universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

DBH PREDICTION AT THE INDIVIDUAL TREE LEVEL: ANALYSING RASTERIZED SCENES FROM ALS POINT CLOUDS USING CONVOLUTIONAL NEURAL NETWORKS

verfasst von | submitted by

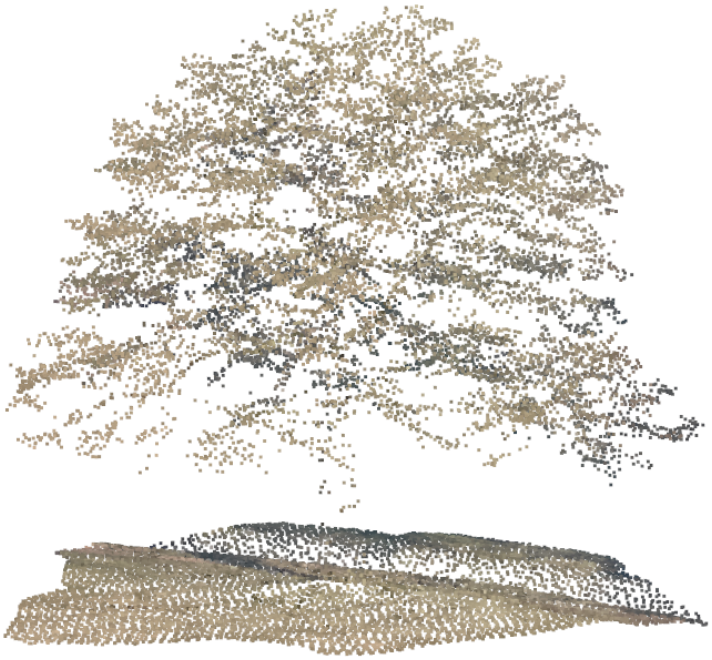
Pit Back BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 856
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Kartographie und Geoinformation
Betreut von | Supervisor:	Univ.-Prof. Dr. Krzysztof Janowicz



Abstract

The use of convolutional neural networks is becoming increasingly popular in forestry. Nevertheless, diameter at breast height (DBH) determination is a challenge, as it cannot be directly obtained from any remote sensing technology on a large scale. The point clouds in a dataset derived from airborne laser scanning contain structural data about trees, which can only be simplified for DBH estimation using statistical predictive approaches. Moreover, there is a lack of research in estimating DBH and other parameters on trees in open landscapes, where trees play an important role as habitats and corridors for endemic wildlife and fauna. This thesis proposes a framework for estimating the DBH of trees in the open landscape by projecting their structure as it is represented in an ALS derived point cloud into a 2D raster file before training a convolutional neural network (ResNet50) on these images to create a model for estimating individual tree DBH. The heterogeneity of trees in the open landscape in terms of morphology, age and species and the use of 2D raster files as training data is separating this work from methodologically similar publications in forestry applications, where high accuracies are achievable due to their use in a more homogeneous environment. The lack of comparable publications is accounted for through the adaptation of a multiple linear regression analysis inspired by publications in the forestry domain in order to create a frame of reference. The results show that overall, the CNN model outperformed the regression analysis by only three centimetres or 8.8% in accuracy, demonstrating its ability as an alternative tool for estimating DBH. In a certain DBH range, the CNN method, an RSME of 17.7cm was achieved, while the MLR model produced a maximal RSME of 19.2cm. However, the accuracy of both the CNN model and the MLR model was dependent on solid training data. The CNN outperformed the regression in every range where solid training data was available. The accuracy of the ranges where less than 15 samples were available suffered significantly and the regression outperformed the CNN method. The relatively low point density of the data used in this study negatively influenced the estimation accuracy while simultaneously proving that it is still possible to achieve good results with very poorly represented objects. Tree species estimation remains a challenge yet to be tested with this method. The work relies on

manual tree crown delineation, which must be automated for use on larger scales. The conclusion presents a list of improvements that can be explored by future researchers in order to build upon this research.

Kurzfassung

Der Einsatz von neuronalen Netzen im Bereich der Forstwirtschaft entwickelt sich schnell und hat bereits einige vielversprechende Publikationen hervorgerufen. Dennoch ist die Erfassung des Brusthöhendurchmessers (BHD) bei einzelnen Bäumen eine Herausforderung, da dieser Wert mit keiner luftgestützen Fernerkundungstechnologie direkt und in großem Maßstäben ermittelt werden kann. Die Punktwolken in einem Datensatz, der aus luftgestütztem Laserscanning stammt, enthalten strukturelle Information über Baumkronen, die für eine Schätzung des BHD mit der Standardmethode, einer multiplen linearen Regressionsanalyse nur vereinfacht verwendet werden können. Demnach iest es nur möglich, die Höhe, den Kronendurchmesser,die Kronenlänge und ähnliche abgeleitete Werte als Indikatorwerte für die Schätzung des Brusthöhendurchbessers mit Regressionsrechnungen heranzunehmen. Darüber hinaus gibt es wenig Forschung zur Schätzung des BHD und anderer Parameter von frei stehenden Bäumen im Offenland, wo Bäume eine wichtige Rolle als Lebensraum und Korridor für endemische Tiere und Pflanzen spielen. Diese Arbeit erforscht einen neuen Ansatz zur Schätzung des BHD von Bäumen in der offenen Landschaft, bei der die Struktur, sowie Form und Größe, von Einzelbäumen aus luftgestützten LiDAR Punktwolken, in eine 2D-Rasterdatei projiziert wird, bevor ein faltendes neuronales Netzwerk (ResNet50) mit diesen Bildern trainiert wird. Ziel ist es, ein Modell zur Schätzung des Brusthöhendurchmessers einzelner Bäume zu erstellen. Diese Methode erreicht eine Genauigkeit von 17.7 Zentimetern für einen bestimmten Durchmesserbereich, während die Standardmethode eine Genauigkeit von 19.9 Zentimetern in diesem Bereich erreicht. Die durchschnittliche Genauigkeit ist ebenfalls um drei Zentimeter oder 8.8 % höher. Das Regressionsmodell liefert lediglich in jenen Bereichen bessere Ergebnisse, in denen nur wenig Trainingsdaten zur Verfügung standen. Die neue Methode erlaubt Verbessungsansätze, die bei der Regressionsanalyse nur sehr schwehr bis garnicht durchgestzt werden können, dies beispielsweise durch die Erkennung von natürlichen Ausreißern beim Verhältnis Höhe - Brusthöhendurchmesser bei besonders alten und untypischen Bäumen. Eine Weiterentwicklung der Methode mit Berücksichtigung der Fehler und Erfahrungen könnte einen signifikanten Beitrag zu diesem Forschungsbereich leisten.

Acknowledgements

I want to thank Corinne Leytem and Thierry Palgen for their direct and open communication as well as for their efforts in providing essential data and important ideas for this project. Further, I want to thank Ass.Prof. Alexander Bucksch from the University of Arizona for taking the time to consider this study. His ideas contributed significantly to the presented workflow by offering insights on how to predict DBH at a large scale. Lastly, I´d like to thank Univ.-Prof. Dr. Krzysztof Janowicz who supervised the project and guided me through the entire process of creating a masterthesis, while offering great feedback.

Contents

Abstract II

German Abstract IV

Acknowledgements V

List of Tables VIII

List of Figures X

Nomenclature XI

1 Introduction 1

1.1 Problems . . . 3
1.2 State of Research . . . 4
1.3 Objectives . . . 8
1.4 Research Questions . . . 9
1.5 Structure . . . 10

2 Dataset 12

2.1 Open Government Data . . . 12
2.1.1 LiDAR 2019 Dataset . . . 12
2.1.2 LiDAR Trees Dataset . . . 14
2.2 Government Data . . . 15
2.3 Field Data . . . 16
2.4 Combined Dataset . . . 19
2.4.1 DBH and Height . . . 19
2.4.2 DBH and Crown Size . . . 20
2.4.3 Height and Crown Size . . . 21
2.4.4 Species . . . 22
2.4.5 Statistical Summary . . . 24

3 ResNet50 DBH Estimation **25**

3.1 Introduction to Residual Neural Networks and ResNet50 25

3.2 Training Data 26

3.2.1 Creating the mask layer 26

3.2.2 Preparing and Indexing the LiDAR Dataset 27

3.2.3 Vertical Tree Profile Extraction 27

3.3 Training the CNN 31

3.4 Hardware, Software and Performance 35

4 Multiple Linear Regression **37**

4.1 Introduction to Regression Analysis 37

4.2 Model Creation 38

4.3 MLR Model 39

5 Results **40**

5.1 CNN Model 40

5.2 Regression Analysis 44

5.3 Comparison 45

5.4 Temporal Mismatch 46

6 Discussion **47**

7 Conclusion **54**

A Dataset **61**

B Code: Select Zip Files For Download **66**

C Code: Laz File Indexer **67**

D Code: creating vertical profile images **68**

E Code: Training (or fitting) the CNN model **71**

F Code:Multiple Linear Regression **73**

List of Tables

1.1 BK18 Biotope definition [Grand Duchy of Luxembourg, 2018]. 2

2.1 Airborne LiDAR Survey Parameters [Sintégra Géomètres Experts, 2019] . . 12
2.2 LiDAR 2019 open government dataset point cloud classes 13
2.3 BHD and Height Statistics 20
2.4 BHD and crown area statistics 21
2.5 Height and Crown Area Statistics 22
2.6 Number of times a single species is represented 24

3.1 Conda virtual environment 36

4.1 Predictor Values 38
4.2 Statistical summary, training and validation data 38

5.1 Accuracies of both models 45

7.1 Qualitative and quantitative findings 56
7.2 Issues encountered in the study and recommended actions for future research 56

A.1 Combined tree dataset 65

List of Figures

1.1 trees captured with terrestrial LiDAR, with tens of thousands of points representing one tree [Henrich et al., 2023] 5
1.2 individual tree at 15 points per m^2 - data sample of this study with a total of 8620 points 5
1.3 mask polygon for point cloud extraction. background image: [ESRI, 2024] . 10
1.4 vertical profile of the tree shown in Figure 1.3 11

2.1 one single large oak tree, aerial image [ESRI, 2024] 14
2.2 LiDAR Trees coverage shows imperfect crown delineation. Background: [ESRI, 2024] 14
2.3 Trees in the remarkable trees dataset. Plot of height (X) and DBH (Y). Note the increasing variance with increasing diameter. 15
2.4 Distribution of remarkable tree samples in Luxembourg, proving that the dataset contains trees from all across the country. The high value in the lower middle portion intersects with the capital city, where many city parcs host a number of remarkable trees. Background:ESRI 16
2.5 Typical pastureland with pyrus trees in open landscape. The depicted trees are part of the self acquired field dataset. 17
2.6 Forestry caliper very similar to the one used in the field data collection to measure tree DBH [HAGLÖF SWEDEN, 2024] 18
2.7 DBH and height plot representing the field data 18
2.8 Plot DBH and Height 20
2.9 Plot DBH and Crown Size 21
2.10 Height and Crown Size 22

3.1 File structure of the training data with the DBH in the filename seperated by the sample number with an underscore, to couple training data value and associated picture to prevent mixups 29

3.2 9 examples of trees extracted from the point cloud. The variation of size and species throughout the dataset is clearly visible. 29
3.3 Examples of badly extracted trees, they are removed for various reasons. This proves that a visual overlook of the training data is necessary. 30
3.4 Loss plot indicating a steep learning curve with the loss function flattening out before the val loss linearly approaches the level of the loss function, showing successful training. 31

5.1 Test split evaluation 41
5.2 Tree extractions with the camera rotated 90 degrees, right and left is always the same tree, but from another angle. This creates a solid validation dataset. 42
5.3 Distribution of dataset samples according to DBH in groups of 25cm 43
5.4 RSME distribution in 25cm groups 43
5.5 Scatterplot estimated vs measured DBH 44
5.6 RSME distribution in 25cm groups 44
5.7 Scatterplot estimated vs measured DBH 45

6.1 A tree wich a large DBH but a small total heigt, that proves why the variance in a model of height, crown size and DBH can be very high. While these are hard to predict statistically, a CNN can be trained on their specific apperance in ALS data. 49
6.2 A very small tree, proving that higher resolutions are beneficial for smaller objects. 51

Nomenclature

AI Artificial Intelligence

ALS Airborne Laser Scanning

ANN Artificial Neural Network

API application programing interface

CNN Convolutional Neural Network

CPU Central Processing Unit

CUDA Compute Unified Device Architecture

DBH Diameter at Breast Height

GPU Graphic Processing Unit

LiDAR Light Detection and Ranging

MLR Multiple Linear Regression

MSE Mean Square Error

OGD Open Government Data

RGB Red, Green and Blue

RMSE Root Mean Square Error

TLS Terrestrial Laser Scanning

UAV Unmanned Aerial Vehicle

WMS Web Map Service

Chapter 1

Introduction

The tree has played a central role in human evolution [Ritter and Dauksta, 2013]. Trees provided shelter and food for the earliest human beings, later civilisations learned to use its wood for tools, building materials and as an energy source, not to mention their spiritual values [Ritter and Dauksta, 2013, Thompson et al., 2021]. Some tree species even possess medicinal properties [Khan and Khan, 2017]. In our modern world, the use of trees has branched out into many different industries and markets [Ritter and Dauksta, 2013]. This is partly due to human societies growing more complex, but also due to a changing climate on earth [Lee et al., 2023]. Trees, and on a larger scale, forests, have become important tools to bind and store carbon from the atmosphere [Drever, 2022] as a means to counterbalance CO_2 emissions. Researchers identified free standing trees as important habitats for certain animals [Parmain and Bouget, 2018], they also play a role as landscape features and provide shade for livestock [Hupke, 2020], even their aesthetical value is significant [Santoro et al., 2022, Hupke, 2020], which lead institutions, governments and organisations to take measures to regulate the use of this resource. In Europe, one of the most influential measures on nature protection is the Habitats Directive (Council Directive 92/42/EEC), by the European Commission, which was adopted in 1992. The directive requires all member states to take act in protecting and restoring habitats and wildlife within the Union [European Commission, 2024].The directive's main measures to achieve that is by presenting a list of ecosystems as well as a list of animal and plant species that need to be protected and preserved. EU member states are required to develop their own strategies to protect said habitats and species [European Commission, 2024]. In an effort to map and quantify certain habitats, the government of the country of Luxembourg created a regulation in 2018 which contains a list of different biotopes [Grand Duchy of Luxembourg, 2018]. The work is based on one of these biotopes, titled *Solitary trees, groups of trees, and rows of trees* and codenamed

BK18 [Grand Duchy of Luxembourg, 2018]. The exact definition of the biotope can be found in *Annex 1* [Grand Duchy of Luxembourg, 2018] of this regulation.

original definition	english translation
Arbres isolés ou structures végétales composées de plusieurs arbres d'essences essentiellement indigènes, qui sont remarquables par leur diamètre qui est supérieur à 30 centimètres à 1,30 mètre au-dessus du sol ou par leur fonction de structure paysagère, de corridor écologique ou d'habitat d'espèces d'animaux. Les groupes d'arbres sont formés par au moins 2 arbres, dont les couronnes se touchent ou dont les troncs sont éloignés de 10 mètres au maximum. Les rangées d'arbres sont formées par au moins 3 arbres dont les troncs sont espacés de 30 mètres au maximum.	Isolated trees or plant structures made up of several trees of essentially indigenous species, which are remarkable for their diameter, which is greater than 30 centimetres at 1.30 metres above the ground, or for their function as a landscape structure, ecological corridor or habitat for animal species. Groups of trees are formed by at least 2 trees whose crowns touch or whose trunks are no more than 10 metres apart. Rows of trees are formed by at least 3 trees whose trunks are no more than 30 metres apart.

Table 1.1: BK18 Biotope definition [Grand Duchy of Luxembourg, 2018].

The definition, which has been specifically adapted for easier use in GIS software, requires tree diameter at breast height (DBH) to be estimated on a national scale for every tree not being part of a forest. With the country being fully covered by airborne laser scanning (ALS) data available publicly for free, and the administration of nature and forests being open for collaboration, the requirement for DBH estimations in a real world application provides an inviting framework to explore different methods of estimating DBH from remote sensing data on a large scale. In this context, the study explores two different alternative ways to estimate the diameter of individual trees by comparing them against each other and against existing literature. The experimental approach integrates itself into existing literature from the field of remote sensing in forestry and compares a multiple linear regression (MLR) analysis commonly used in statistical applications to a hi-tech artificial intelligence (AI) model created using a special kind of deep learning architecture called a convolutional neural network (CNN). The reason for using such an advanced deep learning architecture is that the complex structures which can be found in individual tree crowns could be used as an indicator to the diameter of the tree. Similar deep learning processes have been tested successfully in closely related applications [Zhang et al., 2023].

After the introduction to the topic and the research field, the *problems* section introduces challenges and limitations that form a requirement for an investigation of the presented subject. This is followed by a look at related research that is published in that

field to set the work into context with said publications. The following section will present a short resume of the structure of the thesis as well as a simplified description of the methods employed.

1.1 Problems

A central part of the definition listed in Table 1.1 is the requirement for a DBH of 30 centimetres for the tree to be accepted into this definition. The estimation of individual tree parameters, such as DBH, is still considered to be challenging [Zhao et al., 2023]. Most research about estimating tree parameters from remote sensing is based around forestry applications [Zhang et al., 2023, Xu et al., 2021, Chen et al., 2020, Leite et al., 2020], as it can be an essential tool for accurately managing and quantifying large forest stands with comparably small effort [Zhao et al., 2023]. This technology is already of great value for the timber industry [Xu et al., 2021]. While there is some research on the estimation of individual tree parameters for ecological and nature protection applications [Mao et al., 2023, Chen et al., 2020, Wang et al., 2021, Li et al., 2023], work on freestanding or solitary trees in open environments requires further investigation in order to better understand how to map and protect these ecolocially important [Hupke, 2020] habitats. Due to the technical nature of ALS, where a light detection and ranging (LiDAR) sensor mounted on an aircraft emits a light pulse hundreds or thousands of times a second and records the return time from the bouce back of the light pulse, indifferently from what structure or biotope is being captured by the sensor [Maltamo et al., 2014], the precision and the workflow does not differ between forests and solitary trees, ensuring consistent and comparable data quality between forest trees and open land trees. However, trees in open areas feature different morphology compared to their counterparts within forests [Hupke, 2020]. The availability of light and space to grow in the vertical dimension allows trees to develop more overhanging branches, allowing their crowns to expand and spread out much further [Hupke, 2020]. These variations in tree morphology together with the variations in age and species create a less ideal environment for predicting parameters such as DBH on individual trees using methods based on machine learning or regression. A further challenge of applying the biotope definition of Table 1.1 to the entire country of Luxembourg, which has an area of roughly 2500 square kilometres [World Bank Group, 2024], is the scalability of the method to extract and predict individual tree parameters of such a large area. The requirements of computing power, data storage and processing time for the applied methods will be analysed and evaluated during this research project.

1.2 State of Research

Obtaining individual tree parameters from ALS dataset is still a very active research field and different methods have been proposed [Henrich et al., 2023, Mao et al., 2023, Chen et al., 2020, Wang et al., 2021, Li et al., 2023, Bucksch et al., 2013a, Bucksch et al., 2010, Natesan et al., 2019, Ayrey and Hayes, 2018]. While there are a few companies that can offer a solution for a fully integrated workflow from the data acquisition to the identification of individual trees including height and DBH measurement [GreenValley International, 2024, Katam Technologies AB, 2024], these are mostly limited to plot-level applications, as the data acquisition is mostly done using a combination of terrestrial LiDAR using backpacks and airborne LiDAR with small UAVs. On very large scales however, remotely obtaining 3-dimensional data about forest structure is primarily achieved through airborne applications using small planes [Coops et al., 2022], as this technology can cover large areas while offering a vertical and horizontal precision within the 100mm range
[Administration du cadastre et de la topographie, 2019], while satellite observations are much less precise offering a lower ground sampling distance [Zhang et al., 2023]. The point density of the dataset is 15 points per square metre

[Administration du cadastre et de la topographie, 2019]. The relatively low point density, compared to using terrestrial or low altitude UAV applications, results in a much poorer representation of the three dimensional structure of individual trees. The point density may be the most important challenge and distinction of the study compared to different applications of these methods at a plot level. A standard point density for plot level work seems to be based around 50 points per square metre [Bucksch et al., 2013b]. However, the number may vary significantly depending on specific requirements, scale or the costs involved.

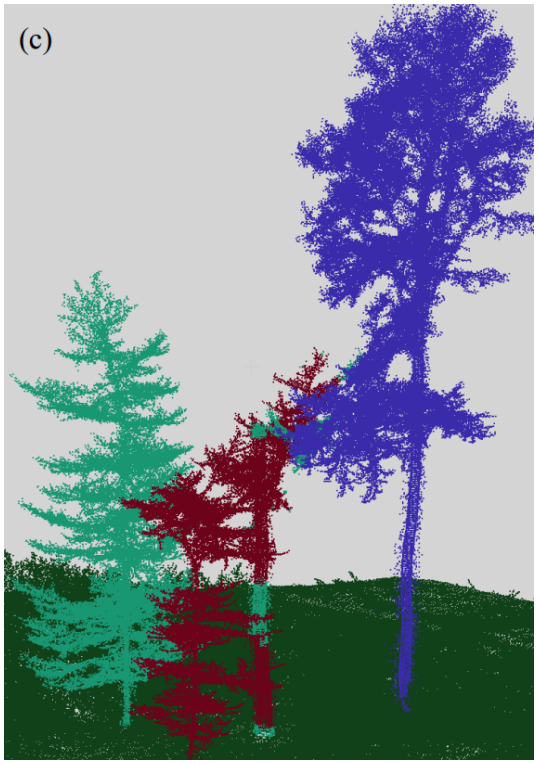


Figure 1.1: trees captured with terrestrial LiDAR, with tens of thousands of points representing one tree [Henrich et al., 2023]

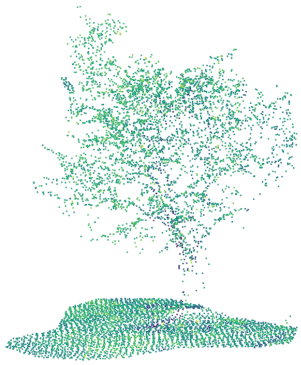


Figure 1.2: individual tree at 15 points per m^2 - data sample of this study with a total of 8620 points

The difference in the number of points on the tree trunk in Figures 1.1 (top) and 1.2 is very obvious. The lower scan angle of the terrestrial laser scanning (TLS) system is much more effective at capturing the forest structure beneath the canopy, offering exact representations of individual tree trunks and branches. The technology basically offers a very accurate three dimensional model of a forest section. Moreover, the use of vehicles in forest environments is challenging and discouraged due to technical and

environmental reasons, that means LiDAR scans from beneath the canopy are almost always captured using a scanning backpack solution[Xu et al., 2021]. This severely impacts the range limitations and time requirements for ground or vehicle based data acquisition. Some companies experiment with lightweight UAVs that feature obstacle avoidance functions[Katam Technologies AB, 2024], but these are also limited in range due to battery capacity and low flight speeds. DBH measurements can effectively be extracted directly from the point cloud. The tree shown by Figure 1.2 is represented much more sparsely. Due to the high vertical and horizontal precision of the individual points, the crown shape is very accurately represented, as well as the upper crown structure. However, due to the relatively high angle of incidence from the airborne LiDAR platform (in this study 60 degrees [Sintégra Géomètres Experts, 2019]), the trunk and lower portion of the tree is very poorly captured. This is also caused by the low number of points per square metre. Nevertheless, the detailed information about the crown structure provides valuable insights about the tree and may be sufficient for a CNN to estimate the DBH with acceptable accuracy. Working with ALS data in combination with single tree extraction and a CNN based regression model would be superior in terms of scalability, cost and practicability, compared to obtaining direct measurements from TLS data. Applications based on artificial intelligence recently gained a lot of popularity in society due to the technology becoming increasingly available to a wider public. This is partly caused by the rise and the availability of deep learning and artificial neural network (ANN) based services like Microsofts Copilot and OpenAIs large language model *ChatGPT*, which sparked a lot of discussion and interest in the topic. However, even before that, the technology was becoming increasingly popular among scientists. ANNs were showing better performance and accuracy than the until then standardised MLR Approaches [Liu et al., 2021]. A much more detailed overview of why the MLR approach was chosen in this study is offered in the *Introduction to MLR* section on page 34: 4.1. Soon, the potential of deep learning in automatic learning structures on raw input data was discovered and CNNs provided the basis for using deep learning in image based applications [Liu et al., 2021]. The ease of access to CNNs through open source application programming interfaces (API) like Googles Tensorflow and Keras allowed more people to experiment with these technologies, that are increasingly used in forestry applications [Zhang et al., 2023]. [Zhang et al., 2023] provided a systematic overview of recent uses of CNNs for tree crown detection in 2023. The authors highlight the use of computational neural networks in combination with airborne and spaceborne red, green and blue (RGB) raster data to be the main methodology in these applications, however they continue to point out the potential of LiDAR data in such research and state that "*Future studies could (1) explore data fusion approaches to take ad-*

vantage of the characteristics of different types of remote sensing data"[Zhang et al., 2023], later naming LiDAR and multispectral data. The use of LiDAR data in CNN applications is limited. These datasets are most often reduced to 2D top-down rasterized views of trees or converted to 2D density maps.

While there are studies presenting using CNNs for predicting DBH, like [Ayrey and Hayes, 2018] which is described in the last paragraph, and some publications using CNNs and even ResNet architecture to estimate different forestry parameters such as species and DBH [Natesan et al., 2019, Mao et al., 2023], **not one paper could be found describing any form of 2-dimensional extraction of the vertical crown structure present in an ALS dataset as training data for a CNN regression analysis with DBH prediction.**

The previously mentioned study by [Mao et al., 2023] discussed the use of LiDAR and multispectral image data to determine DBH and other metrics. However, they extracted individual tree metrics from their LiDAR and their multispectral data. They then estimated DBH indirectly through random forest and MLR, not directly from the 3D point cloud structure. The aim was to compare data originating from 2D raster to LiDAR data. [Kattenborn et al., 2021] explored the possibilities to segment individual trees from larger point clouds by experimenting with leaf-on, leaf-off point clouds acquired using UAV and airplane based ALS data. The authors concluded by offering a large dataset of individually segmented tree point clouds of individual species. They do not mention DBH or the use of computational neural networks[Kattenborn et al., 2021]. However, their data can be valuable for tree parameter extraction research. The data was not used for this thesis as it only contains segmented trees from within forests, which feature a significantly different morphology to free standing trees [Hupke, 2020]. Another work by [Liu et al., 2021] explores segmenting a point cloud acquired from TLS into individual trees before extracting DBH directly from the point cloud. This approach proves the possibility of extracting DBH directly from 3D structural data from individual trees, and further proves the interest in acquiring DBH, but this method is not feasible on large scales due to the amount of work, cost and data size involved [Liu et al., 2020]. Concludingly, there is interest in finding a reproducible and effective way to accurately determine DBH and other parameters such as tree species on an individual tree level. However, the methods presented in recent scientific research are either not scaleable due to data size and time or labour restrictions, or they are computationally intensive if using CNN, despite the potential of CNNs to be very effective at large data processing. The established way to predict DBH would be to infer it from other measures that can be precisely acquired using remote sensing techniques such as tree height and crown diameter, like the validation models of [Ayrey and Hayes, 2018]

and [Mao et al., 2023] For this reason, the study will use an MLR model to compare it to the CNN approach presented in the CNN estimation chapter.

The publication with the closest relation to this study would be Ayrey and Hayes's work with 3D CNN. They used a ResNet50 model and adapted it to work directly with 3D data, before feeding it with LiDAR data of individual trees and DBH and created an estimation model. The approach was computationally very intensive, proven by the following direct quote: *"We did note, however, that training the more complex models (such as Inception V3 and Resnet-50) could take one or more days using GPU acceleration with an Nvidia Tesla k80"* [Ayrey and Hayes, 2018]. Moreover, the models were not validated on the vast diversity of trees that can be found in the open landscape. However, they concluded by finding a 12 percent increase in accuracy of biomass estimation compared to a MLR model [Ayrey and Hayes, 2018].

1.3 Objectives

The main objective of this study is to present a reproducible, open source and fast method of estimating the DBH of individual trees represented in an ALS point cloud. The standard deviation of DBH in relation to height is 0.4685 *cm*. It proves that the trees in the dataset are highly variable in both height an diameter. Some trees are very low but have a very large trunk where some trees present a very large DBH while being very small. An exact distinction between DBH greater or smaller than 30 *cm*, as stated in Table 1.1 is very challenging. A more realistic approach in this case is a statistical approximation of the presence of trees of certain DBH in an Area of interest. An investigation of the possibilities of estimating individual tree diameters with the goal to create a dataset based on the definition from Table 1.1 is the decisive reason for this thesis. The model will be based on rasterized segments of an open government data (OGD) airborne LiDAR point cloud and will use a regression analysis performed by a CNN at its core to predict DBH of unknown trees. To evaluate the performance and accuracy of this method, the same dataset will be used to create and validate a MLR model whose accuracies will be compared to the CNN model. Linear regression models are a very fundamental and proven approach to estimate or predict parameters of individual objects based on the correlation of two or more dependent variables[Johnson and Wichern, 2007], such as tree height, crown length and DBH [Zhao et al., 2023]. The most important metrics of the study will be the RSME, which is the average error with which individual tree DBH will be estimated. Moreover, the work will explore and present the advantages and challenges of using easily accessible CNNs based on open source applications in this forestry-related ecological application. As

these technologies, being open source, have shown a lot of potential to be used more widely in these and other sectors [Zhao et al., 2023]. The study will inform potential readers interested in working with CNNs and MLR models by offering a detailed description of their limitations, time constraints and processing requirements as well as the challenges and problems that have arisen during the process. The presentation of the requirements for training data in terms of sample size, and precision represent another important objective of the study. Furthermore, The creation of GIS data containing information about the diameter of individual trees to be used by the Government of Luxembourg for defining BK18 biotopes, as mentioned in Table 1.1,constitutes an important outcome. This plays a significant role in achieving the goals set by the Habitats Directive of the European Council. Future readers of the study will know what accuracies the technology may yield in similar applications, how much training data is required and how much is optimal. They will further be able to avoid potential problems and data issues and will find a well structured and documented workflow including the full code around the models to replicate the work done in the study. The study presents a list of problems and limiting factors that can be explored by future researchers in order to design an improved workflow based around the presented CNN approach with the aim to create a more accurate and complete model for determining DBH or different values on trees captured by ALS.

1.4 Research Questions

The research questions below create the framework for the methodology that is presented in the following sections. The discussion will include a reflection on these questions that provides insights to how well they can be answered as well as an overview of possible aeras of investigation for future researchers.

- RQ1: While most research has focused on estimating properties such as DBH of trees within afforested areas, solitary trees (and small groups or rows of trees), as defined in the BK18 biotope definition, have received less attention. Can the DBH of these trees be estimated using ALS point clouds together with state-of-the-art machine learning systems such as CNN? This is important for at least two reasons: (1) by definition, the sample size of such BK18 solitary trees is smaller than within forests, and (2) due to their exposed nature and varying tree species, the properties of these trees may provide less predictive power. For instance, solitary trees damaged by storms may exhibit a misleading height-to-DBH ratio.

- RQ2: Multiple linear regression (MLR) has been extensively used in the literature. How does this method compare to novel approaches such as Convolutional Neural

Networks (CNN) given tree height and crown cover on the one hand and rasterized scenes from ALS point clouds on the other hand? Given the relatively low ground density of ALS point clouds today, does the prediction accuracy of CNN differ within the set of solitary trees, e.g., providing relatively less accuracy for small trees?

- RQ3: Due to the BK18 biotope defintion being applied nationwide, can the CNN-based method be used for the estimation of the DBH at the individual tree level on such large areas?

1.5 Structure

Two methods will be compared to each other. To ensure comparable results, both methods need to be executed on the same dataset. For that, one large dataset of more 250 individual trees with known height and DBH as well as crown diameter will be created. This will be done by combining an existing dataset with field measurements. the ground truth dataset will constitute the basis for the multiple regression analysis and it will act as the training and validation data for the CNN model. For the multiple regression analysis, a technique similar to the work of [Machado et al., 2019] and [Mao et al., 2023] will be employed to ensure a methodology that is consistent with techniques that are already being applied in real world applications. The second method to be evaluated for its accuracy, ease of use, and overall applicability is the use of a CNN to estimate the DBH of each individual tree based on the training dataset. To achieve this, a shapefile will be created containing one closed polygon of every single tree in the dataset as seen in Figure 1.3. More detailed explanations on how exactly these polygons are created was done are in the CNN analysis section.



Figure 1.3: mask polygon for point cloud extraction. background image: [ESRI, 2024]

The polygons within this shapefile layer will then be used as masks to segment individual trees from the ALS point cloud. A two dimensional rasterized vertical profile of every single segment will then be created as an image file of every tree in the dataset. Figure 1.4 represents an example of a cherry tree with a BHD of $59cm$ an a height of 8.4 metres.

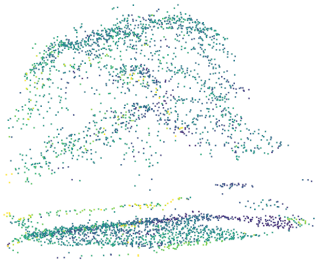


Figure 1.4: vertical profile of the tree shown in Figure 1.3

In the next step, individual *images* will then be used to train a CNN in order to train a model to predict DBH accurately. Another part of the dataset will be held back from the training section to be used as validation data. Please note that this is an oversimplified description of the applied method. The detailed process is described in the following chapters. The accuracy of the trained model is evaluated using the part of the dataset held back for validation and on additional data. Measurements for the proportion of correctly identified DBH as well as the mean deviation from the true value will described and analyzed in the Results section. They will be compared to those of the multivariate regression method. Later, the effectiveness, requirements and limitations will be discussed to create qualitative information about each of the methods.

Chapter 2

Dataset

This chapter introduces the dataset created to compare the methodologies used in this work. It consists of a combination of open government data, government data as well as self-acquired field data.

2.1 Open Government Data

2.1.1 LiDAR 2019 Dataset

The LiDAR 2019 dataset provided by the *Administration du cadastre et de la topographie* consists of a point cloud acquired by a Nationwide ALS survey in February of 2019. It is available online as of April 2024. To create the point cloud, 8 individual flights were carried out using a small airplane in February of 2019.

Description	Prameter
Laser Scanner	RIEGL LMS VQ 780i
RGB Camera	PhaseOne iXU-RS-1000 50MM
Altitude	1100 m
Airspeed	61.7 m/s
Scan Angle	60°
Scan Frequency	1000 kHz
Average Point Density	30.6 ppm^2
Point Density after Class.	11.7 ppm^2
Ground Sampling Distance	10 cm
Resolution	11608x87082 px

Table 2.1: Airborne LiDAR Survey Parameters [Sintégra Géomètres Experts, 2019]

The laser scanner used to acquire 3D data was a RIEGL LMS VQ 780i, in combination

with a PaseOne iXU-RS-1000 50mm RGB camera for color information. The altitude of flight was 1100 metres above ground with an airspeed of $61.7 m/s$. With a scan angle of 60° and a scan frequency of $1000 kHz$, an average point density of 30.6 points per m^2 was acheived, that was reduced back to 11.7 points per m^2 after classification, resulting in an average density of 15 points per m^2. The ground sampling distance of the RGB imagery is $10 cm$ with a resolution of 11608x87082 pixels [Sintégra Géomètres Experts, 2019]. The point cloud is split up into individual square .laz tiles of 500 by 500 metres. Packages of up to 9 individual tiles creating a footprint of up to 1500 by 1500 metres are available for download as compressed folders in the .zip file format. The dataset further includes shapefile and geojson files containing georeferenced square polygons that exactly overlap with the 1500 by 1500 metre tile packages, carrying the naming of the .zip files in the database as an attribute [Administration du cadastre et de la topographie, 2019].
The point density of the dataset consists of 15 points per square metre. However, this value may increase locally due to multiple returns of the laser beam. Horizontal accuracy is stated to be around 3 cm and vertical accuracy is around 6 cm
[Administration du cadastre et de la topographie, 2019]. The point cloud is preclassified according to Table 2.2.

Class N	Classification
0	Unclassified
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Buildings
7	Low Points (noise)
9	Water
13	Bridges, Viaducts
15	High Voltage Power Lines

Table 2.2: LiDAR 2019 open government dataset point cloud classes

Lastly, a basic python script has been published along with the dataset to facilitate the download process. The script is programmed to automatically download all of the .zip folders available in the dataset.
It is available on github [Administration du Cadastre et de la Topographie, 2019a].

2.1.2 LiDAR Trees Dataset

In July of 2022, the *Administration du cadastre et de la topographie* published an open government data shapefile layer containing polygons for individual tree crowns. The polygons also contain data about the tree height, crown area and crown diameter, as well as an elevation (Z) value. The description states that due to the statistical approach to identify individual trees, the dataset may not be an exact representation of the true number and shape of trees within the LiDAR 2019 dataset which it is based on, but the height and location of the trees represented in the dataset are very accurate [Administration du cadastre et de la topographie, 2022]. By observing the dataset and comparing it with observations from aerial imagery and the later described field trips, it is found that this dataset is particularly unreliable at depicting large free standing trees. In forests, tree crowns seem to be more reliably identified, when compared with aerial imagery. Small free standing trees are also generally well represented.



Figure 2.1: one single large oak tree, aerial image [ESRI, 2024]

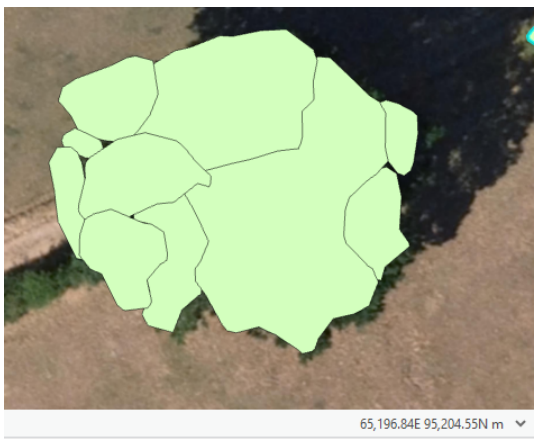


Figure 2.2: LiDAR Trees coverage shows imperfect crown delineation. Background: [ESRI, 2024]

These inconsistencies with the actual situation in the field may be caused by the low density of points within the point cloud paired with the process of identifying individual tree crowns. This processing was done using watershed algorithms with the Terrasolid software.[T. Wantzenrieder, personal communication, 2024].

2.2 Government Data

The *Administration de la nature et des forêts* provided a dataset containing a list of trees that have been mapped for future placement under special protection for being an important part of the natural and or cultural landscape. The dataset consists of 237 *remarkable* trees with species, height, DBH as well as location information in the form of X and Y coordinates. The DBH has been manually measured for these trees [T. Palgen, personal communication, 2024]. This information will be of particular interest for the study as it constitutes the basis for the dataset. References to this dataset will be called by the words *remarkable trees dataset*.

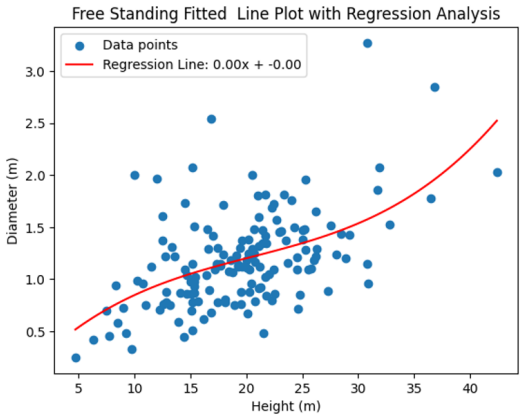


Figure 2.3: Trees in the remarkable trees dataset. Plot of height (X) and DBH (Y). Note the increasing variance with increasing diameter.

Figure 2.3 is created in python using matplotlib and shows the distribution of height and diameter of the free standing trees within the dataset. Figure 2.3 shows that trees with a DBH of 70 cm to 150 cm are most represented in the data. However, as smaller trees play an important role in this research, mainly because of the limiting DBH of 30 cm mentioned in Table 1.1, additional field measurements have been made to improve the dataset in the small (less than $70cm$) and large (greater than $150cm$) DBH ranges. After having separated the free standing trees from the trees that stand within forests in this dataset by intersecting it with a land cover classification from 2018 [Ministry of the Environment, 2018], 188 trees from a total of 237 trees remained within the dataset. After eliminating 2 outliers, which were not trees at all, but hedges made of

a tree species (DBH value of 17 and 12 m), 186 trees remained in the dataset. These trees have an average DBH of 1.27 m. The map pictured in Figure 2.4 shows that the remarkable trees dataset contains trees from all over the country. It proves that a representative distribution is given across the whole nation.

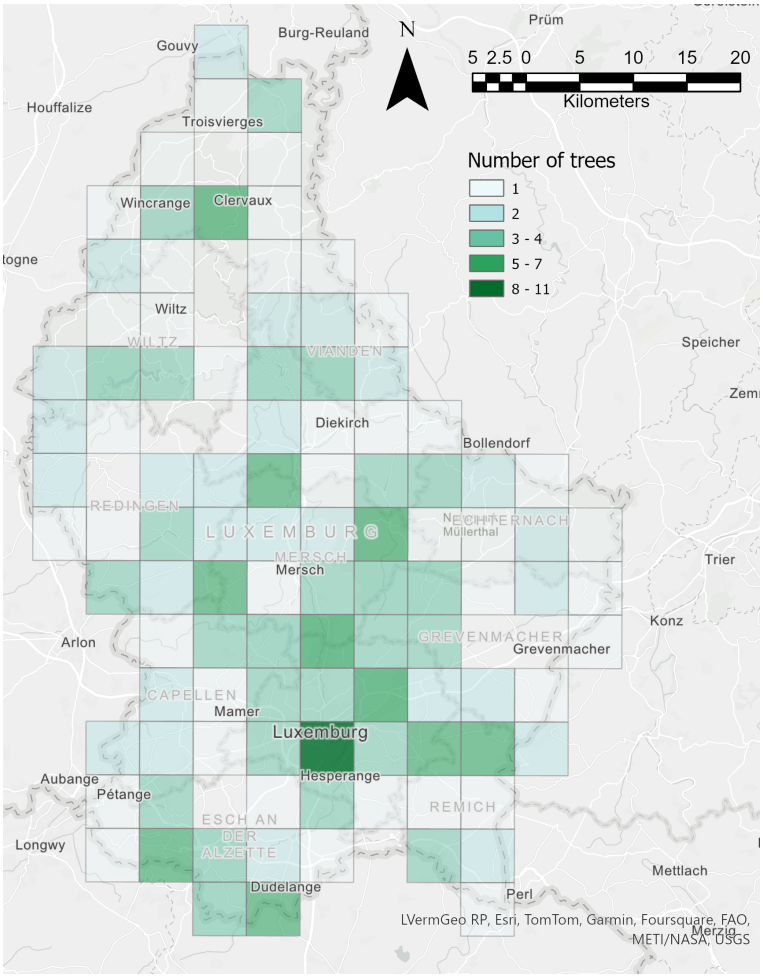


Figure 2.4: Distribution of remarkable tree samples in Luxembourg, proving that the dataset contains trees from all across the country. The high value in the lower middle portion intersects with the capital city, where many city parcs host a number of remarkable trees. Background:ESRI

2.3 Field Data

As mentioned in the previous section, the already existing dataset has been enlarged by field measurements in order to increase the range of DBH measurements represented in the dataset as well as to counterbalance any biases in the remarkable trees dataset, that may have arisen due to the fact that trees considered *remarkable* for their size, position or effect on the natural and cultural landscape may be larger in size or present certain properties different from the average free standing tree in the country. The field measurements have

been taken on four days in late February of 2024. The selection of new dataset trees in the landscape was done at random, but with a preference for including small trees, by roaming around a large area of several square kilometres in the southeast of the country, the area is particularly rich in free standing trees and in pastureland, as can be seen in Figure 2.5 below. Small trees were of particular interest because of the bias for large trees in the remarkable trees dataset.



Figure 2.5: Typical pastureland with pyrus trees in open landscape. The depicted trees are part of the self acquired field dataset.

The data can be representative for the whole country of Luxembourg, as climatological impact on tree growth on such a small territory is negligible [A. Buksch, personal communication, 2023]. The DBH in the field has been measured according to a measurement guide used by the United States Forest Service [United States Forest Service, 2019]. The tools used for this method were a standard measuring tape to determing the height above ground (137 cm) as well as a smartphone (Model: Huawei P20) equipped with a GNSS sensor to determine the position of the tree and to photograph the tree for archival purposes and lastly a forestry caliper provided by the Administration de la nature et des forêts, very similar to the one depicted in Figure 2.6 for reference.



Figure 2.6: Forestry caliper very similar to the one used in the field data collection to measure tree DBH [HAGLÖF SWEDEN, 2024]

The field measurements produced a total of 52 additional trees, with an average diameter of 57 *cm*, which is significantly lower of the average diameter of 127 *cm* that can be found in the remarkable trees dataset. This lower mean DBH extends the range of diameters that are prominently represented in the governmental dataset, especially when considering the fact that the definition in Table 1.1 requires trees of more and less than 30 *cm* of diameter to be distinguished from one another. The collection of data of smaller DBH is therefore important. Figure 2.7 shows that the majority of DBH observations cluster at DBH smaller than 70 *cm*, which fits the data from the remarkable trees dataset.

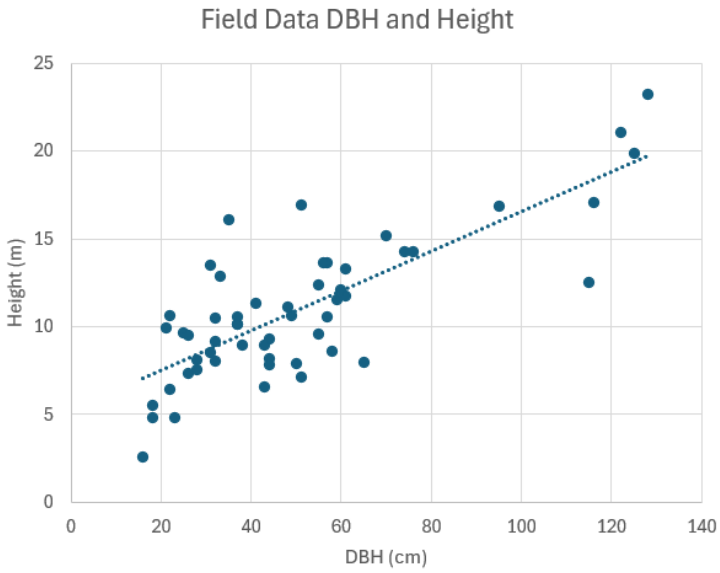


Figure 2.7: DBH and height plot representing the field data

2.4 Combined Dataset

The combined dataset produced from the data offered by the forestry administration and the field observations counts 238 trees with known species, DBH, height and crown area including location information in the shapefile format. 28 percent of the samples in the dataset have been manually collected in the field. Of the remaining 186 samples originating from the remarkable trees dataset, 66 (35.4 percent) trees have been randomly checked for their integrity and represenativeness using Google street view with recent imagery, and 17 trees (9 percent) have been randomly checked and validated in person. These validations ensure a very profound understanding of the represenativeness of the dataset. The data can be found in the form of a table containing a number as well as the attributes for each tree in the Appendix A. Location data and the separation between field observations and government data has been removed for reasons of data protection. The following subsections analyse the relationship of the three tree metrics and the species in this dataset. These calculations and plots have been created in python using the matplotlib, numpy and scipy libraries.

2.4.1 DBH and Height

Figure 2.8 shows a plot of all of the datapoints in the set, with the DBH represented on the X axis and Height on the Y axis. An outlier analysis using a z-score of three creates six outliers. These six outliers represent 2.53 percent of the whole dataset, which indicates an acceptable distribution. The outliers all seem to fit the model, however, 5 of the 6 outliers consist of data points that possess an abnormally high diameter in relation to their height. This is not unusual as large trees can take on such shapes due to environmental reasons, as they can take damage due to severe weather, losing a considerable amount of height in relation to their DBH, which remains unchanged. However, that should not drastically influence the prediction performance of the regression analysis for normally shaped trees.

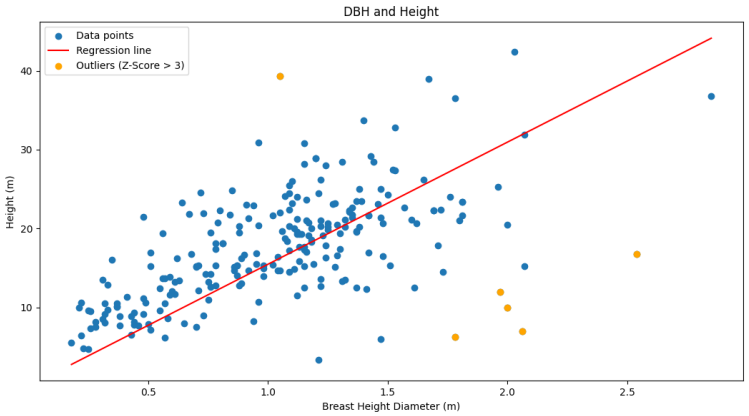


Figure 2.8: Plot DBH and Height

Table 2.3 indicates a standard deviation of 0.47 metres in relation to the tree height. This value is quite high compared to [Gao and Kan, 2022]. They presented standard deviation values of less than 10 centimetres of their DBH using a Structure from Motion method. However, they investigated plots of evenly planted white poplar trees. This high variability of DBH is mainly due to two reasons, our dataset contains many different species, as it should represent and fit all of the species present on the territory of Luxembourg, and trees in open areas tend to be more heterogenous in their appearance due to different reasons such as a higher vulnerability to natural elements such as lightning strikes and wind gusts and a stronger anthropic influence in the form of pruning [Santoro et al., 2022].

Description	DBH (cm)
Mean DBH	1.0385
Standard Deviation for DBH	0.4685
Correlation coefficient for all data	0.5799

Table 2.3: BHD and Height Statistics

The correlation coefficient of all of the trees however indicates a strong correlation of 0.58. The dataset therefore represents a statistically representative positive relationship between the tree height and the tree diameter.

2.4.2 DBH and Crown Size

Figure 2.9 shows a plot of the data points for DBH (m) on the X axis and crown area (m^2) on the Y axis. The crown area or crown size in this case solely consists of the horizontal footprint of a tree. Visually, the plot shows a very strong and tight correlation for smaller

diameters of $<70cm$. For larger diameters, the variation in crown area becomes gradually larger. It can be said that the larger the tree, the less its crown size is predictable. The plot also shows that there is a number of very large trees with very small crown sizes. The strong variation may be caused by the heterogeneity of tree species represented in the dataset, pruning or damage.

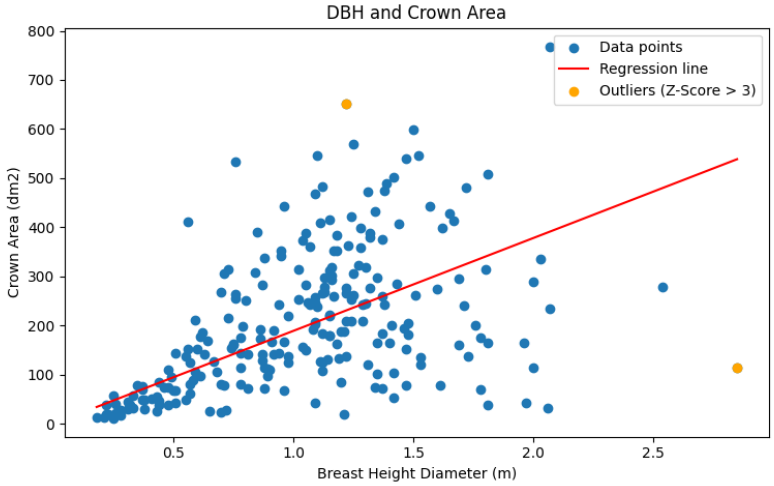


Figure 2.9: Plot DBH and Crown Size

Table 2.4 indicates a correlation coefficient of 0.46 between DBH and crown area. The correlation is therefore slightly weaker then the correlation between DBH and height. This can have numerous natural causes, and it is strongly dependent on tree species. Spruce trees for example may be much higher but thinner in diameter than oak trees. Another limiting factor to the correlation between these two values may be a higher variation in data accuracy for the crown area. As the crown area value is extracted partly from the lidar trees dataset and partly by manually mapping tree crowns on satellite imagery, it may be less accurate.

Description	DBH (cm)
Mean Crown Size	206.2234
Standard Deviation for Crown Size	143.9010
Correlation coefficient for all data	0.4618

Table 2.4: BHD and crown area statistics

2.4.3 Height and Crown Size

Figure 2.10 depicts the relation between the crown size and the height of the 238 trees in the dataset. Similar to Figure 2.9, the points are much denser on an imaginary line from

0x,0y up to 17x,150y, but for heights above 17 m and crown areas larger than 150 dm^2, there is a significantly higher variation in the relationship. However, a consistent increase of crown size with increasing height is visible. The increase is less variable than in Figure 2.9. This graph therefore suggests a stronger correlation between height and crown area. Further, a linear set of points can be spotted on the lower right section of Figure 2.10, indicating a close relationship between height and crown area. The correlation could be caused by a single tree species or genus whose area to height ratio is strongly correlated, but the crown area is generally smaller than most other trees in the dataset.

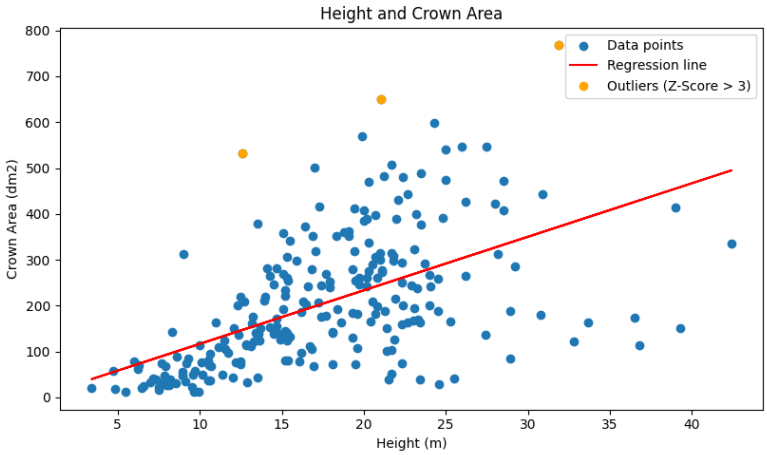


Figure 2.10: Height and Crown Size

Table 2.5 proves that Figure 2.10 depicts a stronger correlation than DBH and crown area. With a value of 0.53, this correlation is significant.

Description	DBH (cm)
Mean Height	17.5440
Standard Deviation for Height	6.897070
Correlation coefficient for all data	0.5328

Table 2.5: Height and Crown Area Statistics

2.4.4 Species

Oak trees (*quercus* genus), being a native species and a typical sight in the open landscape, are the most prominent species in the dataset, followed by the Lime tree (*tilia* genus), which plays an important role in the cultural landscape [Hupke, 2020], they are followed by the horse chestnut species (*aesculus hippocastanum*) and the walnut tree (*junglans regia*). In total, 44 different species of trees are represented in this dataset.

Species	Total Number
Quercus sp.	56
Tilia sp.	48
Aesculus hippocastanum	18
Juglans regia	13
Pyrus sp.	8
Fraxinus excelsior	8
Salix alba	8
Acer campestre	7
Fagus sylvatica	7
Fagus sylvatica 'Purpurea'	7
Prunus avium	6
Acer pseudoplatanus	4
Sorbus domestica	4
Populus sp.	3
Sequoiadendron giganteum	4
Salix sp.	3
Gingko biloba	2
Fagus sylvatica sp.	2
Castanea sativa	2
Salix caprea	2
Pyrus communis	2
Thuja occidentalis	2
Betula	2
Crataegus monogyna	2
Populus nigra L.	1
Sequoia giganteum	1
Fagus sylvatica x pendula	1
Cedrus atlantica	1
Abies pinsapo	1
Ulmus minor	1
Calocedrus decurrens	1
Crataegus sp.	1
Cornus sp.	1
Ginkgo biloba	1

Ulmus glabra 'Pendula'	1
Carpinus betulus	1
Cedrus deodara	1
Aesculus carnea	1
Sequoia sempervirens	1
Populus balsamifera	1
Taxus baccata	1
Platanus acerfolia	1

Table 2.6: Number of times a single species is represented

2.4.5 Statistical Summary

The three variables DBH, crown area and total height are all positively correlated throughout the dataset. The DBH and height of the samples present the strongest correlation with a value of 0.58, DBH and crown area present the weakest with a moderate coefficient of 0.46. Therefore, the dataset can be used for a regression analysis. Due to the variation in tree species and age in the dataset, the DBH predictions will be less accurate than those of similar research done at a forest plot level such as [Bucksch et al., 2013b]. This is acceptable as it represents the most basic and efficient method to estimate DBH from metrics that can be acquired reliably for a very large number of trees based on individual crown segmentation and height measurements.

Chapter 3

ResNet50 DBH Estimation

3.1 Introduction to Residual Neural Networks and ResNet50

The ResNet (Residual neural network) is a very popular model that is mainly being used for image segmentation and classification tasks, but is has already been employed in forestry applications starting back in 2018 [Ayrey and Hayes, 2018] [Natesan et al., 2019]. At its core, the main purpose of ResNet50 is to recognize patterns in visual data an connect them to the associated attributive classes or values. It is called "50" because of its 50-layer depth [He et al., 2016]. The main innovation can also be found in its name, Res = Residual. The vanishing gradient problem has been a challenge in ANN research. Usually, the error signal produced by the learning in the lower layers, where the error is calculated, travels back up through the layers. But sometimes the error signal can be hindered in traveling back up, causing something similar to a traffic jam, which ultimately leads to the error signal vanishing. In this vanishing gradient scenario, the weights of the neurons only change by very small amounts, significantly increasing training time and making it harder and less efficient to recognize complex patterns in the data. ResNet50 introduced something called residual blocks in its architecture, where the error information can skip over certain layers contained in this residual block in order to back propagate this gradient to earlier layers [He et al., 2016]. This allows the error signal to bypass some layer and ensures the flow of information through the network, mitigating the vanishing gradient problem, allowing for deeper networks of more than 100 convolutional layers and better performance at recognizing complex patterns, such as the canopy structure associated to a certain DBH. However, even though the existing literature does not employ the exact strategy that this paper will focus on, they conclude that ResNet architectures can be used for forestry applications. Therefore this residual neural network should be able to

capture the patterns in the data due to its depth. Moreover, the output layer can easily be modified to produce a single output that is needed for a regression application, as shown in the following code snippet.

```
base = ResNet50(input_shape=input_shape, include_top=False)
x = Flatten()(base.output)
#reshapes feature map as dense layer expects flat input
x = Dense(1, activation='linear')(x)
# Single output for DBH estimation
model = Model(inputs=base.inputs, outputs=x)
```

For all of the reasons above, ResNet50 is chosen to be the model to be used in this study, as it is able to demonstrate the ability of CNNs to estimate DBH, partly due to its steep learning curve enabled by the residual layers, allowing solid training on a small dataset.

3.2 Training Data

To produce a model that accurately estimates DBH based on the shape of trees extracted from a LiDAR point cloud, training with point cloud data is necessary due to the different nature of point clouds acquired by different setups. The training data should be representable in the form of two arrays, one containing the DBH, one containing the images of the trees, in the same order.

3.2.1 Creating the mask layer

To create the mask layer, the dataset listed in Appendix A is used. In addition to the attributes listed in the Appendix, X and Y coordinates are available for every data sample (tree). Using these coordinates, the dataset is loaded into ArcGIS Pro 3.2.2 along with a WMS ortophoto layer provided by the topography administration [Administration du Cadastre et de la Topographie, 2019b]. Then, a new polygon layer is made to create individual polygons for every tree in the dataset. The X and Y coordinates in the dataset are used to identify the trees on the aerial image. It is important to use aerial imagery from July and August of 2019 to delineate the individual tree crowns as they are in the same year that the LiDAR data has been captured, as tree crowns can change their shape which could lead to a misrepresentation. The existing LiDAR Trees dataset described in section 2.1.2 could not be used due to its poor tree crown identification performance on solitary trees. If the layer had a good precision on large trees, a fully automated approach would be much easier since reliable crown detection would enable

large scale point cloud segmentation with the proposed method. A spatial join[1] is used to transfer the attribute data from the trees dataset onto the intersecting polygons from the manually created tree crown polygon layer. The polygon layer containing polygons for every tree in the dataset as well as all of the structural parameters including DBH and species information is exported as an ESRI shapefile (.shp).

3.2.2 Preparing and Indexing the LiDAR Dataset

For the mask layer to be applied to the LiDAR tiles, they must be downloaded first. Because the trees in the mask layer are evenly scattered across the whole country of Luxembourg (see Figure 2.4), many LiDAR tiles from across the country had to be downloaded. In order to avoid downloading the entire point cloud, which is very large, a small python script is used including the geopandas library to select only the point cloud zips that intersect with the mask layer. See Appendix B for the entire code. The output is a text file containing the names of the point cloud zip files that intersect with the polygons in the mask layer shapefile. After this, an existing download API script [Administration du Cadastre et de la Topographie, 2019a] is adapted to download and unzip only the files listed in the text file produced by the script mentioned above. To make finding the right tile for each shapefile later in the process easier, the bounds of each of the individual .laz files from the dataset are indexed in another small script before these indexes are saved to a pickle file. Refer to Appendix C for the code of the indexing script.

3.2.3 Vertical Tree Profile Extraction

This part of the data preparation went through the most iterations. The first approach was to iterate through the individual polygons within the shapefile and use them as a mask to create and save many individual point cloud files using the pdal-python library, one for each tree, before plotting them with the pyvista library, setting up a camera and saving an image for every plot. Though this method was stable and linear, meaning it cut, converted and saved one tree after the other, it was very slow and inefficient. The next important iteration did not rely on individual point clouds. The script (shown in Appendix D) works by first mapping the individual features(tree fooprints) in the shapefile to json geometries and putting them into one list. One geometry after the other, it then enumerates through the las bounds index and checks with which laz file(s) one geometry intersects with, before it then maps the intersecting points into one pandas dataframe per intersecting laz file. If the geometry spans over more laz files, the dataframes are being concatenated into one dataframe containing all of the points. It then creates a pyvista

[1] ArcGIS Spatial Join Tool (02.05.2024)

cloudobject, adds the points to it, sets up a camera with the distance of the camera being a factor of the length of the geometry, before taking a screenshot and saving it as a png, while it takes attributes from the shapefile feature for the naming scheme of the pngs. The structure is represented by the following list. Please refer to Appendix D for the entire script with comments.

- mapping of features in the shapefile layer to a json geometry
- initiation of an emtpy list
- enumeration through all of the laz files that the geometry intersects with *(mostly just one, but sometimes trees span over multiple tiles)*
- converting the numpy array to a pandas dataframe
- adding each of the returned dataframes to the list
- concatenating all dataframes in the list into one dataframe
- from the dataframe, creating an array containing the XYZ of the points
- from the dataframe, creating an array containing RGB values
- creating pyvista cloud object, adds points and colors
- plotting the points into the cloud using pyvista plotter
- seting up a pyvista camera
- taking a screenshot
- returned image saved as png with the DBH and tree Number in the name, separated by an underscore

To speed up this process, the *multiprocessing* library is used to process multiple features in parallel. All of the bullet points above are embedded in one function, that is applied to each individual shape within the shapefile. The *fiona* library is used to open the shapefile. The total runtime is also counted to evaluate the performance of this automated script. To create all of the 238 images, the total runtime is 124 seconds, just over two minutes. This is significantly faster than without parallelisation. The program produced 238 raster images, with the DBH in metres in the filename followed by the tree number(*treen*). The reason for the DBH being part of the filename is to extract it later from the filename in the script to train the model in order to prevent the DBH and associated tree images from being out of order. Figure 3.1 shows the file structure in the output folder.

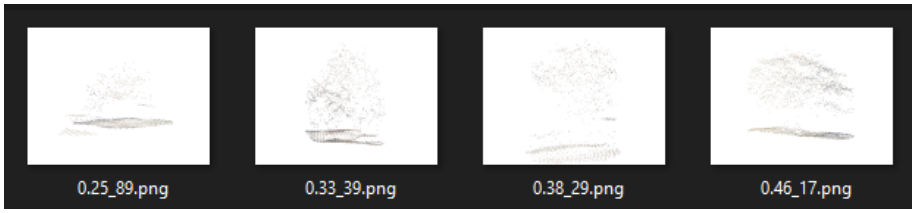


Figure 3.1: File structure of the training data with the DBH in the filename seperated by the sample number with an underscore, to couple training data value and associated picture to prevent mixups

The images are in the portable network graphic (PNG) format and have a resolution of 1024x768 pixels, with a file size of 200KB in average. The PNG file format is chosen in this study because of its lossless compression. Most of the trees in the imageset are representative. Figure 3.2 shows 9 different individual trees representative of the typical structures that can be found throughout the dataset. The tree on the bottom left represents a tree of smaller diameter, easily identifiable by the smaller number of points. Larger trees like the one on the bottom right, which is a large walnut tree, reveal their true crown form and branch structure much better, coniferous trees like the one in the middle of the righternmost row are easily identifiable by their conical crown structure, but often have little to no representation of their tree trunk, just as the small trees do.

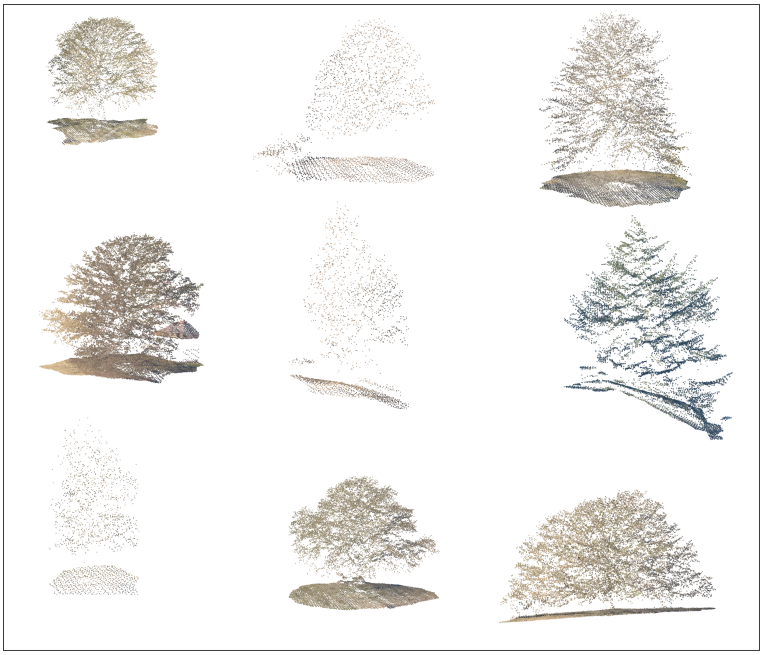


Figure 3.2: 9 examples of trees extracted from the point cloud. The variation of size and species throughout the dataset is clearly visible.

Unfortunately, not all trees are as representative as the set presented above. Figure 3.3 shows a few different examples of badly extracted trees. The top left section represents the

smallest tree in the dataset. It was measured at 16 centimetres, the tree is not properly detected by the relatively sparse point coverage of the ALS survey. The section below (middle left) just shows a small patch of ground, with no tree or any structure above. In total, there are six individual cases of this phenomenon in the dataset. The only possible explanation for this is that these trees have been planted after the ALS survey had been taken. The top right image represents a large tree without any trunk and a very unusual crown shape. Due to it being the only such tree in the dataset, it is also removed to make the dataset more homogeneous. The bottom right representation could also be found four times and could have been caused by outliers or errors in the point cloud making the cloud length longer, which results in the distance of the camera to the centre of the cloud (which is a factor of the cloud length) see Appendix D for the code, being unfavorable for a good depiction with a part of the crown being cut off. These trees are removed from the training dataset by visual examination to prevent it from training on non-representative data. No classes are removed from the classified point cloud in order to train it on real-world data. As can be seen in Figure 3.3 at the bottom left, this sometimes resulted in buildings being represented in the images. The ground points could also have been taken out, but the density of ground points in an extracted image reveals important information about the size of the tree.

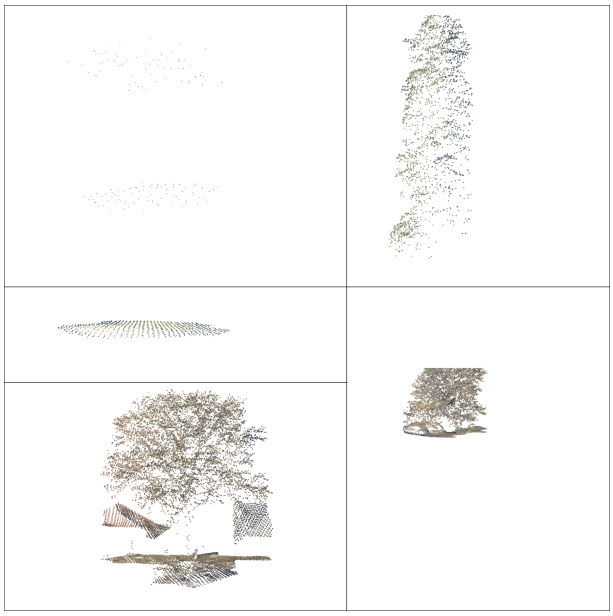


Figure 3.3: Examples of badly extracted trees, they are removed for various reasons. This proves that a visual overlook of the training data is necessary.

From the 238 images initially present in the dataset, 225 remained and 13 have been remove because of either one of the reasons mentioned above. The remaining dataset contained trees with a DBH ranging from 16cm to 206cm. With this step, the preparation

of the training data could be concluded.

3.3 Training the CNN

The training of the model was again subject to many different iterations, with only the most successful one listed here. One central part of creating a precise model is the adjustment of certain parameters. To make the ResNet50 model that is built into the *Keras* python library work for a regression analysis, a few adjustments to the base model had to be made. The documentation of the *tensorflow*[2] library as well as this post[3] on the stackoverflow website are important sources of information to understand the ResNet50 implementation of Keras and its use for a regression application respectively. One change that needs to be made is to remove the fully connected top layer used for classification. Removing this layer helps the neural network directly predict a continuous value for the regression task at hand. It can be set as a parameter:

```
base_model = ResNet50(weights=none, include_top=False)
```

The code also indicates the use of the imagenet weights, but the model is trained without using the pre-trained weights from imagenet, because the extracted vertical profiles from the point cloud do not resemble any real-world pictures, as they are visually more comparable to scatterplots.

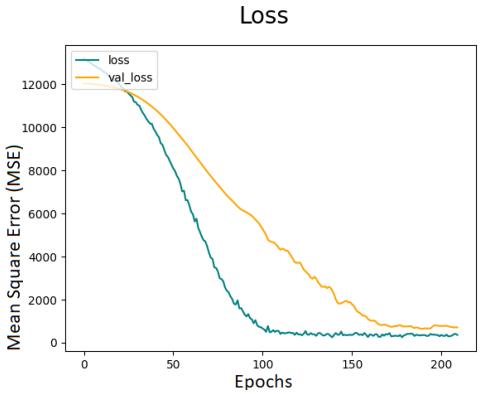


Figure 3.4: Loss plot indicating a steep learning curve with the loss function flattening out before the val loss linearly approaches the level of the loss function, showing successful training.

Figure 3.4 depicts the evolution of the loss (teal) and the validation loss (yellow) through the epochs, the linearity of those lines describes a very good training run, with the validation loss dropping steadily, but not as fast as the training loss. The loss function

[2] ResNet50 Keras Documentation (20.04.2024)
[3] stackoverflow ResNet50 Regression (22.04.2024)

(in this case, MSE loss) describes the squared difference between the actual value (DBH) and the predicted value. If this value becomes smaller over the epochs, this means that the model is able to predict the real value with increasing precision[Brownlee, 2020]. In this case, the loss and validation loss approach each other on a linear curve, meaning the learning increase was static, before staying at a constant proximity for a few epochs. Then, the early stopping function stopped the model because the loss did not decrease for more than 20 epochs. The constant proximity without bouncing up and down means the prediction accuracy was constant for a few iterations, meaning the model is reliably predicting DBH. In the following lines, the code used to train the most successful model is presented. The entire code without comments is also listed in Appendix E.

- import libraries

```
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
import numpy as np
import glob
import cv2
from matplotlib import pyplot as plt
```

- define image path and create lists for images and DBH data

```
image_paths = glob.glob(r'path_to_images\*.png')
images = []
diameters = []
```

- resize images to 224x224p, as required by ResNet, normalize color values and add them to the list

```
for image_path in image_paths:
    img = cv2.imread(image_path)
```

```
    img = cv2.resize(img, (224, 224))  # resize
    img = img / 255.0  # normalize pixel values
    images.append(img) #add to list
```

- extract diameter from filename and add to list

```
for image_path in image_paths:
    filename = image_path.split('\\')[-1]  # get filename from path
    diameter_str = filename.split('_')[0]  # split at underscore
    diameter = float(diameter_str) * 100
    # convert diameter to float and convert to cm
    diameters.append(diameter)
```

- convert lists to arrays

```
images = np.array(images)
diameters = np.array(diameters)
```

- create an 80-20 test split, meaning 80 percent of the data is used for training, 20 is used for validation, this is a standard split.

```
X_train, X_test, y_train, y_test = train_test_split(images, diameters,
test_size=0.2, random_state=52)
```

- initialize the model without weights and without the top layer, as mentioned before

```
base_model = ResNet50(weights='none', include_top=False)
```

- several dense[4] and dropout layers[5] have been added to prevent overfitting [Ibrahim, 2024]. this constellation is the result of several iterations and delivered the best performance.

[4] Weights and Biases Article(22.04.2024)
[5] Keras Documentation (22.04.2024)

```
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.2)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.2)(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.2)(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.2)(x)
predictions = Dense(1, activation='linear')(x)

model = Model(inputs=base_model.input, outputs=predictions)
```

- compile the model using the Adam optimizer and a learning rate of .0.000004, with the loss function being the MSE, which is useful for regression.

```
model.compile(optimizer=Adam(learning_rate=0.000004),
loss='mean_squared_error')
```

- defines early stopping and restore best weights to stop training when the validation loss does not improve over 10 epochs. this allows training for many epochs at a time, which was easier for the environment that the model was trained in.

```
early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)
```

- train (or fit) the model with the X_train (images) and y_train (diameters) with 500 epochs and early stopping, this was acceptable due to the fast processing time of the machine the model was running on.

```
hist = model.fit(X_train, y_train, epochs=500,
validation_data=(X_test, y_test), callbacks=[early_stopping])
```

- plot the performance of the model by plotting the loss and validation loss for visualisation of the models performance

```
fig = plt.figure()
plt.plot(hist.history['loss'], color='teal', label='loss')
plt.plot(hist.history['val_loss'], color='orange', label='val_loss')
fig.suptitle('Loss', fontsize=20)
plt.legend(loc="upper left")
plt.show()
```

- finally, save the model for later validation and use

```
model.save('dbh_rename.h5')
```

3.4 Hardware, Software and Performance

The computation and extraction of the vertical profile images was done on the CPU. The system used to run the script was equipped with an AMD Ryzen 7 5800X 8-core processor and 32 gigabytes of working memory and solid state storage. The extraction script took 128 seconds for a full extraction of the 238 trees from the point cloud tiles. This meant that it takes about one minute to process about 120 trees, making the tool useful for larger scale applications. Training the CNN model however took significantly longer. As multiple independent training runs were required to achieve good results and iterate through different parameter settings, only the time to train the most successful model presented in the results section was used. The training time of this model was 245 seconds, or 4 minutes and 8 seconds. This time was only achievable because the model could be trained using a fast GPU with CUDA support. The exact model used is called NVIDIA GeForce RTX 3070 [6], featuring 5888 CUDA cores with a base clock of 1.5 GHz and a boost clock of 1.73 GHz as well as 8 gigabytes of DDR4 Memory. The use of the GPU was only possible because a specific version of *tensorflow* was installed that offered GPU support. Table 3.1 shows the libraries used and their respective versions. All of these libraries were installed into a conda virtual environment. The computer was running a Windows 10 64-bit operating system.

[6] RTX 3070 Product Page (08.05.2024)

Description	Version
cudatoolkit	11.2.2
cudnn	8.1.0.77
keras	2.10.0
keras-preprocessing	1.1.2
opencv-python	4.9.0.80
matplotlib	3.8.4
numpy	1.26.4
pandas	2.2.2
python	3.10.14
scikit-learn	1.4.2
tensorflow	2.10.0

Table 3.1: Conda virtual environment

Chapter 4

Multiple Linear Regression

4.1 Introduction to Regression Analysis

Linear regression analysis allow the estimation of one dependent value based on one or more (multiple) predictor values that correlate with the dependent value

[Johnson and Wichern, 2007]. This allows the estimation of one value, in this case tree diameter from different predictive values like tree height that can directly be obtained from forest point clouds using specialized software. LiDAR has a lot of advantages for forestry applications, it is fairly straightforward to derive certain metrics such as tree height from airborne LiDAR. Back in the early 2000s, when LiDAR scanning became more popular and applicable, certain studies reported accuracies of 0.02 metres [Yu et al., 2004]. Since then, a strong correlation between parameters which can directly be acquired from airborne LiDAR or airborne imagery and DBH has been proven by many publications [Machado et al., 2019, Mao et al., 2023] as well as the statistics section 2.4.1 of this thesis. *When using textural metrics alone, the estimated DBH accuracy was the highest,[...] using the MLR model*[Mao et al., 2023]. This sentence, copied from the abstract of a 2023 paper that compared MLR with random forest and support vector machine concluded that MLR was the best performing model for their application. Moreover, Liu et al stated in an overview paper from 2021 that MLR and other parametric methods had been commonly used for the identification and estimation of forest structural parameters [Liu et al., 2021]. In another study, by Yao Wei, Krzystek Peter and Heurich Marco, that was carried out in forest plots, DBH was estimated using Height, Crown Area and Crown volume as the predictor values for a regression analysis. They concluded that DBH can be measured to an accuracy (RSME) of 6 cm, with broadleaf trees being less accurately determinable than coniferous trees.[Yao et al., 2012]. [Machado et al., 2019] presented, in their work a predicted DBH accuracy of 8.7cm, but only on one tree with a handful of training trees for regression analysis. While the work of Machado et al. only measured a small number

of trees, the focus of the publication lied on creating a linear regression model. The linear regression model created in this work will be very similar to the models of Machado et al. and Yao et al. since these approaches are representative of a reliable and universally applied approach to estimate DBH.

4.2 Model Creation

The regression model was created in a python environment using the *scikit-learn, scipy, pandas* and *numpy* libraries. This allowed fast data processing with cross validation and statistics. The extracted parameters are listed in Table 4.1.

Parameter	Acquisition Method	Precision	Unit
H_{tree}	Airborne LiDAR	3 cm	metres
A_{crown}	Manual delineation (Aerial Imagery)	unknown	m^2

Table 4.1: Predictor Values

Table 4.2 reveals average statistics of the training and validation data. Since the regression model was calculated using 50-fold cross-validation with no defined random state, for every fold, the split was always 80-20, but the data points were different ones. Therefore, the N stayed the same while the Mean and Standard deviation were different for every fold and got averaged for the table. The same data samples that were used for the CNN model were also used for the MLR model, as some images had to be removed from the dataset, the same rows also had to be removed from the numerical dataset to train the MLR model on exactly the same trees that the MLR model is trained on. This was manually done by removing the 13 rows containing the tree numbers corresponding with those that were removed from the image dataset.

Training Data	A_{crown}	H_{tree}	Validation Data	A_{crown}	H_{tree}
N	180	180	N	45	45
Mean	193.35	17.31	Mean	206.60	16.39
Std.Dv (σ)	136.25	6.65	Std.Dv (σ)	142.59	5.57

Table 4.2: Statistical summary, training and validation data

The linear regression model was created using the *LinearRegression*[1] model that is built into the *Scikit-Learn* (sklearn) python library. Processing time was very short, the python script, including the 50 fold cross validation, and the validation analysis including scatterplot as well as a categorized RSME histogram were integrated into the python

[1] sklearn LinearRegression Documentation (09.05.2024)

script, which can be found in the Appendix F. The decision to incorporate 50 fold cross validation was a result of iterating through a few different settings, I discovered that for 50 folds, the RSME as well as the standard deviation did not improve or change significantly anymore, therefore this created good and repeatable results for this dataset without creating unnecessary computation efforts.

4.3 MLR Model

This subsection describes the MLR equation that was created during the processing. This equation was applied to every data point multiple times to calculate the DBH in the validation process. The coefficients for the independent variables in the model what was saved for evaluation were β_1 is 0.00083393 and β_2 is 0.02905108.

$$DBH = \beta_0 + \beta_1 \cdot H_{\text{tree}} + \beta_2 \cdot A_{\text{crown}} + \epsilon \tag{4.1}$$

where

- DBH is the dependent variable (Diameter at Breast Height)
- β_0 is the y-intercept
- β_1 and β_2 are the coefficients of the independent variables H_{tree} (Height of the tree) and A_{crown} (Area of the crown) respectively
- ϵ is the error term

Chapter 5

Results

This chapter is presenting the results of the study, therefore, the results of both the CNN model as well as the regression analysis are shown in the following sections individually, before being listed next to each other to allow for a more direct comparison between the two models.

5.1 CNN Model

The model based on the ResNet50 neural network was evaluated on the training dataset using the *.evaluate()*[1] function that is built into Keras. The evaluate function was applied to the same test split that the model was tested on and produced an MSE loss of 645.65. Since this loss is the mean squared error of the predicted value from the actual value, its square root is actually the mean error in centimetres.

$$\sqrt{645.65} = 25.41cm \tag{5.1}$$

The calculation above shows that the average error in DBH prediction the validation of the dataset is 25.41 centimetres. Figure 5.1 plots the estimated DBH and the true DBH of the test set. The plot does not show many estimations, nevertheless, Figure 5.1 confirms that the model was able to recognise a positive linear relationship between the raster images and the DBH.

[1] Keras Documentation (10.05.2024)



Figure 5.1: Test split evaluation

To test the model on a larger and unknown dataset, a synthetic but representative dataset had to be created, given the fact that further field trips to collect more data are not possible. To create this dataset, the idea was to extract the individual trees from the point cloud again, but with the camera rotated 90 degrees, since the vertical profile of a tree changes significantly when viewed from another angle. Due to the heterogeneic morphology of free standing trees, most of the extracted rasters looked significantly different than their original counterpart.

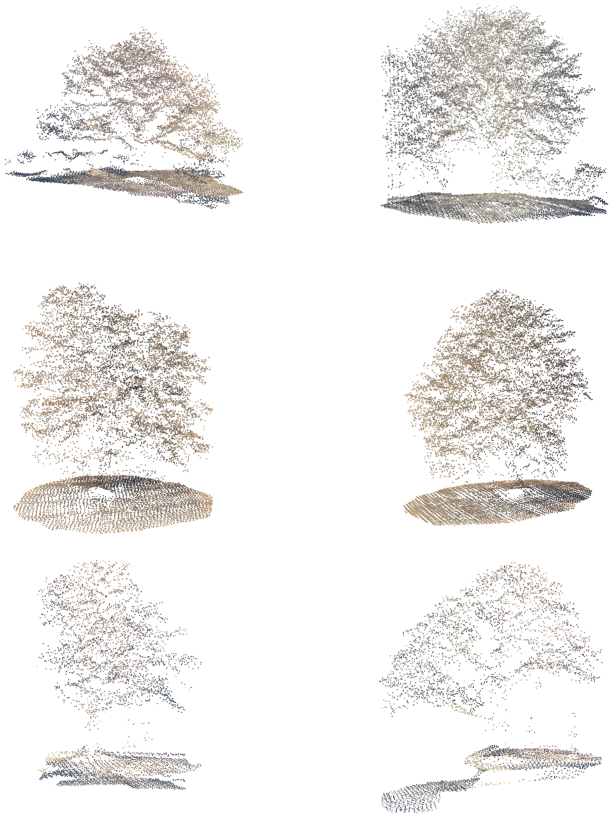


Figure 5.2: Tree extractions with the camera rotated 90 degrees, right and left is always the same tree, but from another angle. This creates a solid validation dataset.

Figure 5.2 shows 3 representative examples of the transformation with trees from the training dataset (left) and their 90 degree rotated counterparts that were used as validation dataset (right). Visually, the results were very good. while some trees were similar in appearance, other trees were surprisingly different. After carefully inspecting the newly created data, I concluded that it was different enough to test the model. The model cannot be tested on the same data that was used for fitting, as this leads to non-representative results. In this case, the structure of every image is different to all of the images the model has seen before. The RSME of the transformed dataset was 32.39 centimetres.

$$\sqrt{1049.55} = 32.39cm \tag{5.2}$$

This average accuracy value seems significant, but due to the heterogenic distribution of points in the dataset worsened by the fact that a large number of trees had been removed from the dataset, the specific accuracy for the ranges where little trainingdata was available contributes to this high average accuracy. It is important to look at the distribution of data samples across the DBH range and the related accuracies. Figure 5.3 proves that the distribution of DBH is not linear at all in the dataset. Most DBH values are in the range

between 50 and 125 centimetres, with only 6 trees remaining in the group between 175 and 200cm.

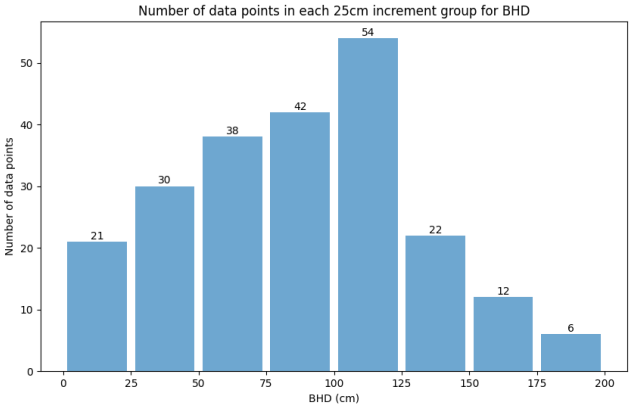


Figure 5.3: Distribution of dataset samples according to DBH in groups of 25cm

This highly unequal distribution strongly corresponds to the difference in RSME for the same groups. However this may not be the only cause for the variance in specific RSME, it is discussed in the next chapter. But it definately shows that the average RSME of 32.39cm does not deliver sufficient information. Figure 5.4 clearly shows a precision of less than 30 centimetres for the groups that offer a more than 30 data points. The ranges with more values correlate to a higher precision. The precision of the RSME for the groups with the most data points are down to 17.7cm and 21.91cm respectively.

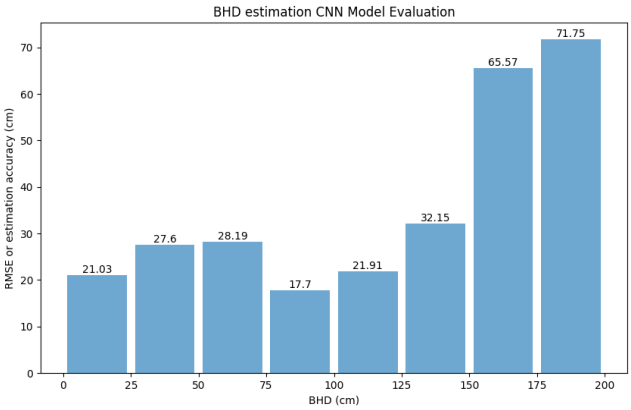


Figure 5.4: RSME distribution in 25cm groups

Figure 5.5 visually shows that up to a range of 120cm in DBH, predictions are mostly linear, but for larger measured diameters, the estimated DBH remains at a constant level despite an increase of DBH. This indicates that the models refrains from predicting higher DBH values.

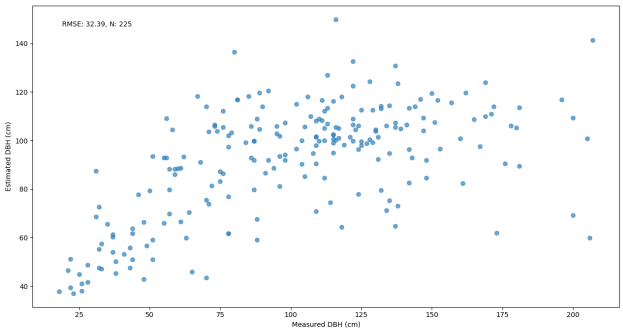


Figure 5.5: Scatterplot estimated vs measured DBH

5.2 Regression Analysis

Same as for the CNN results, Figure 5.3 describes the distribution of the data points in the dataset. The overall RSME of the MLR model was 35.55cm, therefore slightly higher than the RSME of the CNN model. By visually interpreting Figure 5.6 it is obvious that the accuracy again correlates even stronger to the distribution of RSME throughout the dataset. However, for all groups except 150-175cm, the RSME is lower, indicating a slightly better performance of the CNN model.

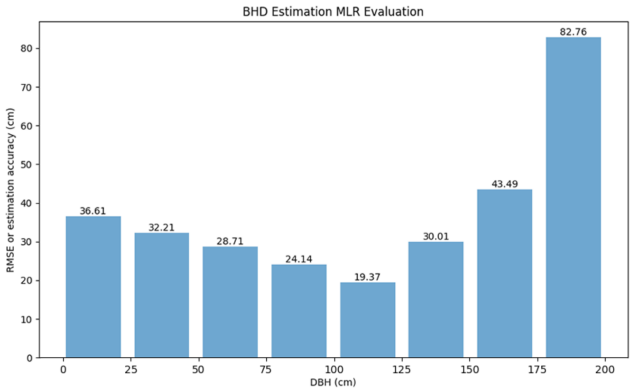


Figure 5.6: RSME distribution in 25cm groups

The scatterplot of Fig.5.7 reveals that the distribution of estimations and measurements are very different from those originating from the neural network. In the smaller DBH ranges of less than 110cm, the estimations are very linear and concentrated, but with increasing diameter, the scattering and divergence become larger. There is a recognisable trend for large DBH of $>$ 150cm to be estimated much smaller than they actually are. There are however more estimations of higher diameters compared to the scatterplot of the ResNet model.

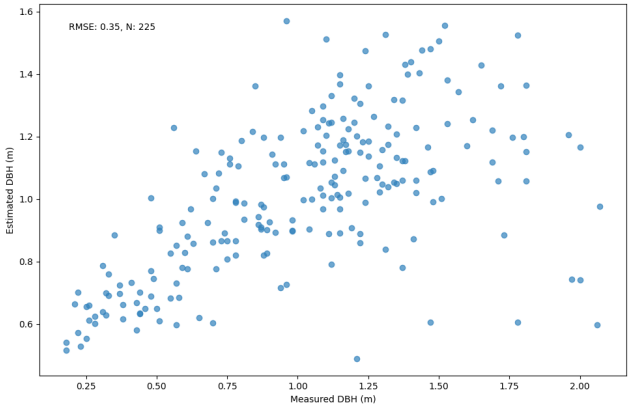


Figure 5.7: Scatterplot estimated vs measured DBH

5.3 Comparison

Table 5.1 juxtaposes the RSME of both methods for every class evaluated. It is important to note that due to the very low number of samples, the class above 200cm was not evaluated in order to more effectively visualise and illustrate the findings in the other classes. The minimal RSME was 17,7cm and 19.4cm for CNN and MLR respectively. The highest RSME was 71.8cm and 82.8cm for CNN and MLR respectively. The CNN model consistenty outperformed the MLR model in classes where solid training data was present is significant.

Description	CNN	MLR
N	225	225
Overall RSME	32.39	35.55
0-25cm	21.03	36.61
25-50cm	27.6	32,21
50-75cm	28.19	28.71
75-100cm	17.7	24.14
100-125cm	21.91	19.37
125-150cm	32.15	30.01
150-175cm	65.57	43.49
175-200cm	71.75	82.76

Table 5.1: Accuracies of both models

5.4 Temporal Mismatch

An important factor possibly influencing the results to a limited degree is the temporal mismatch between the DBH measurements and the data entering the estimation models, both the MLR and the CNN. The DBH measurements originate from early 2024, while the LiDAR survey has been taken in 2019. For the MLR approach, the height does not correspond exactly to the DBH at the time of measurement. The height can be assumed to be generally larger than in 2019. For the CNN model, the input images represent trees with smaller DBH than the measured diameter. However, due to the produced error being consistent for all data samples, it can be described as a systematical error in both models. This means that the models predict DBH that is systematically smaller than real world observations. It also means that the accuracy and RSME are still acceptable, as the precision with which both models are able to predict DBH is not reduced. The ability of the model to link the input data to the images and find patterns in the data is still confirmed by the results. The potential issue of trees not being present has been addressed in the dataset training section, where a visual analysis of the training data is described. The problem caused by the mismatch and possible solutions are further explained in the conclusion.

Chapter 6

Discussion

The results of these different approaches indicate a similar performance of both models on the same dataset. The CNN is successful at reliably identifying and learning the structures present within the individual images and delivering accurate predictions for new data. It is important to note that the accuracies achieved in this study and presented in the *Results* section cannot be compared to those from the forestry publications cited throughout the work. The results must be interpreted in the context that the methods of estimation, especially the MLR was adapted from forestry applications to an ecological application, where free standing trees are measured. The main difference is the nature of the environment that the technology is being used in. Most forests in which tree DBH need to be measured can be considered productive forest plantations with only very little diversity in terms of species and tree age. The trees are often of the same or similar species, straight and on in the same age [Gao and Kan, 2022]. In open landscapes, and especially in ecologically important areas, trees of every size, age and of different species stand close to each other. The heterogeneity of these areas lead to different accuracy requirements and expectations. Unfortunately, no publications on DBH estimations of trees in open landscapes could be identified and therefore the establishment of a frame of reference was only possible through the MLR results. Therefore, the interpretation of the results can only be done with this difference to productive forests in mind. The dataset used to create both models is extensively checked through direct measurements and remote (Google street view) verification of the general appearance, species and size of the tree. The LiDAR data delivers very precise and reliable height information. The precision with which the diameter was estimated was directly dependent on the number of training images for this specific range of diameters. For larger diameters, the precision with both methods was significantly lower, to the point where the estimation was not delivering useful values anymore. However, in these ranges the number of individual

samples in the dataset was very low ($N = 12$ and $N = 6$). The dataset section proves that more large trees were initially planned to be used for model training, but due to many large trees having been removed from the dataset there were most likely not enough samples left to precisely train a neural network for a satisfying estimation of these larger diameters. The lack of samples could also be responsible for the observation that can be obtained from the CNN scatterplot (Figure 5.5) where the neural network did not estimate DBH in ranges of more than $140cm$. Maybe, cross validation could have been a possible solution, as previously mentioned, but the low amount of samples would have been insufficient. However, the lack of data may not be the only cause for a decrease in estimation accuracy with higher diameters. The scatterplot describing the estimation from the linear regression model (Figure 5.7) showed many estimations that were much to small for their diameter. When thinking mathemetically about the equation of the MLR model, predicted DBH that is very small means that either the height and or the crown area must have been very small. This can be caused by the lack of species specific correction factors, but also by the presence trees in the open landscape that do not fit the linear regression of diameter, height and crown size. As trees grow with age, larger diameters correspond with older trees. When thinking about the fact that a very large tree with a diameter of around 2 metres is most likely few hundred years old, the tree had to endure many storms, possibly lightning strikes and damage due to human activity. These are all factors that can reduce the crown size and the height of a tree while the DBH does not shrink. Figure 6.1 demonstrates this by depicting a large tree that has been struck by lightning[1].

[1] Picture by "Archie UK" on flickr.com (10.05.2024)



Figure 6.1: A tree wich a large DBH but a small total heigt, that proves why the variance in a model of height, crown size and DBH can be very high. While these are hard to predict statistically, a CNN can be trained on their specific apperance in ALS data.

Trees can endure this kind of damage and continue to live and stay in the landscape for many years. These tree shapes are very challenging to accurately estimate using any parameters that can be extracted for a linear regression analysis. In fact, the variance of DBH increases with age and therefore usually with the height. The estimation of the diameter of these trees also poses challenges for the CNN approach. Since such trees often translate to most LiDAR points being located on the tree trunk and the main branches themselves compared to a normal tree, the model struggles with identifying the structures present in this depiction. But with the presence of good training data, the neural network could learn to associate their LiDAR signature to the correct diameter. The heterogeneity of trees in the open landscape, even within a single species due to the factors mentioned above will remain one of the challenges that will be hardest to address for both models, this may be the most important reason for the difference in precision of the results from the multiple linear regression compared to the presented literature findings which were much more precise,presenting accuracies of 6 centimetres [Yao et al., 2012]. One important factor that needs to be mentioned is the effect of the tree species on the DBH and the height of a tree. The fact the trees have specific growth rates and height to DBH ratios is established knowledge. This has been analysed and documented by many publications like [Yao et al., 2012] and [Schelhaas et al., 2018]. Species classification in the field of

remote sensing can be considered to be a challenge on its own and the development of different methods to accurately classify trees by their species could also be achieved using the presented CNN approach. Instead of the regression analysis, different classes could be created, one for each tree species to identify trees. The dataset presented more than 40 different species of trees and for a very large portion of the species, only a very insignificant number of trees were present. Table 2.6 lists the species and the number of trees belonging to one species. Therefore, and because of the limited amount of resources available for the study, the applicability of this approach for species detection remains to be tested. Many different tree species present an individual branch structure and crown geometry that could theoretically be identified with computer vision and therefore also with neural networks that can be trained with image datasets. For multiple regression analysis, correcting factors that are species-specific can be added to increase the precision of the regression analysis, but the automatic identification of the species must be executed otherwise. Since the objective of the study is the analysis of tree DBH estimation performance the MLR method and the applicability of the CNN method for estimating DBH, tree species is not a requirement. While it should be clear that knowing the tree species and a specific correcting factor improves estimation performance, both approaches can be evaluated for their performance and precision without taking species into account. An important limiting factor to the precision of the CNN approach was the point density that was available for this study, with 15 points per square metre on average, the density is fairly low compared to other studies that did similar research [Kattenborn et al., 2021]. The density had a particularly strong effect on the representation quality of small trees. Because of the nature of LiDAR scanners, every area gets the same amount of measurements, smaller objects are less covered and therefore harder to detect. The smallest trees within the dataset presented a diameter of around $20cm$, they were barely represented on the image, sometimes without any hits on the trunk itself, as shown in Figure 6.2.



Figure 6.2: A very small tree, proving that higher resolutions are beneficial for smaller objects.

The precision of the CNN model was not significantly lower for smaller DBH ranges than for larger ones, this could be caused due to the models ability to estimate DBH from the amount and distribution of ground points and crown points, associating low DBH values with low total point numbers. Nevertheless, a higher number of hits on the tree corresponds to a better representation of the trees structure, especially to its underlying trunk and branch structure, from which DBH can theoretically be directly measured. By the end of 2024, a new LiDAR point cloud of 30 points per m^2 on average will be available publicly for the entire area of the country of Luxembourg. [Administration de la nature et des forêts, personal communication, 2024] A doubling of the point density significantly improves estimation accuracies. It further benefits the identification and the distinguishing of different tree species from each other. Another important factor to consider is individual tree identification and segmentation. Like tree species identification, it constitutes important research field in its own right. Individual tree identification and delineation can be carried out using watershed algorithms or density approaches in LiDAR point clouds[Henrich et al., 2023]. Alternatively, they can be identified in multispectral aerial imagery and be used to segment the point cloud or by combining data from both acquirement methods [Deng et al., 2016]. While this study was planning on using a dataset containing already segmented tree crowns described in the *Open Government Data* section, its precision did not meet the requirements for proper extraction. Therefore, the crowns were delineated manually based on aerial images. Another promising approach is to segment the trees from aerial imagery using CNNs. Esri analytics published a pretrained cnn model[2] which can be downloaded and used to segment trees in any RGB

[2] ESRI Tree Segmentation

aerial imagery dataset . With this in mind, the CNN approach proved to be a reliable and strong alternative to the more widely used MLR analysis techniques. Both methods were employed with minimal additional factors included, tree species, soil type, crown length and more parameters can be added to both techniques in order to significantly increase their accuracies. The ease of use of the CNN model when it is trained with images only means that the technology may be used more widely in the future by researchers that have not yet had the access or skills to train and use CNNs before. With a larger training dataset, a higher point cloud resolution and the resulting possibility of identifying and taking the effect of the species of tree into consideration, the best achieved mean accuracy of 17.7cm can be replicated or improved for every diameter range, making the technology much more useful for a wider set of applications including forestry uses. For very large datasets, processing time would be higher for the CNN approach than for the MLR approach. The extraction of individual tree points from the point cloud and the translation to vertical profile rasters takes about 2 minutes for 220 trees, and the estimation by the CNN model of 200 trees takes 25 seconds. All of this takes significantly longer than applying one regression equation to a very large dataset of a few thousand trees. When working on a well equipped stationary computer. While the estimation of such datasets is possible using CNN models, it requires expensive hardware to be used on large scales. Moreover, it is important to state that the compatibility of the estimation model must be tested against different point clouds. Since there are no standards for the point density, scan angle or flight parameters of airborne LiDAR surveying, point clouds originating from different companies and organisations differ from each other. If the model estimates tree DBH based on the point density (among other factors) in the image, there will be a systematical estimation error when using the same model on a different point cloud featuring a higher point density, as a tree with a DBH of 20cm could now be represented by a multitude of the original number of points (Figure 6.2).

In relation to RQ1 in Section 1.4 Research Questions, the accuracy of the model does not offer the possibility to reliably distinguish between trees with a DBH higher or lower than 30 cm, as expected in the BK18 Biotope definition. Neither the accuracy of 27.6 cm for the specific DBH range nor the maximal accuracy of 17.7 cm could be used for a clear distinction of this precise value. Nevertheless, an accuracy of 17.7 cm in the range of diameters found in nature is useful for telling small, medium and large trees apart, and it can definitely be used to identify large and monumental trees in the open landscape worthy of protection. In the previous chapters, some key issues with the process have been found that can be fixed such as the temporal mismatch, the unequal distribution of the diameter of the training samples and the lack of special samples. Even species estimation

and subsequent species-specific DBH estimation can be included in this methodology to be used on a large scale. These improvements could lead to a significant increase in the accuracy of the predictions and may lead to a satisfying accuracy regarding the 30 *cm* benchmark. However, RQ2 can be answered by more positive results. An 8.8% increase in accuracy proves that the CNN method provided more precise estimations despite having restrictions due to the reasons mentioned above. Therefore, this increase in accuracy will increase if some of these problems have been accounted for. The prediction accuracy was not significantly affected by the poor representation of small trees due to ground density issues, despite some trees being very sparsely represented. Notably, this dataset only contained trees with diameters down to about 20 *cm*. Smaller trees have not been evaluated. The prediction accuracy dropped for diameters above 150 *cm*, this could be caused by 2 reasons: (1) the low amount of training samples led to the network not being able to learn the characteristics of large trees and (2) the model learned that by guessing smaller numbers, it had a higher likelihood of generating an accurate prediction. In relation to RQ3, the research also confirms that the CNN approach constitutes a promising alternative on large scales, with its higher accuracy and acceptable processing times for up to a few thousand trees on a performance oriented windows computer, the method can be applied on larger Scales. For 100.000 individual trees, the machine used in the study would require roughly 15 hours of processing time. By using the much more powerful processing capabilities of specialized labs, the approach presented in the study can be an influential innovation in this field that is practical at large scale applications. For such applications, it is worthwhile to address the issues to investigate the full potential capacity of the presented method. Due to the predicted release of new, more precise scans in late 2024, this thesis ends at this level of precision. The efforts to train a new model using the existing shapefile and DBH values are very low, the scripts can be run again in a similar configuration with the new data. It was positively observed that the field measurements have been coincidentally collected at roughly the same time the flights of the next survey have been undertaken.

Chapter 7

Conclusion

The CNN was able to predict the DBH of trees throughout the dataset with an average error of 32.39cm. Considering that the diameters in the dataset ranged from 18cm to 207cm, a closer analysis of the diameter-dependent accuracy revealed that the average error of the ranges with significant training data available moved towards 20cm, with 17.7cm being the most precise result. In a range of 18-207cm centimetres, a precision of 17.7cm is an acceptable value, especially given the relatively small size, low point density, and heterogeneity of the training and validation dataset. The regression analysis, which is a standard predictive approach in forestry applications was inspired by established publications and delivered similar, but slightly less precise estimations with an average error of 35.55cm and the smallest error being 19.37cm. This proves that training a residual neural network approach was, despite sub optimal training data, 8.8% more accurate. It solidly confirms the applicability of the model for estimating DBH. A key issue with temporal mismatch described in the *Results* chapter is making the models systematically predict smaller diameters than in the real world as they were trained with 5 year old structural and height data and much more recent measurements. But its impact on the accuracies that can be acheived with both methods is still limited. Possible solutions that could have been applied include correcting the DBH in the dataset using allometric growth equations. One of the largest investigations of european tree diameter increment models was carried out by Schelhaas et al. in 2018 and it considered 99 potential variables to diameter growth [Schelhaas et al., 2018]. Another solution would be to collect the data at roughly the same time that the LiDAR scans are being performed to prevent temporal mismatch. LiDAR surveys at large scales are still a very expensive undertaking and therefore often not carried out regularly. A lot of planning must go into creating a LiDAR dataset and its derivative products. Field measurements for these products must be planned accordingly. Improvements and developments to the presented estimation

accuracy can be made by adding species information to the training data and by including irregular trees to the training dataset in order to account for the increasing variance in older trees. Despite all of these opportunities for improvement, working with artificial neural networks is computationally intensive and requires a certain amount of processing time an knowledge of the digital environments and technical platforms that are a requirement for the software to run. The widespread use of deep learning may lead to more specialized computing hardware becoming increasingly affordable in the future, but the established method of using regression analysis for DBH estimation will continue to be far more efficient energy wise and in terms of scaleability but it must be said that given appropriate resources, the CNN approach can also be applied on large scales. The advantages of CNNs lie in the visual interpretability of the data. Outliers in the dataset such as oddly shaped trees or trees that broke off and lost an important part of their height cannot be identified by the commonly used parameters in a regression analysis, but they can be identified visually by CNNs. Finally, individual tree extraction remains a challenge and many recent publications explore different ways of segmenting different remote sensing datasets. The automatic extraction based on manually drawn crown circumference polygons presented in the study is a solid and fast alternative for the time being, but cannot be fully automatised. The creation of point cloud extractions relies on reliable individual tree segmentation, and is a requirement for a fully automated workflow and for large datasets. Figure 7.1 summarizes the most important qualitative and quantitative findings of the study. Table 7.2 shows the limitations of this work, especially of training and applying the CNN model and proposes ideas that could develop the findings to create a fast, precise and efficient tool for DBH estimation. Further research must investigate the possibilities for improvement to explore the full capacity of the presented framework on higher resolution data. Due to the complex structure of trees and the high variance of the DBH in individual trees, I expect a certain limit to the accuracy of this method on large scale ALS data. This limit should be explored to fully comprehend the ability of using residual neural networks to estimate parameters based solely on structural vegetation data.

Description	CNN	MLR
RSME	32.39cm	35.55cm
RSME min	17.7cm	19.37cm
RSME max ($DBH < 2m$)	71.75cm	82.76cm
Individual Tree Extraction	Required	Not required
Computation requirements	High	Low
Odd tree identification	Yes	No
Lidar resolution requirements	High	Low
Species identification	Yes	No
Possibility to add additional factors	Yes	Yes

Table 7.1: Qualitative and quantitative findings

Issue	Potential Solutions
unequal distribution of data samples across the diameter range	when performing a regression task with CNN, training data samples must be evenly distributed across the entire range of possible estimations
manual delineation of tree crowns	automatic individual tree identification using LiDAR poind cloud calculations or multispectral aerial imagery
inaccurate DBH estimation of large trees	expand range of trees in the training data
inaccurate DBH estimation of oddly shaped trees	identification and addition of oddly shaped trees to training data
no species estimation	representation of many samples of common tree species in the dataset and implement species detection into this method
limited representation of smaller trees	accessing high resolution point cloud datasets
point cloud compatibility	evaluation of the CNN model on different point clouds
temporal mismatch	apply correction models and formulas or synchornise remote data collection with field measurements

Table 7.2: Issues encountered in the study and recommended actions for future research

Bibliography

[Administration du cadastre et de la topographie, 2019] Administration du cadastre et de la topographie (2019). OGD LiDAR dataset description. https://data.public.lu/fr/datasets/lidar-2019-releve-3d-du-territoire-luxembourgeois/#description. 22.04.2024.

[Administration du Cadastre et de la Topographie, 2019a] Administration du Cadastre et de la Topographie (2019a). opendatalu download api. https://github.com/opendatalu/api_examples/blob/master/download_resources.py. 22.04.2024.

[Administration du Cadastre et de la Topographie, 2019b] Administration du Cadastre et de la Topographie (2019b). Ortophoto 2019 wms. https://data.public.lu/fr/datasets/bd-l-ortho-webservices-wms-et-wmts. 06.04.2024.

[Administration du cadastre et de la topographie, 2022] Administration du cadastre et de la topographie (2022). OGD Tree Crown Shapefile layer. https://data.public.lu/fr/datasets/lidar-2019-arbres/#resources. 22.04.2024.

[Ayrey and Hayes, 2018] Ayrey, E. and Hayes, D. J. (2018). The use of three-dimensional convolutional neural networks to interpret lidar for forest inventory. *Remote Sensing*, 10(4).

[Brownlee, 2020] Brownlee, J. (2020). How to choose loss functions when training deep learning neural networks. https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/. 22.04.2024.

[Bucksch et al., 2013a] Bucksch, A., Lindenbergh, R., Abd Rahman, M. Z., and Menenti, M. (2013a). Breast height diameter estimation from high-density airborne lidar data. *IEEE Geoscience and Remote Sensing Letters*, 11(6):1056–1060.

[Bucksch et al., 2013b] Bucksch, A., Lindenbergh, R., Abd Rahman, M. Z., and Menenti, M. (2013b). Breast height diameter estimation from high-density airborne lidar data. *IEEE Geoscience and Remote Sensing Letters*, 11(6):1056–1060.

[Bucksch et al., 2010] Bucksch, A., Lindenbergh, R., and Menenti, M. (2010). Skeltre: Robust skeleton extraction from imperfect point clouds. *The Visual Computer*, 26:1283–1300.

[Chen et al., 2020] Chen, W., Xiang, H., and Moriya, K. (2020). Individual tree position extraction and structural parameter retrieval based on airborne lidar data: Performance evaluation and comparison of four algorithms. *Remote Sensing*, 12(3):571.

[Coops et al., 2022] Coops, N. C., Tompalski, P., Goodbody, T. R. H., Achim, A., and Mulverhill, C. (2022). Framework for near real-time forest inventory using multi source remote sensing data. *Forestry: An International Journal of Forest Research*, 96(1):1–19.

[Deng et al., 2016] Deng, S., Katoh, M., Yu, X., Hyyppä, J., and Gao, T. (2016). Comparison of tree species classifications at the individual tree level by combining als data and rgb images using different algorithms. *Remote Sensing*, 8(12):1034.

[Drever, 2022] Drever, J. (2022). Forest carbon 101. *The Nature Conservancy*.

[ESRI, 2024] ESRI (2024). World imagery tile layer. https://www.arcgis.com/home/item.html?id=10df2279f9684e4a9f6a7f08febac2a9#! 22.04.2024.

[European Commission, 2024] European Commission (2024). Council directive 92/43/eec. https://environment.ec.europa.eu/topics/nature-and-biodiversity/habitats-directive_en. 22.04.2024.

[Gao and Kan, 2022] Gao, Q. and Kan, J. (2022). Automatic forest dbh measurement based on structure from motion photogrammetry. *Remote Sensing*, 14(9).

[Grand Duchy of Luxembourg, 2018] Grand Duchy of Luxembourg (2018). Règlement grand-ducal du 1er août 2018. https://legilux.public.lu/eli/etat/leg/rgd/2018/08/01/a774/consolide/20220722#intituleAct. 22.04.2024.

[GreenValley International, 2024] GreenValley International (2024). Lidar in forestry. https://www.greenvalleyintl.com/forestry/. 22.04.2024.

[HAGLÖF SWEDEN, 2024] HAGLÖF SWEDEN (2024). Mantax blue product page. https://haglofsweden.com/project/mantax-blue/. 22.04.2024.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Henrich et al., 2023] Henrich, J., van Delden, J., Seidel, D., Kneib, T., and Ecker, A. (2023). Treelearn: A comprehensive deep learning method for segmenting individual trees from forest point clouds. *arXiv preprint arXiv:2309.08471*.

[Hupke, 2020] Hupke, K.-D. (2020). *Small Biotopes and Their Importance for Biodiversity and Nature Conservation*. Springer Nature.

[Ibrahim, 2024] Ibrahim, M. (2024). The Basics of ResNet50. https://wandb.ai/mostafaibrahim17/ml-articles/reports/The-Basics-of-ResNet50---Vmlldzo2NDkwNDE2#step-4:-building-resnet-50-model. Weights and Biases.

[Johnson and Wichern, 2007] Johnson, R. A. and Wichern, D. W. (2007). Applied multivariate statistical analysis. 6th. *New Jersey, US: Pearson Prentice Hall*.

[Katam Technologies AB, 2024] Katam Technologies AB (2024). About katam. https://www.katam.se/company/. 22.04.2024.

[Kattenborn et al., 2021] Kattenborn, T., Leitloff, J., Schiefer, F., and Hinz, S. (2021). Review on convolutional neural networks (cnn) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:24–49.

[Khan and Khan, 2017] Khan, A. S. and Khan, A. S. (2017). An introduction to medicinally important trees. *Medicinally Important Trees*, pages 1–20.

[Lee et al., 2023] Lee, H., Calvin, K., Dasgupta, D., Krinner, G., Mukherji, A., Thorne, P., Trisos, C., Romero, J., Aldunce, P., Barret, K., et al. (2023). Ipcc, 2023: Climate change 2023: Synthesis report, summary for policymakers. contribution of working groups i, ii and iii to the sixth assessment report of the intergovernmental panel on climate change [core writing team, h. lee and j. romero (eds.)]. ipcc, geneva, switzerland.

[Leite et al., 2020] Leite, R. V., Silva, C. A., Mohan, M., Cardil, A., Almeida, D. R. A. d., Carvalho, S. d. P. C. e., Jaafar, W. S. W. M., Guerra-Hernández, J., Weiskittel, A., Hudak, A. T., et al. (2020). Individual tree attribute estimation and uniformity assessment in fast-growing eucalyptus spp. forest plantations using lidar and linear mixed-effects models. *Remote Sensing*, 12(21):3599.

[Li et al., 2023] Li, X., Zheng, Z., Xu, C., Zhao, P., Chen, J., Wu, J., Zhao, X., Mu, X., Zhao, D., and Zeng, Y. (2023). Individual tree-based forest species diversity estimation by classification and clustering methods using uav data. *Frontiers in Ecology and Evolution*, 11:1139458.

[Liu et al., 2020] Liu, H., Shen, X., Cao, L., Yun, T., Zhang, Z., Fu, X., et al. (2020). Deep learning in forest structural parameter estimation using airborne lidar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:1603–1618.

[Liu et al., 2021] Liu, Q., Ma, W., Zhang, J., Liu, Y., Xu, D., and Wang, J. (2021). Point-cloud segmentation of individual trees in complex natural forest scenes based on a trunk-growth method. *Journal of Forestry Research*, 32(6):2403–2414.

[Machado et al., 2019] Machado, M., Tommaselli, A., Tachibana, V., Martins-Neto, R., and Campos, M. (2019). Evaluation of multiple linear regression model to obtain dbh of trees using data from a lightweight laser scanning system on-board a uav. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:449–454.

[Maltamo et al., 2014] Maltamo, M., Næsset, E., and Vauhkonen, J. (2014). Forestry applications of airborne laser scanning. *Concepts and case studies. Manag For Ecosys*, 27:460.

[Mao et al., 2023] Mao, Z., Lu, Z., Wu, Y., and Deng, L. (2023). Dbh estimation for individual tree: Two-dimensional images or three-dimensional point clouds? *Remote Sensing*, 15(16):4116.

[Ministry of the Environment, 2018] Ministry of the Environment (2018). LIS-L Land Use. https://data.public.lu/fr/datasets/lis-l-land-use-2007-2015-2018/#resources. 22.04.2024.

[Natesan et al., 2019] Natesan, S., Armenakis, C., and Vepakomma, U. (2019). Resnet-based tree species classification using uav images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:475–481.

[Parmain and Bouget, 2018] Parmain, G. and Bouget, C. (2018). Large solitary oaks as keystone structures for saproxylic beetles in european agricultural landscapes. *Insect Conservation and Diversity*, 11(1):100–115.

[Ritter and Dauksta, 2013] Ritter, E. and Dauksta, D. (2013). Human–forest relationships: ancient values in modern perspectives. *Environment, development and sustainability*, 15:645–662.

[Santoro et al., 2022] Santoro, A., Piras, F., Fiore, B., Frassinelli, N., Bazzurro, A., and Agnoletti, M. (2022). The role of trees outside forests in the cultural landscape of the colline del prosecco unesco site. *Forests*, 13(4).

[Schelhaas et al., 2018] Schelhaas, M.-J., Hengeveld, G. M., Heidema, N., Thürig, E., Rohner, B., Vacchiano, G., Vayreda, J., Redmond, J., Socha, J., Fridman, J., et al. (2018). Species-specific, pan-european diameter increment models based on data of 2.3 million trees. *Forest Ecosystems*, 5:1–19.

[Sintégra Géomètres Experts, 2019] Sintégra Géomètres Experts (2019). Rapport technique lidar luxembourg.

[Thompson et al., 2021] Thompson, J. C., Wright, D. K., Ivory, S. J., Choi, J.-H., Nightingale, S., Mackay, A., Schilt, F., Otárola-Castillo, E., Mercader, J., Forman, S. L., et al. (2021). Early human impacts and ecosystem reorganization in southern-central africa. *Science Advances*, 7(19):eabf9776.

[United States Forest Service, 2019] United States Forest Service (2019). DBH Measurement Guide. https://conservationgateway.org/ConservationPractices/cities/hthc/library/Documents/Manual%20and%20Appendices/Field%20Work%20Support%20Documents/DBH%20Measurement%20Guide.pdf. 26.04.2024.

[Wang et al., 2021] Wang, D., Liang, X., Mofack, G. I., and Martin-Ducup, O. (2021). Individual tree extraction from terrestrial laser scanning data via graph pathing. *Forest Ecosystems*, 8:1–11.

[World Bank Group, 2024] World Bank Group (2024). Country of luxembourg overview page. https://data.worldbank.org/country/luxembourg.

[Xu et al., 2021] Xu, D., Wang, H., Xu, W., Luan, Z., and Xu, X. (2021). Lidar applications to estimate forest biomass at individual tree scale: Opportunities, challenges and future perspectives. *Forests*, 12:550.

[Yao et al., 2012] Yao, W., Krzystek, P., and Heurich, M. (2012). Tree species classification and estimation of stem volume and dbh based on single tree extraction by exploiting airborne full-waveform lidar data. *Remote Sensing of Environment*, 123:368–380.

[Yu et al., 2004] Yu, X., Hyyppä, J., Hyyppä, H., and Maltamo, M. (2004). Effects of flight altitude on tree height estimation using airborne laser scanning. *Proceedings of the Laser Scanners for Forest and Landscape Assessment–Instruments, Processing Methods and Applications*, pages 02–06.

[Zhang et al., 2023] Zhang, Z., Wang, T., Skidmore, A. K., Cao, F., She, G., and Cao, L. (2023). An improved area-based approach for estimating plot-level tree dbh from airborne lidar data. *Forest Ecosystems*, 10:100089.

[Zhao et al., 2023] Zhao, H., Morgenroth, J., Pearse, G., and Schindler, J. (2023). A systematic review of individual tree crown detection and delineation with convolutional neural networks (cnn). *Current Forestry Reports*, pages 1–22.

Appendix A

Dataset

Number	Species	DBH (cm)	Height (m)	Crown Area (m^2)
1	Aesculus hippocastanum	0.98	15	138.13925
2	Aesculus hippocastanum	0.78	15.36	80.80075
3	Taxus baccata	0.67	21.84	126.7199
4	Fagus sylvatica Purpurea	1.07	24.08	243.090583
5	Tilia cordata	0.87	15.36	129.575558
6	Acer pseudoplatanus	1.1	23.24	238.5906
7	Fagus sylvatica	0.76	19.5	254.691632
8	Tilia sp.	0.86	15.1	191.39715
9	Tilia sp.	1.97	12	42.3978
10	Ulmus glabra 'Pendula'	0.57	6.2	61.874618
11	Fagus sylvatica 'Purpurea'	0.96	30.9	442.252187
12	Quercus sp.	1.62	20.7	398.298377
13	Sorbus domestica	0.89	16.2	97.363208
14	Tilia platyphyllos	0.91	23	167.5037
15	Fagus sylvatica Purpurea	0.94	22.9	243.5036
16	Quercus sp.	0.89	13.1	111.31565
17	Crataegus monogyna	0.46	7.7	74.792118
18	Fraxinus excelsior	1.15	28.2	312.52175
19	Cedrus deodara	1.16	23.7	292.282316
20	Tilia sp.	2.06	7	32.25585
21	Quercus sp.	1.42	17	501.439209
22	Tilia cordata	1.15	30.8	179.685504
23	Populus balsamifera	0.64	23.3	169.72493
24	Tilia sp.	0.78	17.4	175.3066
25	Quercus robur	1.34	22.1	431.25685
26	Tilia platyphyllos	1.35	22.7	164.020974
27	Fraxinus excelsior	1.8	21	314.627274
28	Quercus sp.	0.98	14	219.995172
29	Crataegus monogyna	0.38	7.7	30.117078
30	Quercus sp.	1.18	18.6	163.41572
31	Quercus sp.	1.37	19.7	260.299649
32	Quercus robur	1.69	22.3	160.24355
33	Quercus robur	1.69	22.3	294.4499
34	Quercus sp.	1.6	21.1	273.559919
35	Salix alba 'sericea'	0.7	7.5	23.32825
36	Tilia sp.	1.09	22.4	201.5442
37	Quercus robur L.	1.23	19.1	362.014106
38	Salix caprea	1.78	6.3	69.844705
39	Fagus sylvatica x pendula	0.33	9.7	58.1409

40	Tilia sp.	2	20.5	288.846616
41	Fraxinus excelsior	1.06	19.7	246.809081
42	Tilia sp.	1.76	24	200.926761
43	Tilia sp	0.85	24.8	391.084184
44	Salix sp.	0.88	12.8	114.475349
45	Quercus sp.	1.35	21.8	297.477479
46	Quercus sp.	1.25	20.7	306.302578
47	Pyrus communis	1.22	12.7	209.017104
48	Quercus sp.	1.5	24.3	598.048397
49	Aesculus hippocastanum	0.92	21.3	189.076196
50	Salix alba	2	10	112.9901
51	Fagus sylvatica	1.48	20.7	182.246965
52	Acer pseudoplatanus	1.18	18.3	352.651842
53	Prunus avium	1.13	19.3	278.643012
54	Quercus sp.	1.81	23.4	38.11705
55	Quercus sp.	1.81	23.4	163.9141
56	Quercus sp.	1.16	21	300.71957
57	Quercus sp.	1.05	22	388.942957
58	Quercus sp.	1.11	20	408.571057
59	Salix sp.	0.9	16.7	110.85225
60	Aesculus hippocastanum	0.92	14.7	141.599794
61	Quercus sp.	1.12	19.6	108.200449
62	Fagus sylvatica 'Purpurea'	0.79	20.8	198.3592
63	Fagus sylvatica	0.84	21.8	308.146356
64	Quercus sp.	0.88	20.3	338.03345
65	Quercus sp.	1.02	21.7	314.201126
66	Quercus sp.	1.15	17.3	415.968049
67	Fagus sylvatica	0.88	19.5	72.060675
68	Salix caprea	1.21	3.4	19.721553
69	Fraxinus excelsior	1.13	17.7	268.621207
70	Quercus sp.	1.18	20	384.462248
71	Populus sp.	1.67	39	414.275106
72	Quercus sp.	1.32	13.5	378.858948
73	Tilia sp.	1.22	26.2	265.44353
74	Fraxinus excelsior	1.17	19.1	351.28665
75	Tilia sp.	0.71	15.3	305.981332
76	Abies pinsapo	1.47	21.4	151.7018
77	Salix alba 'tristis'	0.87	14.1	282.228589
78	Tilia sp.	1.44	28.5	407.01095
79	Tilia sp.	0.75	11	163.062995
80	Ulmus minor	1.43	29.2	284.812984
81	Tilia sp.	1.31	13.3	121.153513
82	Tilia sp.	0.48	9.2	73.760699
83	Tilia sp.	1.17	20.8	260.039507
84	Fagus sylvatica sp.	1.38	20.2	243.08625
85	Quercus sp.	1.48	16.5	203.63995
86	Platanus acerfolia	2.07	31.9	768.46295
87	Tilia sp.	0.86	14.7	172.619198
88	Tilia sp.	0.68	16.8	104.6169
89	Pyrus sp.	0.25	4.7	57.73593
90	Salix alba 'tristis'	0.7	15.1	268.615864
91	Aesculus hippocastanum	0.98	15.3	125.0409
92	Tilia sp.	0.59	13.9	211.1581

93	Pyrus sp.	0.78	12.8	114.92815
94	Fagus sylvatica	1.09	24.5	258.379814
95	Thuja occidentalis	0.72	24.6	28.5997
96	Thuja occidentalis	1.09	25.5	42.1784
97	Gingko biloba	0.96	20.4	165.869689
98	Sequoia giganteum	1.96	25.3	164.66865
99	Tilia sp.	1.37	19.6	182.594702
100	Quercus robur L.	1.72	22.4	480.024043
101	Tilia sp.	0.62	16.2	185.470795
102	Fraxinus excelsior	1.38	25	474.751367
103	Tilia cordata	1.47	6	78.562
104	Pyrus communis	0.76	12.6	532.851993
105	Aesculus hippocastanum	1.02	15.4	254.086098
106	Fagus sylvatica 'purpurea'	1.3	17.4	244.68753
107	Acer pseudoplatanus	0.73	21.9	214.31805
108	Juglans regia	1.32	21.1	277.131484
109	Aesculus hippocastanum	1.42	21.7	52.13165
110	Aesculus hippocastanum	1.42	21.7	103.962372
111	Sequoiadendron giganteum	1.78	36.5	174.77045
112	Sequoiadendron giganteum	2.85	36.8	113.375366
113	Tilia sp.	1.22	13.6	138.2251
114	Tilia sp.	1.04	14.7	155.9575
115	Tilia sp.	1.14	19.3	130.938029
116	Fagus sylvatica	1.57	22.7	442.50367
117	Aesculus hippocastanum	1.37	23.5	376.276688
118	Aesculus hippocastanum	0.81	18.1	71.9771
119	Aesculus hippocastanum	0.81	18.1	140.85505
120	Fagus sylvatica	1.22	20.3	275.781419
121	Gingko biloba	1.46	23.1	193.709946
122	Quercus robur	1.1	26	546.292491
123	Acer pseudoplatanus	0.8	22.3	250.637002
124	Carpinus betulus	0.63	13.4	140.493173
125	Tilia sp	2.54	16.8	278.517605
126	Salix sp.	0.94	8.3	142.767397
127	Aesculus hippocastanum	1.19	15.5	132.04479
128	Tilia platyphyllos	1.15	17.7	178.93535
129	Juglans regia	0.73	9	313.560659
130	Fagus sylvatica sp.	1.27	23.1	322.783801
131	Tilia cordatta	1.37	12.5	71.78545
132	Quercus sp.	1.52	27.5	546.4985
133	Tilia sp.	1.53	27.4	135.949834
134	Aesculus hippocastanum	1.15	15.2	222.131117
135	Cornus sp.	0.75	13.2	160.3412
136	Tilia sp.	0.78	18.1	142.962688
137	Sorbus domestica	0.51	15.2	143.823819
138	Sorbus domestica	0.87	15.3	141.495252
139	Salix alba	1.73	14.5	138.033438
140	Fagus sylvatica 'Purpurea'	1.24	28	421.7183
141	Tilia sp.	1.34	22.3	74.32451
142	Quercus robur L.	1.61	12.5	79.20384
143	Tilia sp.	0.48	21.5	38.30008
144	Quercus sp.	1.04	16.4	372.622318
145	Tilia platyphyllos	1.28	15.1	357.652124

146	Quercus sp.	1.25	20.2	261.663789
147	Sequoia sempervirens	1.05	39.3	150.547
148	Tilia platyphyllus	1.2	28.9	188.57255
149	Sequoiadendron giganteum	2.03	42.4	334.725195
150	Tilia sp.	1.12	21.2	482.138922
151	Cedrus atlantica	1.53	32.8	121.37716
152	Calocedrus decurrens	1.4	33.7	164.300001
153	Tilia platyphyllus	1.2	28.9	84.8773
154	Juglans regia	1.32	20.2	389.012107
155	Aesculus hippocastanum	1.29	16.6	243.05885
156	Quercus sp.	1.24	17.9	253.242159
157	Quercus sp.	1.39	23.5	489.208768
158	Fagus sylvatica Purpurea	1.31	28.5	472.127182
159	Quercus robur	1.05	14.7	281.942926
160	Quercus robur L.	1.65	26.2	427.284835
162	Aesculus hippocastanum	1.09	17.2	206.842241
163	Ginkgo biloba	1.21	24.5	188.648426
164	Quercus sp.	1.47	25	539.448656
165	Sorbus domestica	1.08	18.4	191.858063
166	Quercus robur L.	1.81	21.7	507.934826
167	Castanea sativa	1.71	17.9	240.567759
168	Pyrus sp.	1.41	12.3	200.9673
169	Crataegus sp.	0.56	19.4	411.419137
170	Populus nigra L.	1.24	16.3	208.213407
171	Quercus robur	0.95	15.5	342.38085
172	Quercus sp.	1.07	18.8	361.136914
173	Quercus sp.	2.07	15.2	233.535509
174	Castanea sativa	1.35	21.4	101.5759
175	Prunus avium	0.71	12.2	77.249502
176	Quercus petraea	1.29	20.5	208.10815
177	Quercus robur	1.09	20.3	468.978224
178	Sequioadendron giganteum	0.96	10.7	67.879563
179	Aesculus hippocastanum	1.12	11.5	123.897772
180	Quercus robur	1.3	19.4	318.74915
181	Aesculus hippocastanum	1.09	14.5	246.33685
182	Quercus sp.	1.13	15.9	298.016024
183	Tilia sp.	1.12	24	266.7779
184	Aesculus carnea	1.11	14.9	126.83041
185	Aesculus hippocastanum	1.12	19.3	183.864504
186	Quercus sp.	1.51	15.3	261.01844
187	Prunus avium	0.5	7.91	67.6155
188	Prunus avium	0.59	11.57	105.83345
189	Juglans regia	0.6	12.1	151.1087
190	Tilia sp.	0.38	8.92	48.06485
191	Juglans regia	0.23	4.85	18.75155
192	Juglans regia	0.26	7.32	39.6255
193	Juglans regia	0.18	4.79	4.309
194	Juglans regia	0.44	9.32	83.70615
195	Prunus avium	0.44	8.18	39.0416
196	Quercus sp.	1.25	19.89	569.72175
197	Quercus sp.	1.15	12.49	219.687794
198	Acer campestre	0.58	8.58	89.47439
199	Quercus sp.	0.56	13.67	151.6809

200	Juglans regia	0.33	12.87	32.3977
202	Juglans regia	0.61	13.26	176.708984
203	Fraxinus excelsior	0.95	16.87	351.261545
204	Acer campestre	0.22	10.6	37.3308
205	Pyrus sp.	0.43	6.58	25.07815
206	Fraxinus excelsior	0.61	11.72	96.96545
207	Pyrus sp.	0.32	10.49	37.7429
208	Betula	0.35	16.08	78.60645
209	Betula	0.51	16.93	67.7638
210	Tilia sp.	0.37	10.55	70.6587
211	Acer campestre	0.22	6.43	19.4487
212	Acer campestre	0.31	8.53	30.50095
213	Acer campestre	0.25	9.66	11.55085
214	Acer campestre	0.21	9.95	12.87165
215	Acer campestre	0.31	13.51	43.6926
216	Quercus sp.	0.57	13.66	123.734
217	Juglans regia	0.18	5.5	12.5226
218	Juglans regia	0.76	14.26	264.6993
219	Tilia sp.	0.41	11.36	50.2811
220	Tilia sp.	0.28	8.14	26.58705
221	Juglans regia	0.55	12.39	137.14305
222	Tilia sp.	0.32	9.14	35.63715
223	Tilia sp.	0.32	8.07	33.1867
224	Tilia sp.	0.37	10.15	48.83205
225	Quercus sp.	1.28	23.21	398.6031
226	Quercus sp.	0.74	14.28	153.038418
227	Quercus sp.	1.22	21.06	650.601424
228	Populus sp.	0.7	15.2	80.497318
229	Pyrus sp.	0.57	10.53	79.859213
230	Pyrus sp.	0.55	9.61	48.7575
231	Pyrus sp.	0.43	8.94	55.9067
232	Populus sp.	0.51	7.16	41.91885
233	Salix alba	0.65	7.96	25.97065
234	Quercus sp.	1.16	17.03	317.802512
235	Prunus avium	0.28	7.52	17.351598
236	Tilia sp.	0.44	7.83	47.2564
237	Salix alba	0.48	11.13	109.354635
238	Salix alba	0.49	10.64	94.902779
239	Quercus sp.	0.26	9.54	21.84445
240	Quercus sp.	0.37	10.11	76.95835

Table A.1: Combined tree dataset

Appendix B

Code: Select Zip Files For Download

```
import geopandas as gpd

# Load the shapefile mask to cut out pointcloud
trees = gpd.read_file('path_to_pointcloud_mask')

#Load the lidar tiles shapefile
tiles = gpd.read_file('path_to_masklayer')

# Perform the intersection
intersection = gpd.overlay(trees, tiles, how='intersection')

# Filter the DataFrame to only include rows where the geometry is not empty
intersection = intersection[intersection.geometry.notnull()]

# Get the 'zip_file' attribute, convert to lowercase,
replace underscores with hyphens
intersection['zip_file'] = intersection['zip_file'].str.lower().str
.replace('_', '-')

# Write the 'zip_file' attribute of intersecting shapes to a text file
intersection['zip_file'].to_csv('download_lidar.txt', index=False)
print(f'successfully created text file')
```

Appendix C

Code: Laz File Indexer

```
import json
import glob
import pdal
import time
import pickle
from shapely.geometry import shape, mapping, box
#Count time from beginning
start_time=time.time()
# Get a list of all .laz files in the directory
laz_files = glob.glob("path_to_laz_files\\*.laz")
# Create a dictionary to store the spatial extents of each .laz file
laz_extents = {}
# Loop through each .laz file and get its spatial extent
for laz_file in laz_files:
    pipeline = pdal.Pipeline(json.dumps({
        "pipeline": [
            {
                "type": "readers.las",
                "filename": laz_file
            },
            {
                "type": "filters.info"
            }
        ]
    }))
    pipeline.execute()
    print("indexed extents for",laz_file)
    metadata = pipeline.metadata  # Access the metadata directly
    bounds = metadata['metadata']['readers.las']
    laz_extents[laz_file] = box(bounds['minx'], bounds['miny'],
    bounds['maxx'], bounds['maxy'])
# Save laz_extents to a file
with open('laz_extents.pkl', 'wb') as f:
    pickle.dump(laz_extents, f)
print(f"created pickle file with indexes")
# Calculate time
indexing_time = time.time() - start_time

print(f"done indexing. elapsed time: {indexing_time} seconds")
```

Appendix D

Code: creating vertical profile images

```
import os
import json
import fiona
import pdal
import pickle
import time
import pandas as pd
import pyvista as pv
import matplotlib.pyplot as plt
from shapely.geometry import shape, mapping, box
import multiprocessing as mp

# Define parameters
laz_extents_file = 'laz_extents.pkl'
shapefile_path = "path_to_shapefiles\\*.shp"
raster_output_directory = "path_to_raster\\"
camera_zoom = 1.5  #the camera zoom is a function
of the cloud.length multiplied by this value

#Count time from beginning
start_time=time.time()

with open(laz_extents_file, 'rb') as f:
    laz_extents = pickle.load(f)

print(f"imported extents from pickle file")

# Define a function to process a single feature
def process_feature(feature):
    # Access the "treen" field value
    treen_value = feature["properties"]["treen"]
    # Convert the geometry to a JSON-serializable format
    geometry = shape(feature["geometry"])
    json_geometry = mapping(geometry)
    print("mapped", treen_value,"to json geometry")
    # Find the .laz files whose extents intersect with
    the current polygon
    intersecting_files = [laz_file for laz_file,
```

```
extent in laz_extents.items() if extent.intersects(geometry)]

# Initialize an empty list to store the dataframes
df_list = []

# Loop through each intersecting .laz file
for j, laz_file in enumerate(intersecting_files):
    # Create a PDAL pipeline for this feature
    pipeline = {
        "pipeline": [
            {
                "type": "readers.las",
                "filename": laz_file
            },
            {
                "type": "filters.crop",
                "polygon": json.dumps(json_geometry)
            }
        ]
    }

    # Run the pipeline
    pipeline = pdal.Pipeline(json.dumps(pipeline))
    pipeline.execute()

    # Get the data as a numpy array
    data = pipeline.arrays[0]

    # Convert the numpy array to a pandas DataFrame
    #and append it to the list
    df_list.append(pd.DataFrame(data))

# Concatenate all dataframes in the list into one dataframe
df = pd.concat(df_list, ignore_index=True)

print("converted", treen_value, "to numpy array"
"and stored in memory")

# Convert the DataFrame to a numpy array
points = df[['X', 'Y', 'Z']].values
colors = df[['Red', 'Green', 'Blue']].values / 65535
# Normalize color values

# Create a PyVista point cloud object
cloud = pv.PolyData(points)
cloud['Colors'] = colors  # Add color information to
#the point cloud

# Create a plotter object and add the point cloud to it
plotter = pv.Plotter(off_screen=True)
plotter.add_mesh(cloud, point_size=3.0, rgb=True,
scalars='Colors', show_scalar_bar=False)
```

```
    # Set up a side view camera
    camera_position = list(cloud.center)
    camera_position[0] += cloud.length * camera_zoom
    focal_point = list(cloud.center)
    view_up = [0, 0, 1]
    plotter.camera_position = [camera_position, focal_point
    , view_up]

    # Render the view to an image
    img = plotter.screenshot(return_img=True)

    #access the diameter at breast height (DBH)
    DBH = feature["properties"]["BHD"]
    # Save the image to a PNG file
    # Generate a unique filename by appending "_n" to the DBH
    filename = f"{DBH}_{treen_value}.png"
    file_path = os.path.join(raster_output_directory, filename)

    # Save the image to a PNG file
    plt.imsave(file_path, img)
    print("saved image file for object", treen_value)

if __name__ == '__main__':
    # Open the shapefile
    with fiona.open(shapefile_path, "r") as shapefile:
        # Create a multiprocessing Pool
        with mp.Pool() as pool:
            # Use the Pool's map method to process all features
            pool.map(process_feature, shapefile)

    # Calculate total time
    total_time = time.time() - start_time

    print(f"done generating images. total runtime was {total_time}"
    "seconds.")
```

Appendix E

Code: Training (or fitting) the CNN model

```
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
import numpy as np
import glob
import cv2

from matplotlib import pyplot as plt
image_paths = glob.glob(r'path_to_images\*.png')

images = []
diameters = []

for image_path in image_paths:
    img = cv2.imread(image_path)
    img = cv2.resize(img, (224, 224))  # resize
    img = img / 255.0  # normalize pixel values
    images.append(img) #add to list
for image_path in image_paths:
    filename = image_path.split('\\')[-1]  # get filename from path
    diameter_str = filename.split('_')[0]  # split at underscore
    diameter = float(diameter_str) * 100
    # convert diameter to float and convert to cm
    diameters.append(diameter)

images = np.array(images)
diameters = np.array(diameters)

X_train, X_test, y_train, y_test = train_test_split(images, diameters,
test_size=0.2, random_state=52)

base_model = ResNet50(weights='imagenet', include_top=False)
```

```
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.2)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.2)(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.2)(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.2)(x)
predictions = Dense(1, activation='linear')(x)

model = Model(inputs=base_model.input, outputs=predictions)

model.compile(optimizer=Adam(learning_rate=0.000004),
loss='mean_squared_error')

early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)

hist = model.fit(X_train, y_train, epochs=500,
validation_data=(X_test, y_test), callbacks=[early_stopping])

fig = plt.figure()
plt.plot(hist.history['loss'], color='teal', label='loss')
plt.plot(hist.history['val_loss'], color='orange', label='val_loss')
fig.suptitle('Loss', fontsize=20)
plt.legend(loc="upper left")
plt.show()
model.save('dbh_rename.h5')
```

Appendix F

Code:Multiple Linear Regression

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

# Load spreadsheet
xl = pd.ExcelFile('path_to_excelfile.xlsx')

# Load a sheet into a DataFrame by name
df = xl.parse('Sheet5')

# Convert the 'shapearea' and 'Height' columns to numeric,
coercing non-numeric values to NaN
df['shapearea'] = pd.to_numeric(df['shapearea'], errors='coerce')
df['Height'] = pd.to_numeric(df['Height'], errors='coerce')
df['BHD'] = pd.to_numeric(df['BHD'], errors='coerce')

# Drop the rows where either 'shapearea', 'Height' or 'BHD' is NaN
df = df.dropna(subset=['shapearea', 'Height', 'BHD'])


# Define the dependent variable
Y = df['BHD']

# Define the independent variables
X = df[['shapearea', 'Height']]

rmse_values = []

# Cross-validate the model 50 times with different train-test samples
for _ in range(50):
    # Split the data into training and validation sets
    X_train, X_test, Y_train, Y_test =
    train_test_split(X, Y, test_size=0.2)

    # Print the statistics for the training and test data
    print(f"Training data: {X_train.shape[0]} rows")
    print(f"Test data: {X_test.shape[0]} rows")
```

```
    # Create a linear regression model
    model = LinearRegression()

    # Train the model
    model.fit(X_train, Y_train)

    # Use the model to make predictions
    Y_pred = model.predict(X_test)

    # Calculate and store the RMSE
    rmse = np.sqrt(metrics.mean_squared_error(Y_test, Y_pred))
    rmse_values.append(rmse)


# Print the average RMSE
print('Average RMSE: %.2f' % np.mean(rmse_values))


# Calculate and print the mean and standard deviation for the
training and validation data
print(f"Mean of training data: {X_train.mean()}")
print(f"Standard deviation of training data: {X_train.std()}")
print(f"Mean of validation data: {X_test.mean()}")
print(f"Standard deviation of validation data: {X_test.std()}")

# Use the model to make predictions for the whole dataset
Y_pred_all = model.predict(X)

# Create a scatter plot of the real vs predicted values
plt.figure(figsize=(10, 6))
plt.scatter(Y, Y_pred_all)
plt.xlabel('Measured DBH')
plt.ylabel('Estimated DBH')

# Calculate the RMSE for the whole dataset
rmse_all = np.sqrt(metrics.mean_squared_error(Y, Y_pred_all))

# Display the RMSE and N values in the plot
plt.text(0.05, 0.95, f'RMSE: {rmse_all:.2f},
N: {len(Y)}', transform=plt.gca().transAxes, verticalalignment='top')

plt.show()
```

Eidesstattliche Erklärung

Ich versichere:

- dass ich die Masterarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- dass alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Publikationen entnommen sind, als solche kenntlich gemacht sind.
- dass ich dieses Masterarbeitsthema bisher weder im In- noch im Ausland (einer Beurteilerin/ einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

DATUM

UNTERSCHRIFT