-nebo-jehlicnatych-stromu/.

[45] Å. Persson, J. Holmgren, and U. Söderman. Detecting and measuring individual trees using an airborne laser scanner. *Photogrammetric Engineering and Remote Sensing*, 2002. ISSN 00991112.

[46] S. C. Popescu, R. H. Wynne, and R. F. Nelson. Measuring individual tree crown diameter with lidar and assessing its influence on estimating forest volume and biomass. *Canadian Journal of Remote Sensing*, 2003. doi:10.5589/m03-027.

[47] D. A. Pouliot, D. J. King, F. W. Bell, and D. G. Pitt. Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration. *Remote Sensing of Environment*, 2002. doi:10.1016/S0034-4257(02)00050-0.

[48] Python Software Foundation. Python. retrieved 2020. URL https://www.python.org/.

[49] Rina Dechter. Learning While Searching In Constraint-Satisfaction-Problems. *Annals of Mathematics*, 1986.

[50] J. Rouse, R. Haas, J. Schell, and D. Deeering. Monitoring vegetation systems in the Great Plains with ERTS (Earth Resources Technology Satellite). *Third Earth Resources Technology Satellite-1 Symposium*, 1973.

[51] J. Schmidhuber. Deep Learning in neural networks: An overview. *Neural Networks*, 2015. doi:10.1016/j.neunet.2014.09.003.

[52] S. Solberg, E. Naesset, and O. M. Bollandsas. Single tree segmentation using airborne laser scanner data in a structurally heterogeneous spruce forest. 2006. doi:10.14358/PERS.72.12.1369.

[53] TensorFlow. TernorFlow. retrieved 2020. URL https://www.tensorflow.org/.

[54] Trimble. eCognition. *Trimble*, 2020. URL https://geospatial.trimble.com/products-and-solutions/ecognition.

[55] A. M. Turing. COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, 1950. doi:https://doi.org/10.1093/mind/LIX.236.433.

[56] United Nations. Transforming Our World: The 2030 Agenda for Sustainable Development. *United Nations*, 2018. doi:10.1891/9780826190123.ap02.

[57] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios. Building detection in very high resolution multispectral data with deep learning features. 2015. doi:10.1109/IGARSS.2015.7326158.

[58] J. Vauhkonen, L. Ene, S. Gupta, J. Heinzel, J. Holmgren, J. Pitkänen, S. Solberg, Y. Wang, H. Weinacker, K. M. Hauglin, V. Lien, P. Packalén, T. Gobakken, B. Koch, E. Næsset, T. Tokola, and M. Maltamo. Comparative testing of single-tree detection algorithms under different types of forest. *Forestry*, 2012. doi:10.1093/forestry/cpr051.

[59] W. vdS, J. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. Warner, N. Yager, E. Gouillart, and T. Yu. scikit-image: image processing in Python. *the scikit-image contributors*, 2014, retrieved 2020. doi:https://doi.org/10.7717/peerj.453.

[60] M. Voss and R. Sugumaran. Seasonal effect on tree species classification in an urban environment using hyperspectral data, LiDAR, and an object-oriented approach. *Sensors*, 2008. doi:10.3390/s8053020.

[61] F. Warmerdam, A. Kiselev, M. Welles, and D. Kelly. LibTIFF - TIFF Library and Utilities). retrieved 2020. URL `http://www.libtiff.org/`.

[62] B. M. Wolf and C. Heipke. Automatic extraction and delineation of single trees from remote sensing data. *Machine Vision and Applications*, 2007. doi:10.1007/s00138-006-0064-9.

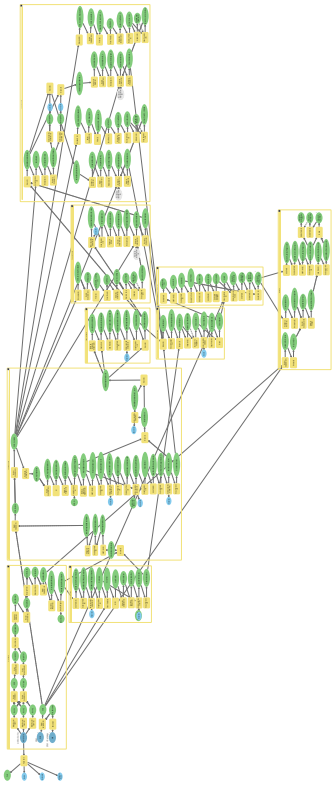
[63] J.-C. Wu. Two filters for dem extraction. *American Society for Photogrammetry and Remote Sensing*, 2005.

[64] X. Zhang. Simple Understanding of Mask RCNN. *medium.com*, 2018. URL `https://medium.com/@alittlepain833/simple-understanding-of-mask-rcnn-134b5b330e95`.

[65] X. X. Zhu, D. Tuia, L. Mou, G. S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. 2017. doi:10.1109/MGRS.2017.2762307.

Appendix A

GIS Model Scheme



Appendix B

GIS Model

The GIS model toolbox, as well as, the python code are stored on attached USB flash drive.

GIS Model Toolbox:

- *GIS Model Toolbox: GIS_ Model_ Toolbox\Detekce.apx*
- *GIS Model Script: GIS_ Model_ Toolbox_ code\Detekce.py*

Appendix C

Deep Learning Environment Setup

The following lines of conda comands are necessary to be able to work in deep learning environment. Following third-party dependencies are compatible with ArcGIS pro 2.5.

- *conda create –prefix %PATH%\deeplearning –clone arcgispro-py3*
- *activate %PATH%\deeplearning*
- *conda install tensorflow-gpu=1.14.0*
- *conda install keras-gpu=2.2.4*
- *conda install scikit-image=0.15.0*
- *conda install Pillow=6.1.0*
- *conda install fastai=1.0.54*
- *conda install pytorch=1.1.0*
- *conda install libtiff=4.0.10 –no-deps*
- *proswap deeplearning*

Appendix D

Mask R-CNN Model

The trained model is stored on attached USB flash drive in model folder: Mask_RCNN.

Files:

- *__pycache__\ArcGISInstanceDetector.cpython-36.pyc* // Compiled python script
- *ArcGISInstanceDetector.py* // Python raster function
- *Trained.DLPK* // Trained Model
- *Trained.EMD* // Esri Model Definiton file
- *Trained.PTH* // Path to the trained model