Skoltech
Skolkovo Institute of Science and Technology

Skolkovo Institute of Science and Technology

A deep learning framework for mixed dense forests parameter estimation at individual tree scale

Doctoral Thesis
by

Ivan Dubrovin

Doctoral Program in Engineering Systems

Supervisor
Professor, Clement Fortin

Moscow – 2024

A deep learning framework for mixed dense forests parameter estimation at individual tree scale

by

Ivan Dubrovin

Saturday 28th December, 2024 16:58

Submitted to the Center for Digital Engineering
on October 2024, in partial fulfillment of the requirements for the
Doctoral Program in Engineering Systems

Abstract

Detailed forest inventories are necessary for almost all modern activity related to managing forests, from responsible harvest planning to modeling and regulating atmospheric CO_2. Remote sensing data is extensively used to facilitate conducting such inventories over extensive areas, which are unfeasible for traditional manual forest inventories. The current state of commercially available sensors allows for very detailed observations, unlocking the possibility of working on the scale of individual trees but requiring robust algorithms for detection and segmentation of individual trees in multimodal data. Such algorithms already exist for forests that are either sparse or predominantly coniferous, but robustly segmenting dense mixed forests with complex canopy structures remains an open problem. Targeting UAV-based LiDAR point clouds augmented with RGB orthophoto data as the main input source, I present my work on a framework that uses deep learning to segment point clouds over dense mixed forests into individual trees and post-process the segments with a collection of specialized classic machine learning models to predict the required forest attributes for each individual tree. I use original datasets of a detailed forest inventory covered by UAV LiDAR and RGB orthophoto surveys and a collection of manually extracted point clouds of individual trees collected in Perm Krai, Russia, to train the segmentation network and the attribute prediction models for classifying and regressing individual trees into the required parameters. The segmentation network is trained on patches of synthetic forest constructed from point clouds of individual trees, heavily relying on data augmentations to mitigate the drawbacks of a having a small dataset and make the synthetic forest look more realistic. The thesis contains a description of the design, implementation, and experimental verification of the proposed framework, as well as recommendations for further improvements. Both datasets were released into open access during the work on this thesis. The code is also open-access.

Publications

1. Ivan Dubrovin and Anton Ivanov. Remote Sensing Evidence for the Harmful Algal Bloom Explanation of the Ecological Situation in Kamchatka in Autumn of 2020. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 6764–6767, Kuala Lumpur, Malaysia, July 2022. IEEE. ISBN 978-1-66542-792-0. doi:10.1109/IGARSS46834.2022.9884841

2. Ivan Dubrovin and Clement Fortin. An Exploration of Properties of Point Clouds of Individual Trees Extracted From a Larger UAV Lidar Survey. In *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 4503–4506, Athens, Greece, July 2024. IEEE. ISBN 9798350360325. doi:10.1109/IGARSS53475.2024.10641061

3. Ivan Dubrovin, Clement Fortin, and Alexander Kedrov. An open dataset for individual tree detection in UAV LiDAR point clouds and RGB orthophotos in dense mixed forests. *Scientific Reports*, 14(1):21938, September 2024. ISSN 2045-2322. doi:10.1038/s41598-024-72669-5

Acknowledgments

As any large research project, finishing this thesis wouldn't be possible without the support of many people. I would like to thank some of them, whose help played the biggest roles in my successfully finishing this. First, my supervisor, professor Clement Fortin, who picked up the lead when the project was in a disarray and guided me through to the end, helping to prioritize and focus on what was significant. Second, my previous supervisor, professor Anton Ivanov, under whose supervision the project started and gradually evolved into what it finally became. Third, Alexander Kedrov and Albert Vasiliev from Space Technologies and Services Center, Ltd, who allowed me to work with their data and taught me about the inventories in the field and current approaches of extending them with remote sensing data used in industry. And last, but also the most important, is my beautiful patient wife, who was supportive and forgiving from the very first moments of my PhD studies, and without whose help I would have given up many times over. Thank you all, sincerely, for the guidance, the wisdom, the knowledge, the help, the support.

Contents

List of Figures

List of Tables

Glossary

ABA area-based approach. 27

CHM canopy height map. 34

dbh diameter at breast height. 30, 40, 41, 75, 76, 78, 81

EPSG European Petroleum Survey Group, EPSG Geodetic Parameter Dataset is a registry of spatial reference systems from the group. 40

GNSS global navigation satellite system. 40

ISRO Indian Space Research Organisation. 14

LiDAR Light Detection and Ranging. 13, 15–19, 22–24, 26, 27, 30, 31, 33–39, 41, 42, 45, 47, 48, 50, 59, 84, 85

MAE mean absolute error. 75, 76, 78

MLP multilayer perceptron. 26

MODIS Moderate Resolution Imaging Spectroradiometer. 13

NASA National Aeronautics and Space Administration. 13, 14

OLI Operation Land Imager. 13

RCNN Region-Based Convolutional Neural Network. 35

RGB red, green, blue: a true color image. 15–19, 22, 32, 37–39, 47–49, 51, 54, 84, 86

RMSE root mean squared error. 23, 75, 76, 78

RTK Real-Time Kinematic. 40

SAR synthetic aperture radar. 13, 14, 22, 23

SRTM Shuttle Radar Topography Mission, a near-global digital elevation model. 42

UAV unmanned aerial vehicle. 14–19, 22, 33, 35–39, 42, 47, 48, 84, 85

UTM Universal Transverse Mercator coordinate system. 40

VNIR visible and near-infrared. 23, 37

Chapter 1

Introduction

This chapter aims to give a general introduction to the research project and put it into wide scientific and societal context. It defines the main research question and the hypothesis, and gives a high-level overview of the proposed framework. It also provides the links to the original datasets and the code.

1.1 Context

Forests are a crucial part of the global ecosystem, both environmentally and economically. They cover a third of the land area, contain over 80% of terrestrial biodiversity, and somewhere around one-third of humanity depends on forests and forest products for their livelihoods [Aerts and Honnay, 2011, Sta, 2020]. Forests are an essential renewable natural resource and a huge, dynamic part of the global carbon cycle. Figure 1-1 offers a map of the global tree coverage based on data from Hansen et al. [2013] to highlight the extent of forests on the planet. Responsible management of forests allows using the resources efficiently and sustainably, preserving the biodiversity, and regulating atmospheric CO_2, which is becoming especially important as the anthropogenic climate change is ongoing and accelerating [Fahey et al., 2010, Forster et al., 2024]. This drives the need for accurate, detailed, up-to-date information about various forest attributes such as distributions of tree species, average heights and ages of trees, estimates of trunk diameter, timber volume, and above ground biomass, and others.

The traditional manual forest inventories that rely on people going out into the forest to count and measure trees will always remain the most reliable and accurate way to assess forest parameters [Burley et al., 2004]. However, they are also extremely labor-intensive and time-consuming, which makes them infeasible to cover extensive areas with sufficient detail, speed, and frequency. They will always be required for quality assessment and validation of any indirect method of estimating forest parameters through remote sensing, but better methods of estimation will help to reduce these the time and effort. This is especially relevant in countries where massive areas are covered by forests, such as Russia, Brazil, Canada, USA, and China, which are the top five countries for forest area according to Glo [2020]. For that reason, various remote sensing techniques are widely used to extend and extrapolate the traditional forest inventories. All sorts of data, from satellite and aerial imagery to very detailed terrestrial Light Detection and Ranging (LiDAR) surveys, are used in all sorts of applications that require mapping forest attributes. Some such applications are mentioned in the next chapter dedicated to reviewing the literature.

Many national space agencies even provide open access to satellite data that can be used for mapping forests. European Space Agency offers free and open access[1] to the Sentinel missions through its Copernicus Data Space Ecosystem platform, including C-band synthetic aperture radar (SAR) data from Sentinel-1 and medium-resolution multispectral data from Sentinel-2, both with global coverage. They also have an upcoming P-band SAR mission called BIOMASS, designed specifically to study global forest biomass and carbon cycles [Quegan et al., 2019], which further indicates the growing importance and interest in the topic of forest mapping. NASA provides free access to an abundance of satellite data through its Earthdata platform, Landsat mission with the Operation Land Imager (OLI) instrument and Terra mission with the Moderate Resolution Imaging Spectroradiometer (MODIS) instrument being the most relevant for forest mapping applications. The Indian Space Research Organisation, Brazilian National Institute for Space Research, National

[1] Although the access is not open for everyone equally, as I have observed silent bans of the accounts connecting from Russian IP addresses without any response to support inquires in the previous platform, and an outright IP ban in the current one.

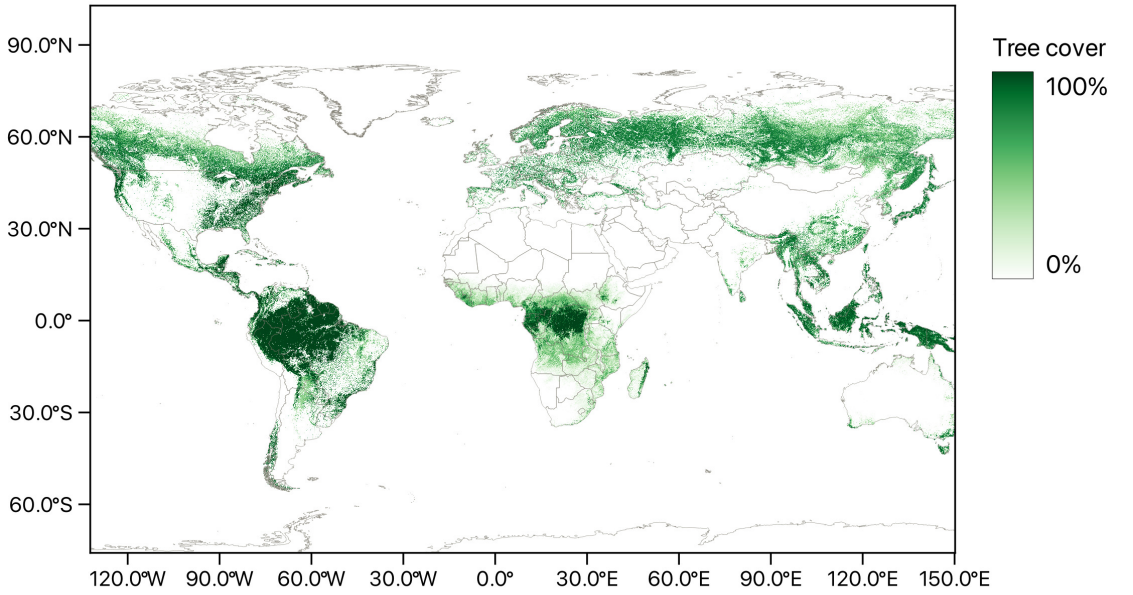


Figure 1-1: Global tree cover for 2010. Area occupied by tree canopies relative to the total area of the pixel. Data from Hansen et al. [2013]

Space Research and Development Agency of Nigeria all offer free satellite data that can be used for this purpose. NASA and ISRO also have a joint upcoming mission called NISAR with two fully-polarimetric SAR sensors at L-band and S-band [Kellogg et al., 2020], which will benefit forestry applications greatly.

The open satellite data is an essential tool for wide-area studies, but it usually has coarse resolution (tens to hundreds of meters per pixel), which limits the achievable accuracy and level of detail. It also offers no ability to control the observation parameters, as they are fixed by the instrument configuration and orbit parameters, both of which are outside the data consumer control. Higher resolution data with a limited ability to control the acquisition parameters is available commercially, but the prices are steep. Aerial observations with sensors mounted on planes or helicopters are both more controllable and cheaper alternatives, although still expensive and sill limited in the flexibility of the control of acquisition parameters. A much more affordable and controllable alternative is UAV-based observation. It allows for fine-grained control and on the fly adjustment of many important parameters such

as flight height, flight path overlap, combination of used sensors, and so on.

The most common way to use UAV remote sensing, be it LiDAR, multispectral, hyperspectral, or other data modalities, for mapping forest attributes in industry is what is known in the LiDAR community as the area-based approach, described in detail in Section 2.3. It is based on extrapolating measurements from ground plots made in traditional inventories by aggregating remote sensing data to the grid with cell area of a ground plot, which results in coarse resolution maps. It is easy to use, but its results are often not detailed enough when working on smaller scales. However, modern sensors and processing techniques allow working on the level of individual trees without any aggregation on data level. This is as detailed as a survey can possibly get, allowing for any level of aggregation for downstream tasks. This requires robust algorithms that allow detecting individual trees in dense multimodal data. The task is relatively[2] easy in urban environments, manually planted and managed forest stands, or forests that are either sparse or predominantly coniferous, where the structure of the canopy is easy to interpret. In some such environments, state-of-the-art results can be achieved by simple local maxima detection algorithms, that rely on the assumption that a tree can be detected by finding peaks of canopies because they correspond to tree tops. Forest that are mixed and dense, which are a huge part of forests in countries mentioned earlier, are much harder to work with and are a very active area of research for developing methods of detection of individual trees. The canopies in such forests are very complex, especially because the top of the crowns of deciduous tree species often do not have a single pronounced height maximum, and crowns of nearby trees often overlap.

The framework described in this thesis focuses on the fusion of two remote sensing data sources, UAV LiDAR point clouds and UAV RGB orthophotos, to detect individual trees in dense mixed forests and predict required attributes for each tree individually, producing the most detailed maps possible. The choice of data sources is driven by their complementary nature, which is key for semantically parsing such complex environments. LiDAR is an active sensor, which means it does not depend

[2]The word relatively does a lot of heavy lifting here, as the problem is by no means easy on its own and takes a lot of effort from many researchers to continue to make progress on.

on external conditions such as lighting. Cloud and terrain shadows, incidence angles, and weather conditions do not affect LiDAR surveys. Moreover, LiDAR provides 3D vertical structural information about the forest, as laser pulses penetrate the canopy and reach both the undergrowth and the ground. This structural information is essential for understanding complex environments like dense forests. High-resolution RGB imagery does depend on the illumination conditions, but is still an invaluable tool, as it offers detailed data with fixed resolution, continuous coverage of surfaces, unlike the discrete representation of laser scanning, and captures many fine details and textures. It can also benefit from a huge variety of well-established tools and processing techniques from the field of computer vision. Neither of these data sources on their own is enough to reliably and with sufficient robustness separate individual trees in dense mixed forests, and the key to success lies in their fusion.

According to the Design Research Methodology framework [Blessing and Chakrabarti, 2009], this research project can be classified as type 3: Development of Support. The Descriptive Study I is review-based, as understanding of the existing situation is obtained primarily from the literature review, and the most focus is on the comprehensive prescriptive study, followed by an initial Descriptive Study II to evaluate the results.

1.2 Research question and hypothesis

The main research question could be formulated as follows: "How to reduce effort and cost required for detailed inventories of dense mixed forests without losing accuracy?" The main hypothesis is then: "An accurate and detailed inventory with reduced effort and cost can be achieved through fusion of UAV LiDAR and RGB data using machine learning". The benchmarks to measure against are both the traditional manual forest inventory and the widely used area-based approach.

Cost and effort are closely connected, as any effort is in the end converted to cost. Still, it often makes sense to separate the two to highlight the nature of the effects the proposed system should have. The cost reduction was partly addressed in the previous section during the discussion of the platform of choice: UAV-based

remote sensing offers a great balance of upfront cost, effort to operate, and versatility in observation parameters. The effort reduction comes from greatly decreasing the amount of field inventory data required when using the proposed system, from the relative ease of collection of remote sensing data compared to alternatives, and from relative ease of tuning the system to operate in new areas.

1.3 Overview of the framework

The thesis describes a framework – details a possible way that a system for estimating forest parameters on the scale of individual trees can be built, with a relatively accessible amount of data and effort. Potential ways to create all the components a system like that needs are described, as well as the way they are assembled into a complete system. The heart of the proposed approach is the reduction of effort required for data collection for training a tree segmentation model. It is achieved by circumventing the need to create very labor-intensive datasets of fully segmented point clouds of forests where each point is assigned an ID of the tree it belongs to by using a significantly easier to create data of individual trees extracted from a larger point cloud of a forest. The individual trees are then arranged into synthetic forest patches, in which the per-point tree ID labels arise naturally from ordinal numbers and positioning of the trees within a patch, creating a reasonable alternative to manual segmentation.

The proposed framework consists of neural network-based tree segmentation in UAV LiDAR point clouds enhanced with RGB orthophoto-based features, with further processing of the segments using a collection of specialized classic machine learning models that predict the parameters of interest for each detected tree. The tree segmentation model is trained on synthetic forest patches constructed from a dataset of point clouds of individual trees extracted manually from a large UAV LiDAR survey, heavily relying on augmentations to make the synthetic forest closer to real forest. The parameter prediction classification and regression models are trained on the same dataset of individual trees, each specializing in a single parameter of interest that is available for the extracted tree – species, dbh, etc.

The most important parts of the framework are the approach to generation of data for training the tree segmentation network and the arrangement of the components into a single system. All other components can be swapped with different ones that serve a similar purpose. In fact, as mentioned in Section 5.1, such replacement are encouraged to improve the system. For example, the example implementation described in the thesis uses PointNet++ as the backbone for the segmentation network, but any other architecture that can operate directly on point clouds and output per-point predictions can be used instead. Using a more modern and powerful architecture will be beneficial, as the PointNet++ was chosen for ease of implementation and experimentation. Similarly, the specialized parameter prediction models in the example implementation are Random Forests trained using a collection of widely used manual point cloud features. Models trained on better feature sets, or any other models that can reduce a point cloud to a single prediction can be used instead, even neural network ones, even though the source dataset will probably need to be considerably larger for training even a small network.

Figure 1-2 is a schematic representation of the framework, showing the required inputs in red, processing steps in yellow, and artifacts in cyan. The field inventory is phased out, but it is still required, as application of any framework requires quality assessment and validation. The system consists of a trained tree segmentation neural network and a collection of trained attribute prediction models. The inputs for the framework are a UAV LiDAR point cloud and an RGB orthophoto. The point cloud is preprocessed by removing duplicates and noise points, classifying the ground points, and normalizing height. The orthophoto is processed by a feature extractor that encodes context information, and the planar coordinates of points of the point cloud are used to sample the orthophoto to combine both data sources into one. The tree segmentation neural network is then applied to the augmented point cloud to separate it into individual trees. The segments are then processed by a collection of specialized models, each predicting a single parameter of interest, like tree species, diameter at breast height, or others. The overall results are both the map with locations of individual trees, that can be created by reducing the segments into single planar points, and the required predictions of the parameters for each of the

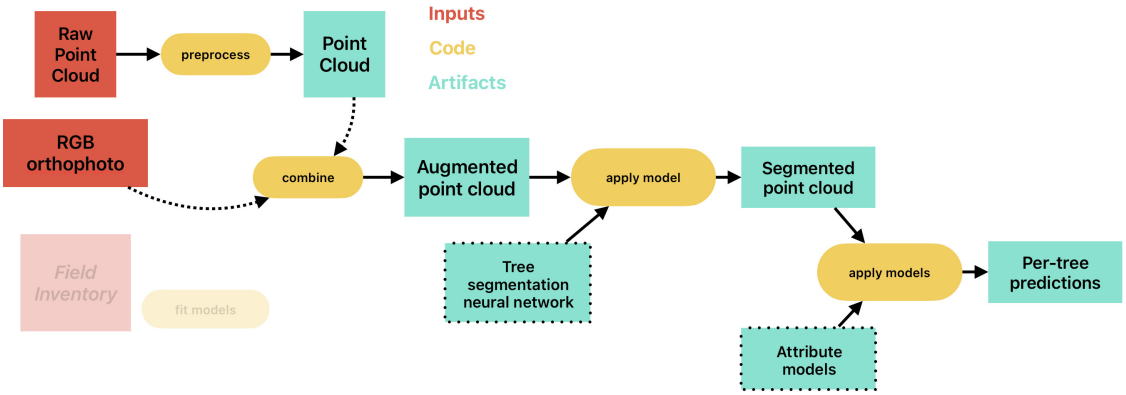


Figure 1-2: The schematic representation of the framework in the application stage. The required inputs are a UAV LiDAR point cloud and an RGB orthophoto used to augment the point cloud. The augmented point cloud is segmented into individual trees by the segmentation neural network. The segments are then processed by a set of specialized tree parameter prediction models.

trees.

Figure 1-3 is a schematic of the preparation step for the framework. Each individual node is described in detail in Chapter 3. To create a working implementation, a manual forest inventory covered by a UAV LiDAR point cloud and an RGB orthophoto are required. The inputs are used to create a dataset of point clouds of individual trees – which is significantly easier than to fully segment the point cloud into individual trees. This dataset is then used for generating synthetic forest patches for training the neural network and for training the tree parameter prediction models. The trained models are then used during inference as described in the previous paragraph.

1.4 Thesis Structure

Chapter 1 - Introduction The Introduction chapter aims to give a general introduction to the research project and put it into wide scientific and societal context. It defines the main research question and the hypothesis, and gives a high-level overview of the proposed framework. It also provides the links to the original datasets and the code.

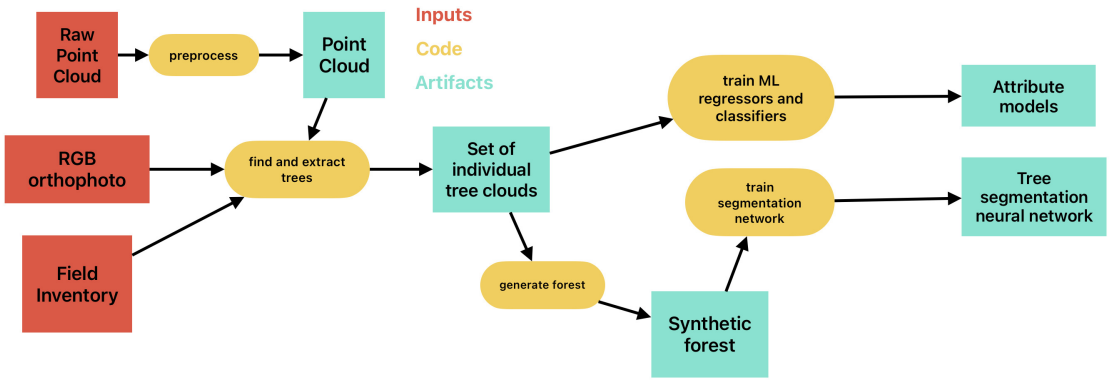


Figure 1-3: The schematic representation of the framework in the preparation stage. The required inputs are a manual forest inventory and a UAV LiDAR point cloud and an RGB orthophoto that cover it. The inputs are used to create a dataset of individual tree point clouds used for generation of synthetic forest patches for training the segmentation network and for training the parameter prediction models directly. The resulting trained models are then used for inference as shown in Figure 1-2.

Chapter 2 - Literature Review The Literature Review chapter aims to give an overview of the scientific literature on topics most relevant to the project. Its main goals are to provide the reader with context for the research described in the thesis, provide references for in-depth materials on topics that are out of scope of this work, and to highlight the research gap that the work tries to address.

Chapter 3 - Materials and methods The Materials and Methods chapter describes in detail the datasets, methods and methodological choices used in the proposed framework. Its aim is to make the work reproducible and to explain the methodological choices made.

Chapter 4 - Results The Results chapter describes the results of each stage of the framework preparation and the validation approach used to verify its applicability and effectiveness on realistic data.

Chapter 5 - Conclusion The Conclusion chapter offers a brief summary of the thesis as a whole, potential perspectives for further improvement of the proposed framework, and some concluding thoughts.

1.5 Data and code availability

Original datasets described in the thesis are openly available on Kaggle here and here. All the code used for the project is available on GitHub at iod-ine/phd. An HTML version of this thesis is hosted through GitHub Pages and is available here. The thesis document was developed using Quarto Allaire et al. [2024] using the literate programming approach Knuth [1984], and a link to an HTML version hosted through GitHub Pages is available in the repository. The deep learning part is implemented using PyTorch Ansel et al. [2024], PyTorch Geometric Fey and Lenssen [2019], and PyTorch Lightning Falcon and The PyTorch Lightning team [2019], with experiment tracking using MLflow. Classic machine learning models implementations are from scikit-learn Pedregosa et al. [2011]. NumPy Harris et al. [2020], SciPy Virtanen et al. [2020], pandas The pandas development team, scikit-image van der Walt et al. [2014] libraries are used for processing the data. ratserio Gillies et al. [2013/], geopandas, laspy, lazrs libraries are used for working with geospatial data formats. matplotlib Hunter [2007] and seaborn Waskom [2021] libraries are used for visualization.

Chapter 2

Literature review

This chapter provides an overview of the current state of literature on a variety of subjects related to the thesis. Its main goals are to provide the reader with context for the research described here and to highlight the research gap that the work tries to address. Where appropriate, references for in-depth materials on topics that are out of scope of this work are provided.

2.1 Remote sensing for forestry applications

As was mentioned in the introduction, remote sensing is widely used for extending labor-intensive and time-consuming manual forest inventories. This section provides examples of various remote sensing techniques used in various forestry applications, before going in more detail into specifically UAV LiDAR and RGB, which are the focus of the described framework.

Hansen et al. [2020] explore the usage of C-band SAR data from the Sentinel-1 mission for binary land use classification into forest/non-forest using a collection of classic machine learning classifiers. SAR is an active sensor, making it independent on external conditions and observation time, and it penetrates cloud cover, making it very reliable during almost any weather. They report accuracies from 80% in the worst case to the 93% in the best case, depending on the area of application.

Ferrari et al. [2023] use Fully Convolutional Networks [Long et al., 2015] for fusion of multispectral Sentinel-2 and C-band SAR Sentinel-1 data for clear-cut logging

detection in the presence of clouds, which limit the use of optical-only approaches and are very common in tropical areas. They show that fusion performs better than single-modality variant for pixels obscured by clouds. This combination of active SAR and passive multispectral data is very common, as the sensors compliment each other nicely, similarly to laser scanning and orthophotos.

Sinica-Sinavskis and Grube [2022] combine LiDAR point clouds with Sentinel-2 images to predict timber volume on a stand level. They use an unusual approach, using Sentinel-2 images for species detection by clustering, LiDAR point clouds for estimating tree counts and average tree heights, and combining them into two variables used to fit the final regression models. The reported relative root mean squared error (RMSE) values are 14-22%, with errors larger for deciduous tree species.

Illarionova et al. [2021] use multispectral satellite imagery for mapping dominant tree species. They utilize a hierarchical set of binary classification neural networks, first separating the image into forest/non-forest, then separating the forest into coniferous and deciduous, and finally separating deciduous forest into aspen and birch and coniferous into spruce and pine. The proposed approach achieved a significantly better F_1-score than straightforward multiclass classification: 0.84 versus 0.70.

Illarionova et al. [2022] use 5-meter resolution WorldView-2 VNIR data to predict canopy height with a convolutional neural network. They use airborne LiDAR data to create the canopy height map used as the target for training the model. They achieve a mean absolute error of 2.4 meters and show that the predicted canopy height map can be successfully used as an additional input for the tree species classification task.

LiDAR has been used for forestry applications for a long time, with publications on the topic dating back 40 years. Nelson et al. [1984] is one of the earliest studies that explores usage of airborne LiDAR for measuring forest canopy profiles and estimating tree heights and canopy closure (a measure of forest canopy coverage that indicates what proportion of the sky is obscured by the tree crowns when viewed from the ground). Nilsson [1996] is a study looking into tree height and timber volume estimation using airborne LiDAR across a range of point densities and seasons on a coniferous forest stand. Næsset [1997a] and Næsset [1997b] are

studies exploring the use of airborne LiDAR for estimating mean tree height and timber volume, suggesting the ways to correct the systematic underestimation of height and showing how regression on LiDAR-derived metrics can predict volume. Carson et al. [2004] offers an overview of some of the applications and approaches and a summary of contemporary state-of-the-art.

2.2 Machine learning and deep learning on point clouds

The reader is assumed to be familiar with general concepts of machine learning and deep learning. For an introduction or a refresher, one of the best resources is Goodfellow et al. [2016]. For a more detailed exploration, Wang et al. [2020] offer a selection of papers on topics relevant to modern deep learning techniques. As point clouds are a less well-known modality in machine learning and deep learning, the reader is also referred to Bello et al. [2020] and Guo et al. [2021], offering detailed reviews of the deep learning approaches used in various problems related to processing point clouds. Figure 2-1 is a high-level taxonomy of these approaches. Two main groups are structured grid-based, which rely on transformation of point clouds into regular structures that are then processed by 2D or 3D convolutional neural networks, and raw point cloud based, which consume point clouds directly. There are also more modern and powerful approaches, like the transformer-based Point Transformer [Zhao et al., 2021, Wu et al., 2022, 2024b] that utilizes the power of the attention mechanisms that dominates natural language applications and performs well in computer vision. Another novel addition is PointMamba [Liang et al., 2024], another architecture popularized in the natural language processing domain, aiming to improve on the transformer architecture by using a less computationally complex algorithm called state space model. The section is focused on providing a very short overview of these topics as it relates to the framework that is the focus of the thesis.

Classic machine learning approaches rely on feature engineering: manual preparation of features used as inputs for models based on domain expertise and various feature selection techniques. Two main groups of tasks are per-point predictions,

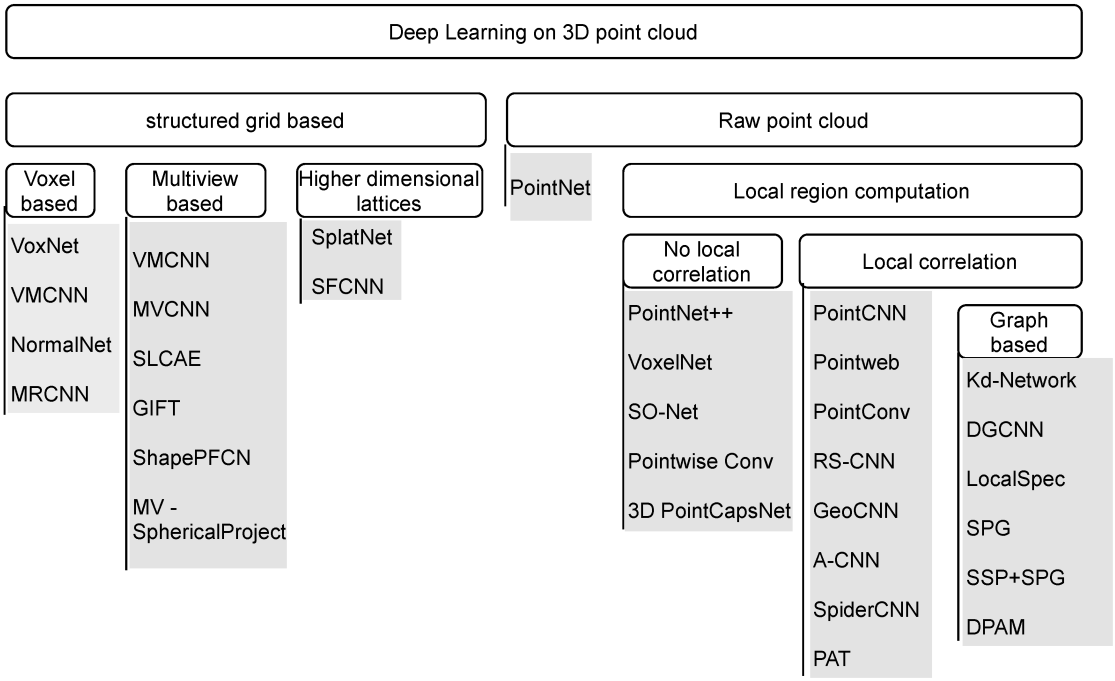


Figure 2-1: A high-level taxonomy of deep learning approaches for point clouds. Figure from Bello et al. (2020)

which is in many ways similar to the task of semantic segmentation of images, that requires per-point features, and per-cloud predictions that either process entire point clouds or individual segments, separated by some preprocessing routine. Weinmann et al. [2013] show that careful selection of features is crucial for accurate and efficient semantic interpretation of point cloud data: a shotgun approach of using many simple features without much consideration results in worse performance than a surgical approach of using a few carefully selected ones. They also provide definitions of some of the most used manual features that aim to describe the 3D structure of point sets. The features are based on combinations of eigenvalues of local covariance matrices of coordinate vectors of a set of points. They can be calculated on a per-point basis by using fixed-size or nearest-neighbor neighborhoods, or for whole segments of point clouds. The used features were originally introduced by West et al. [2004], Pauly et al. [2002], and Mallet et al. [2011], and include linearity, planarity, and scatter, aiming to indicate the presence of a linear, planar, or volumetric structures, and also omnivariance, anisotropy, eigentropy, the sum of eigenvalues, and curvature.

The formulas for the features are provided in Section 3.5, where they are used for training parameter prediction models for segmented trees. Simpler features, that are especially often used in area-based approach, include various statistics describing height and reflection intensity distributions of points, like percentages of points above mean height, deciles of height, cumulative percentages of points below height deciles, and others. Since many LiDAR sensors can record multiple reflections from a single pulse, features that use the reflection numbers are also often used, i.e., relative return number and total number of returns.

Point clouds are by their nature irregular, and non-uniform point densities across the cloud often become a problem for both manual feature creation and representation learning as part of a deep learning model. Many researches explore ways to handle this non-uniformity. For example, Özdemir [2021] proposes a framework for semantic segmentation of photogrammetric point clouds in urban environments, the key components of which are voxel-grid filtering-based downsampling for equalizing the point density across the cloud, manual addition of geometric features in a nearest-neighbor neighborhood, and processing the result with a convolutional network for assigning labels to each point, followed by a post-processing step to upscale the labels back to original point cloud size.

The first deep learning model to work directly on point clouds without constructing any intermediate representation that can be processed by convolutional models is the PointNet introduced by Qi et al. [2017a]. Figure 2-2 shows its architecture. A critical aspect of any model that aims to operate directly on point clouds is permutation invariance, as point clouds are unordered sets, and shuffling the points does not change the point cloud semantically. PointNet achieves that invariance by using a combination of shared multilayer perceptrons (MLPs) to process point coordinates and features and max pooling for construction of global feature vector. That feature vector can then be used directly for point cloud classification tasks, or concatenated with point features and processed further by shared MLPs for per-point predictions.

Qi et al. [2017b] introduces PointNet++, aimed to address the main drawbacks of the original PointNet model, namely the use of only two scales for context encoding – per-point and global, which limits the ability of the model to respond to local

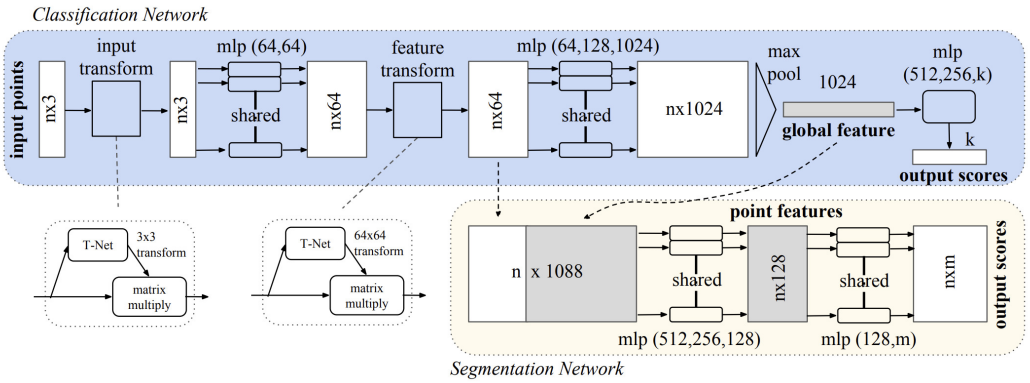


Figure 2-2: PointNet architecture. Figure from (Qi, Su, et al. 2017)

structures and fine patterns and thus work in complex scenes. To address this, PointNet++ uses a hierarchical architecture that applies PointNet recursively on nested subsets of the original point set, allowing to learn features on multiple increasing scales. To achieve that, network uses stacked set abstraction layers, that first sample a subset of the point cloud using farthest point sampling – an algorithm aimed to create representative subsets even for point clouds with uneven point density that selects the next point by maximizing its distance to already selected set – then constructs a neighborhood around each selected point using either fixed distance or fixed number of neighbors, and finally applies PointNet to each neighborhood to reduce into a feature vector for the sampled point. Then, similar to the original PointNet, the subset can be further reduced to a final global feature vector used for point clouds classification, or passed to an upscaling branch with skip-connections and k-nearest neighbors interpolation to upscale the features to the original point cloud coordinates. The architecture is visualized in Figure 2-3.

2.3 Area-based approach

The most common way to use LiDAR for mapping forest attributes is the area-based approach (ABA) [White et al., 2013]. Figure 2-4 shows its schematic representation. It consists of a LiDAR survey covering the whole area of interest and a manual forest inventory providing ground truth data for fitting statistical models and validating

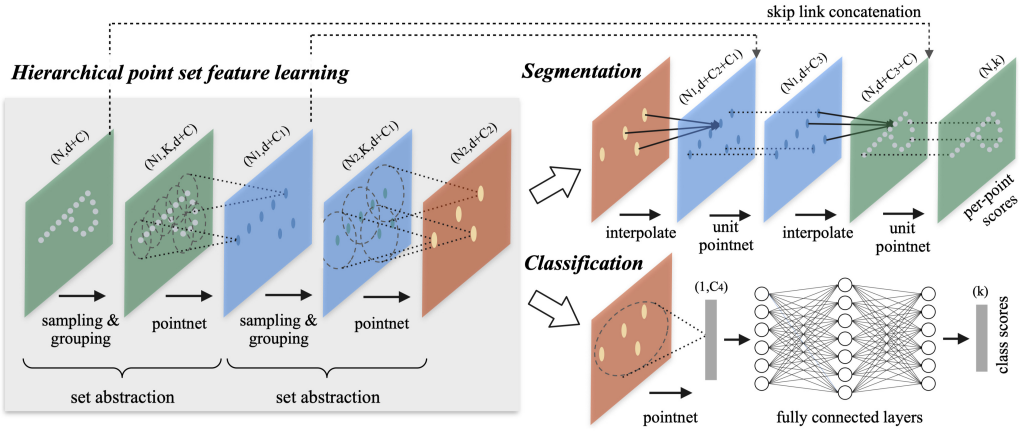


Figure 2-3: PointNet++ architecture. Figure from (Qi, Yi, et al. 2017)

the results. The inventory usually consists of many circular ground plots with every tree within counted and attributes of interest either directly measured or calculated and averaged. The point cloud is clipped by the extents of the ground plots, and for each plot it is reduced to a collection of manually selected metrics. The metrics usually include descriptions of the height distribution of the points, but often reflection intensities and other sensor-provided information is used as well, such as the return number, the number of returns, etc. (a brief discussion on the use of intensity-based features can be found in Section 3.1). In general, any summary statistic that can be derived from a collection of points can be used, including features mentioned in Section 2.2. These metrics are then used as input features for fitting regression and classification models to predict the forest attributes measured on the corresponding plots. The same metrics are calculated for the entire area of interest, using a grid with a cell size similar in area to the area of a single ground plot. The models are then applied to the grid, generating an extrapolation of the required attributes.

Area-based approach is extensively used both in research and in industry because it provides many advantages. It is relatively easy to implement. In fact, basic familiarity with the R programming language is enough to create your own area-based approach pipelines since a full, scalable implementation exists in the `lidR` package [Roussel et al., 2020]. It is also straightforward to extend with other data sources such as satellite or aerial images, and it works even with sparse data: for

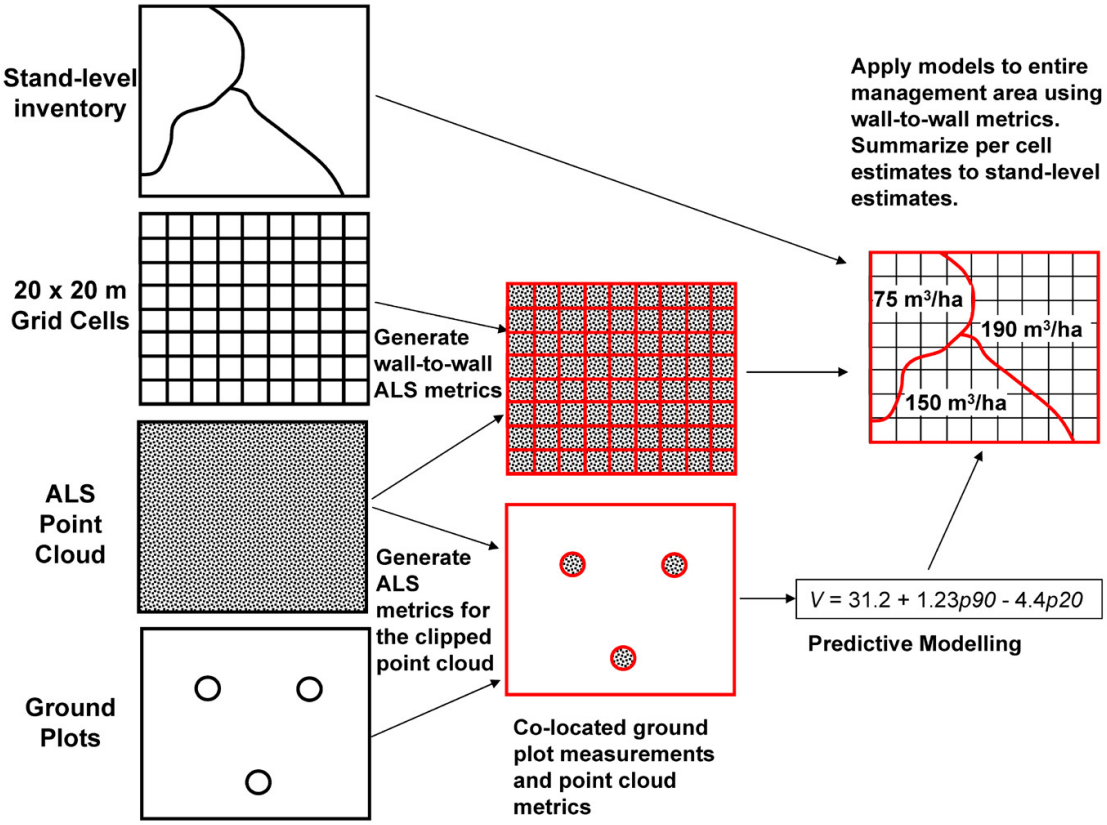


Figure 2-4: Area-based approach schematic. Figure from (White et al. 2013)

successful plot and stand level modeling point densities as low as 0.5 points per square meter have been reported to be enough [Treitz et al. [2012]; Jakubowski et al. [2013]]. Still, it requires a lot of field inventory data to work, since every ground plot becomes a single example for the models. The models that can be used are also relatively simple, because of how expensive the data collection is. Data-hungry approaches like neural networks usually do not have enough data to train. The results are also very coarse – predicted on a grid with the size defined by the area of a plot (a common plot shape is a circle with 9-meter radius, which is approximately equivalent to a square grid cell with 16-meter side). This is why they are usually further aggregated to stand level.

2.3.1 Examples of studies using ABA

As mentioned, area-based approach is used widely both throughout industry and research. I have personally taken part in a couple of projects where it was used for mapping forest attributes on large scales. This subsection offers some examples of its continuing use in research.

Bouvier et al. [2015] suggest a set of 4 metrics they use to fit models for predicting timber volume, above ground biomass, and basal-area on stand level, instead of most commonly used metrics based on the distribution of height. Their metrics are aimed to capture different aspects of the canopy geometry. The authors argue that usage of a very limited set of carefully engineered diverse metrics results in improvement of model generalization ability without loss of accuracy.

Zhang et al. [2023] use a modification of the area-based approach to predict plot-level diameter at breast height (dbh) by utilizing known allometric dependencies between tree height and dbh to limit the size of the hypothesis set for fitting regression models. They use airborne LiDAR measurements with an average density of 9.6 points per square meter and report a relative error of 11

Vermeer et al. [2023] use a U-Net [Ronneberger et al., 2015] image semantic segmentation model on LiDAR-derived 1-meter resolution digital terrain models and canopy height maps to predict the distribution of three main tree species in a Norwegian forest. They achieve a macro F_1 of 0.70 when including the background class,

and 0.63 when including only the target species classes, as evaluated on independent field inventory plots.

Kc et al. [2024] is a classic example of an area-based approach study utilizing airborne LiDAR survey to predict forest above ground biomass. They use a set of 32 metrics derived from height distribution of points, an automatic feature selection procedure, and two regression models: linear regression and Random Forest to compare the results. They report coefficients of determination of 0.85 and average errors around 83 tons per hectare.

2.4 Individual tree-based approach

With the constant improvement of accessibility and quality of high-resolution remote sensing data, there is a growing interest in the development of methods that operate on the scale of individual trees. This subsection gives an overview of some of the research in this area, split by the main data source.

Li et al. [2012] developed a well performing algorithmic method for segmentation of individual tree crowns in LiDAR point clouds for coniferous forests that relies on the point shape characteristic of many coniferous species and segments the trees from top to bottom.

Lucas et al. [2019] use the set of eigenvalue-based features calculated for each point in a fixed-radius neighborhood, with other geometrical features including maximum local height difference and height standard deviation, local radius and local point density, to identify linear vegetation elements in segmented point clouds. They also use two point-based features that do not rely on a neighborhood: the number of returns and the normalized return number, proposed in Guo et al. [2011], and use Random Forest classifier to separate the points that belong to vegetation after first removing the planar features corresponding to grass, soil, and water surfaces. The approach is somewhere in the middle between the area-based and individual tree-based, but still is a useful example of application of classic machine learning to segment out vegetation from larger point clouds.

2.4.1 Image-only

Many approaches rely only on images, mostly high-resolution RGB and multispectral ones, but lately also hyperspectral. The main advantage of such approaches is a well-established and well-known toolbox in terms of both algorithmic processing and deep learning, as many of the most important deep learning milestones were achieved in the field of computer vision. They also can rely on consistent resolution, capture of fine details and textures, and continuous representation of sensed environments. The main disadvantages were mentioned in the introduction: passive sensors rely on the sun as the source, and thus greatly depend on lighting such as cloud and terrain shadows, time of day, season. They also offer no information on vertical structure. Still, they are very popular and achieve outstanding results in many environments.

Weinstein et al. [2020] introduces a Python package for training and inference of ecological object detection neural networks in airborne imagery. It uses a convolutional object detection network described in Weinstein et al. [2019] to predict bounding boxes for individual trees. The data the model is trained on is not as dense as the forests that are the target of this thesis. Moreover, the model requires fine-tuning to be applicable to new data, which greatly limits its applicability, since developing bounding box annotations for individual trees within dense forests is an extremely tedious and labor-intensive task.

Lassalle et al. [2022] use high-resolution satellite imagery to delineate individual tree crowns in mangrove forests by using DeepLabv3+-based Multi-Task Encoder-Decoder network (MT-EDv3), originally proposed by La Rosa et al. [2021], to predict for each pixel the distance to the tree crown border. This distance map is then enhanced by applying the Laplacian over Gaussian filter, and finally the watershed segmentation algorithm is applied to delineate individual crowns. The approach seems to rely on there being semi-clear separation between the tree crowns, which is rarely the case when the forests are dense and mixed.

The same network was also successfully used to map tree species in tropical urban environment in Rio de Janeiro, Brazil in Martins et al. [2021]. They develop a post-processing approach to combine the semantic segmentation map and the distance map to classify tree species with an average F_1-score of 79.3% and resulting in a

realistic tree species map.

Osco et al. [2020] use a convolutional neural network on UAV multispectral images to count citrus trees in an orchard by predicting a confidence map that shows the likelihood of each pixel containing a tree. They report F_1-scores of up to 0.95, which is impressive even for managed stands.

Ventura et al. [2024] describe a network they call HR-SFAnet, consisting of a VGG-16 [Simonyan and Zisserman, 2014] backbone feature extractor, a confidence head, and a parallel attention head, for detecting individual trees in urban environments using high-resolution multispectral images. The network is a deeper modification of the SFANet proposed in Zhu et al. [2019] for counting the number of people in photos. They report F_1-scores of 0.75, with average positioning error of 2.2 meters.

2.4.2 Terrestrial LiDAR

Terrestrial LiDAR surveys usually provide very dense and detailed point clouds. Most importantly for the task of localizing individual trees, terrestrial measurements always capture the trunks of the trees clearly. Tree trunks are very useful for tree detection and segmentation, and there are many algorithms that use bottom-to-top approaches that trace the trunks into the canopies.

Burt et al. [2018] introduce `treeseg`, a software package written in C++ for extracting individual trees from LiDAR point clouds. Even though it is designed and presented as a platform-agnostic method made to operate on both terrestrial and UAV LiDAR point clouds, it relies on trunk detection, which is common for many methods based on terrestrial LiDAR and often is not applicable to UAV LiDAR data.

López Serrano et al. [2022] introduce another software package called `AID-FOREST` for fully automatic processing of terrestrial LiDAR point clouds based on trunk detection. The software takes in a raw point cloud, performs all required preprocessing steps, and runs an object detection neural network on a series of horizontal slices to find cross-sections of trunks, and then processes the detection results from each level to track individual trees.

Allen et al. [2022] describe an approach for classifying terrestrial LiDAR point clouds of individual trees extracted by `treeseg` using multi-view representation based on Goyal et al. [2021] and a convolutional ResNet network [He et al., 2016]. They report high classification accuracies both overall and per-species on a dataset of almost 2500 individual trees.

Wilkes et al. [2023] introduce `TLS2trees`, another automated framework for segmentation of individual trees in terrestrial LiDAR point clouds, consisting of a set of Python command line tools. The framework consists of three steps: preprocessing, semantic segmentation, and instance segmentation into a set of individual trees. After that, a set of attributes are computed for each tree. The software is designed to be horizontally scalable (meaning it's easy to speed up calculations by using more machines doing them in parallel, as opposed to vertically, meaning by increasing the computational resources available to a single machine) to address huge datasets produced during terrestrial laser scanning surveys.

All these software packages are impressive, but non-applicable to the data described in this work.

Viana et al. [2022] use a small dataset to compare tree-level inventory metrics extracted from a terrestrial LiDAR survey and from a manual inventory. They report very good results, showing that terrestrial LiDAR can serve as a replacement for manual inventories in diverse secondary forests (secondary forests are forests that regenerate naturally after significant disturbances like logging, storms, or fires).

Nurunnabi et al. [2024] offer a compelling example of how detailed terrestrial LiDAR point clouds can be used to provide very detailed analyses on tree level. The authors report high accuracies on the task of segmenting the point clouds into wood and leaf points by utilizing geometric features mentioned earlier to locate linear and non-linear areas. They then apply an octree-based segmentation algorithm to develop a precise 3D structure of a tree.

2.4.3 UAV LiDAR

A common approach to utilize LiDAR point cloud data for tree detection is by calculating canopy height maps (CHMs) – images where each pixel represents the height

of vegetation – and applying the same techniques as for image-only approaches.

A substantial number of different tree detection algorithms are variations of the local maxima filter, differing in what the filter is applied to, how many times, with what windows, and how the results are combined or preprocessed. For example, Douss and Farah [2022] offers an exploration of how different window size functions for the local maxima filtering applied to LiDAR-derived canopy height maps affect the quality of individual tree detection. They show that for a moderately dense forest in France, it is possible to find the parameters for the local maxima filter window that produce satisfactory results. It is, however, clear that it requires great deal of manual adjustment, and visual inspection of the results of detection makes it clear that they are only locally consistent. Any change in the canopy structure patterns noticeably affects the quality of the detection even for sophisticated window functions.

Lisiewicz et al. [2022] propose a way to improve the results of canopy height map-based individual tree segmentation algorithms. Their approach consists of three steps. The first step is the classification of segments into correct, under-segmented, or over-segmented using a Random Forest classifier, the second step is the refinement of under-segmentation errors by re-segmenting them with adjusted parameters, and the last step is the refinement of over-segmentation errors by merging with the correct segments using an algorithmic approach based on a collection of intensity-derived features. The authors suggest that this approach is universal in terms of forest composition and complexity, unlike many other ad-hoc correction methods based on the study area and thus non-generalizable.

Wang et al. [2023] propose a two-stage network they call Tree Region-Based Convolutional Neural Network (RCNN) to detect trees in UAV LiDAR point clouds. The first step is generation of dense anchors across the point cloud that are then processed by the RCNN to generate proposals for individual tree locations. The second step involves multi-position feature extraction to refine the proposals. The approach is evaluated on the NewFor benchmark [Eysn et al., 2015] and outperforms all benchmark methods that come with it.

Fu et al. [2024] propose a method for segmenting individual trees in UAV LiDAR

point clouds based on a multiscale adaptive local maximum filter applied to the smoothed canopy height map to detect tree tops, a region-growing method for crown delineation that are then refined by a voxel-based clustering algorithm. The method is developed and tested on 21 synthetic plots and one actual survey, and the authors report good accuracy for estimation of both tree locations and heights. Worth noting that both the synthetic and real forest have relatively well-defined an unambiguous canopy structure.

2.4.4 Fusion of data

Many works do not just use one data source and instead combine multiple complimentary ones for better representation. This subsection gives an overview of data fusion approaches for forestry applications on individual tree scale.

Lian et al. [2022] describe an approach for calculating individual tree biomass by combining UAV multispectral, UAV LiDAR and terrestrial LiDAR data. Terrestrial and UAV LiDAR are merged to get a more representative point cloud, showing the trees equally well both from above and from the ground. The multispectral image is used to generate a species classification map, which is then used to segment the point cloud. The segmented cloud is then used to estimate diameter at breast height and height and calculate above ground biomass.

Li et al. [2023] explore the contributions of multispectral images, very high resolution panchromatic images, and LiDAR data during fusion on feature and decision level for tree species classification using Support Vector Machine and Random Forest classifiers on manually delineated tree crowns from an urban environment. They report that fusion consistently improves the results over using each of the data sources on its own. They also report that fusion on decision level resulted in the highest overall accuracy metrics.

Balestra et al. [2024] offers a review of 151 publications concerning fusion of LiDAR data with other remote sensing data. The authors report that in most cases fusion improves the results. Most relevant to this thesis, they report that for individual tree segmentation the results of fusion-based approaches compared to LiDAR-only ones are consistently better. La et al. [2015] show an increase from

63% to 92% for low-density forests, and from 62% to 70% for high-density forests, Aubry-Kientz et al. [2021] show and improvement by 5%, and Zhen et al. [2014] and Arenas-Corraliza et al. [2020] improved their results by 2-4%.

Ferreira et al. [2024] use a U-Net semantic segmentation model for fusion of VNIR images and airborne LiDAR-derived feature maps including surface normals of the canopies, reflection intensities, canopy heights, and leaf-area index for mapping tree species in urban tropical areas. Their results show that adding LiDAR-based features improves F_1-scores across all considered species, with average F_1-score 84.1. They also use the segment anything model [Kirillov et al., 2023] to automatically segment tree crowns with outstanding 98% boundary F_1-score.

Wu et al. [2024a] use a combination of high-resolution RGB images and LiDAR to classify urban tree species by extracting seven different feature types and using a Random Forest for pixel-level classification. They report on seven experiments using different combinations of extracted features. The results show an improvement of 18.5% in accuracy when using fusion of LiDAR and RGB compared to RGB-only.

2.4.5 Summary

There is a lot of research going on currently on individual tree-level inventories using UAV remote sensing data. Most results that can be considered successful are either in more mild environments, such as urban or managed forests, in predominantly coniferous forests, or use a much more detailed, but at the same time much more labor-intensive data source – terrestrial LiDAR. Mild forest types usually have canopy structures that are a lot easier to interpret, since the trees often stand far enough apart to be separable. At the same time, they allow for much more signal penetration, resulting in a good representation of tree trunks in the data, which is tremendously useful for the task of detecting trees. Predominantly coniferous forests, even dense ones, also have a canopy structure that is easy to interpret because conifer trees have a characteristic triangle shape that results in spikes in canopy height even when trees are standing close to each other. Terrestrial LiDAR data is different from UAV LiDAR in the amount of detail and, most importantly, the perspective. It surveys the trees from below, virtually guaranteeing the presence

of trunks in the point clouds. Several fully automated approaches to detailed forest inventories using terrestrial LiDAR data exist that rely on the detection of trunks. A gap persists in solutions that would consistently work on UAV remote sensing data over dense mixed forests, with complex overlapping canopies. The active nature and the vertical structure information provided by LiDAR is essential in such complex environments. At the same time, many results indicate that fusion of LiDAR point clouds with other data sources consistently improves the results in forestry applications, which is why the proposed framework relies on fusion of LiDAR with RGB imagery.

Chapter 3

Materials and methods

This chapter describes in detail used datasets, processing techniques and the methods, and comments on the reasoning behind the choices. The code of the project on my GitHub might be a helpful addition for that.

3.1 Lysva dataset

The main original dataset used in the study is the Lysva field inventory dataset, named by the closest town to its location. The dataset is released into open access with an accompanying paper that describes the data in detail and provides a basic baseline for individual tree detection [Dubrovin and Fortin, 2024]. The study area is located in Perm Krai, Russia, 86 kilometers to the east of Perm. The forest in the region is natural, with the average age of coniferous trees over 85 years and deciduous over 60 years. The dataset consists of a field inventory of 3600 trees across 10 rectangular ground plots 100 meters in lengths and 50 meters in width fully covered by a UAV LiDAR and RGB orthophoto surveys. Figure 3-1 shows the locations of the ground plots over the full size RGB orthophoto and a visualization of the field inventory for a single plot on top of the LiDAR point cloud. The low vegetation areas visible on the orthophoto are naturally regenerating old logging and agricultural sites. No artificial afforestation has been done in the area. Colored points represent trees, with different colors mapping to different species. The point cloud is visualized as a 2D scatter plot with points colored by height (darker points

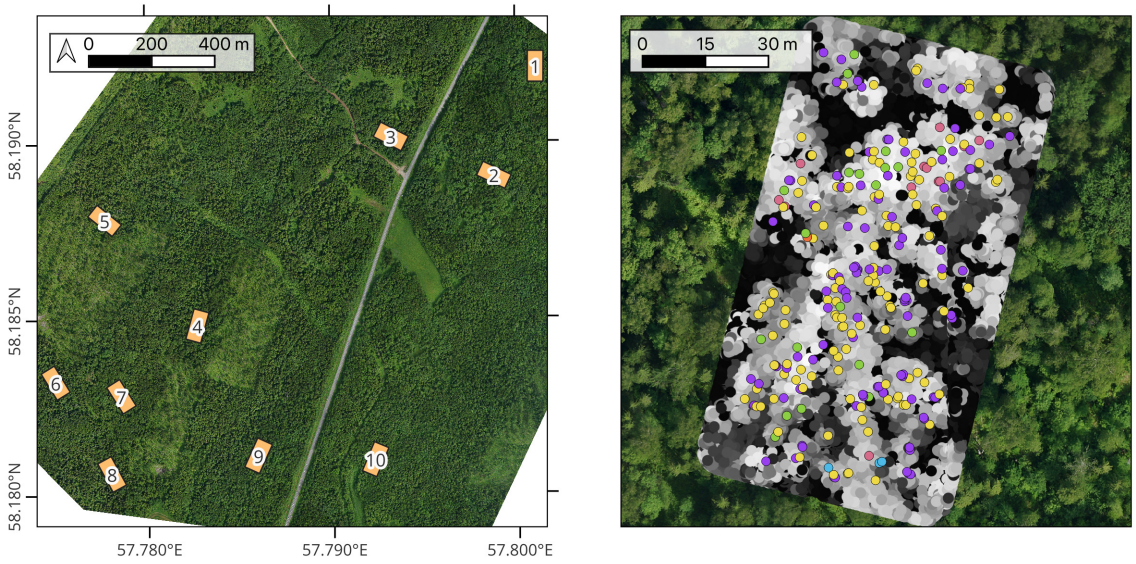


Figure 3-1: The study region for the Lysva field inventory dataset. **Left**: The locations of field survey plot boundaries on a full-size RGB orthophoto. Each plot is a 50 by 100 meter rectangle, with every tree with dbh of 6.1 cm or higher measured and recorded. Buffered cutouts of the orthophoto come with the dataset, along with LiDAR point clouds. **Right**: A close up of plot number 4. Each colored point represents a single tree of a different species, on top of a point cloud colored by the value of the height of each point (lower points are dark, higher points are bright) and the same orthophoto. Note that the points of the point cloud are unsorted, and some lower points overlap higher points. Figure reused from Dubrovin et al. [2024].

are lower, brighter points are higher, and points are unsorted – some lower points end up over higher ones). The baseline is a simple one-pass local maxima filter applied directly on the point cloud with a fixed window size.

The field inventory is a tabular dataset where every row represents a single tree. According to the state-mandated requirements that were in place during the collection of the data, all trees with dbh starting from 6.1 centimeters were included. Table 3.1 shows a random sample of entries from the field inventory table. Every tree is represented by a point in UTM 40N coordinate reference system (EPSG:32640). The coordinates of the trees were determined using the South NTS-360R total station with a reflector prism using the closed traverse method. The total station measurements used two points basis determined using a dual-frequency South G1 Plus IMU GNSS receiver in Real-Time Kinematic (RTK) mode. The basis points were selected to have reliable satellite signal. The average error of coordinate mea-

surements is 10 centimeters, with a maximum of 30 centimeters. Every tree has a species label and diameter at breast height (dbh), measured with calipers at 1.3 m from the ground at two perpendicular directions and averaged. Figure 3-2 shows the distribution of species in the data: the dominant species is spruce, but overall the trees are evenly split between deciduous and coniferous, 1793 and 1807 respectively, with seven species in total: spruce, birch, fir, aspen, tilia, alder, and willow. Approximately 20% of the trees have height data measured during the inventory, and 10% have ages measured on core samples, shown in the table in meters and years respectively.

Table 3.1: Example of data in the field inventory table. Each row is a recorded tree and is represented by a point in UTM 40N coordinate reference system (EPSG:32640). Coordinates are not shown.

Plot	Tree ID	Species	d_1	d_2	dbh	Age	Height	Angle	Comment
7.0	111.0	Birch	23.0	23.0	23.00	–	–	0.0	–
5.0	136.0	Fir	23.5	23.0	23.25	90.0	17.5	0.0	Rotten
3.0	119.0	Aspen	36.1	42.1	39.10	89.0	25.5	0.0	–
9.0	345.0	Spruce	19.7	22.0	20.85	–	15.9	0.0	–
6.0	267.0	Spruce	12.9	12.9	12.90	–	–	0.0	–

The LiDAR sensor used for the survey is AGM-MS3 produced by AGM Systems. It has 640 kHz acquisition rate, 300-meter range, and spatial accuracy of 3–5 centimeters. The raw point clouds were processed with the combination of the AGM ScanWorks software from the sensor vendor and the TerraScan software. The point clouds were preprocessed by removing duplicate points and high and low noise points. The duplicate removal was run with a threshold distance between points of 1 mm. Noise was removed by visually inspecting the point cloud and manually selecting height thresholds to cut off points that are lower than the ground or higher than the canopies. Ground point classification was performed and ground points were used to normalize height by subtracting the ground level from the Z coordinate of every point. Height normalization allows treating the Z coordinate as height above ground rather than the absolute elevation, which simplifies many subsequent steps.

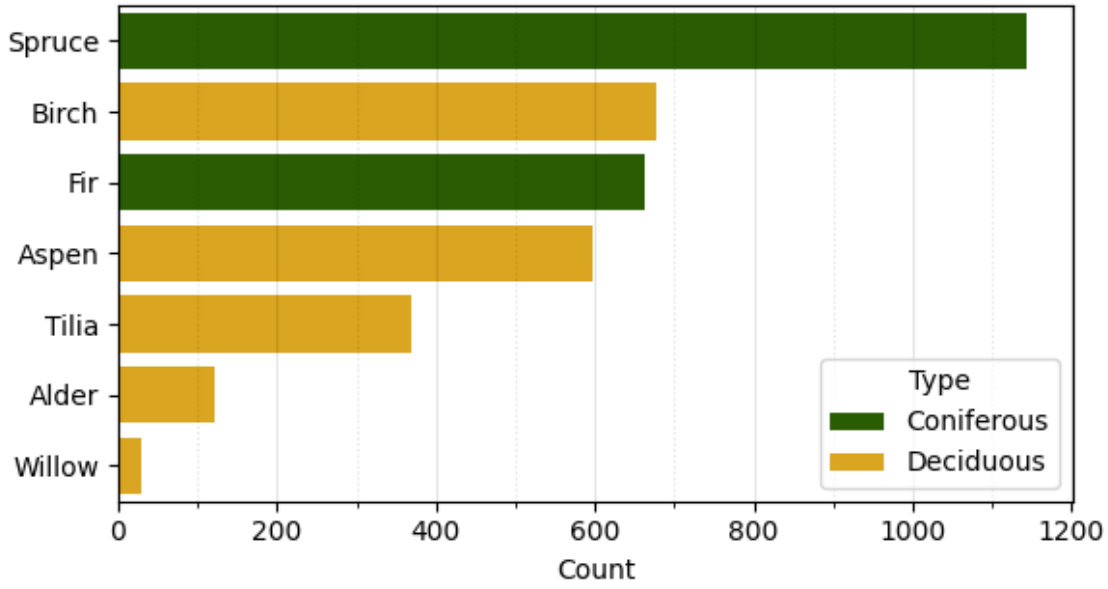
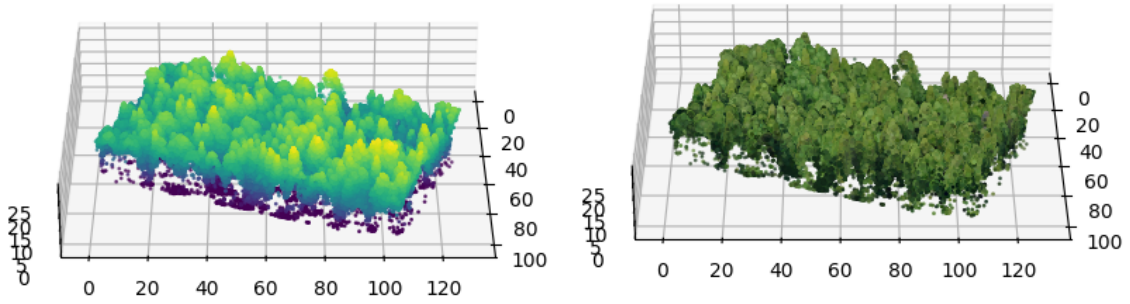


Figure 3-2: Distribution of species in the Lysva field inventory dataset. The dominant species is spruce, but overall the split between coniferous and decidious species is even: there are 1807 coniferous and 1793 deciduous trees. Figure reused from Dubrovin et al. [2024].

The camera used to collect raw digital images for creating the orthophoto is Sony A6000. Agisoft Metashape software was used to generate the orthophoto mosaic. A canopy-height map was created from the LiDAR point cloud and used as a reference for adjusting the planar coordinates of the orthophoto to align them. The resolution of the orthophoto is 7 centimeters per pixel. Figure 3-3 is a 3D visualization of the point cloud over plot number 10. It shows the unmodified point cloud on the left, with points colored by height above ground, and a point cloud enriched with color information by sampling the orthophoto at the planar coordinates of the points. Figure 3-4 shows the orthophoto for the same plot. The carrier UAV used for the LiDAR survey is DJI Matrice 600 Pro hexacopter. The speed during the survey was 10 meters per second. The UAV was configured to follow the terrain at 150 meter height using the SRTM elevation map as a reference [Farr and Kobrick, 2000]. The carrier UAV used for the orthophoto survey is the fixed-wing Geoscan 201 Geodesy. The speed during the survey was 16.6 meters per second. Figure 3-5 shows the UAVs used for the remote sensing data collections. Table 3.2 summarizes the sensor and platform characteristics, as well as some of the acquisition parameters.



(a) Points colored by height.

(b) Points assigned color by sampling the orthophoto.

Figure 3-3: A 3D visualization of the UAV LiDAR point cloud of plot 10.

Table 3.2: Summary of the equipment used for remote sensing data collection in for the Lysva dataset.

	LiDAR survey	**Orthophoto survey**
Sensor	AGM-MS3	Sony A6000
Resolution	37 points/m^2	7 cm/px
Carrier UAV	DJI Matrice 600 Pro	Geoscan 201 Geodesy
Carrier UAV type	Rotary-wing (hexacopter)	Fixed-wing
Acquisition height	150 m	150 m
Acquisition speed	10 m/s	16.6 m/s

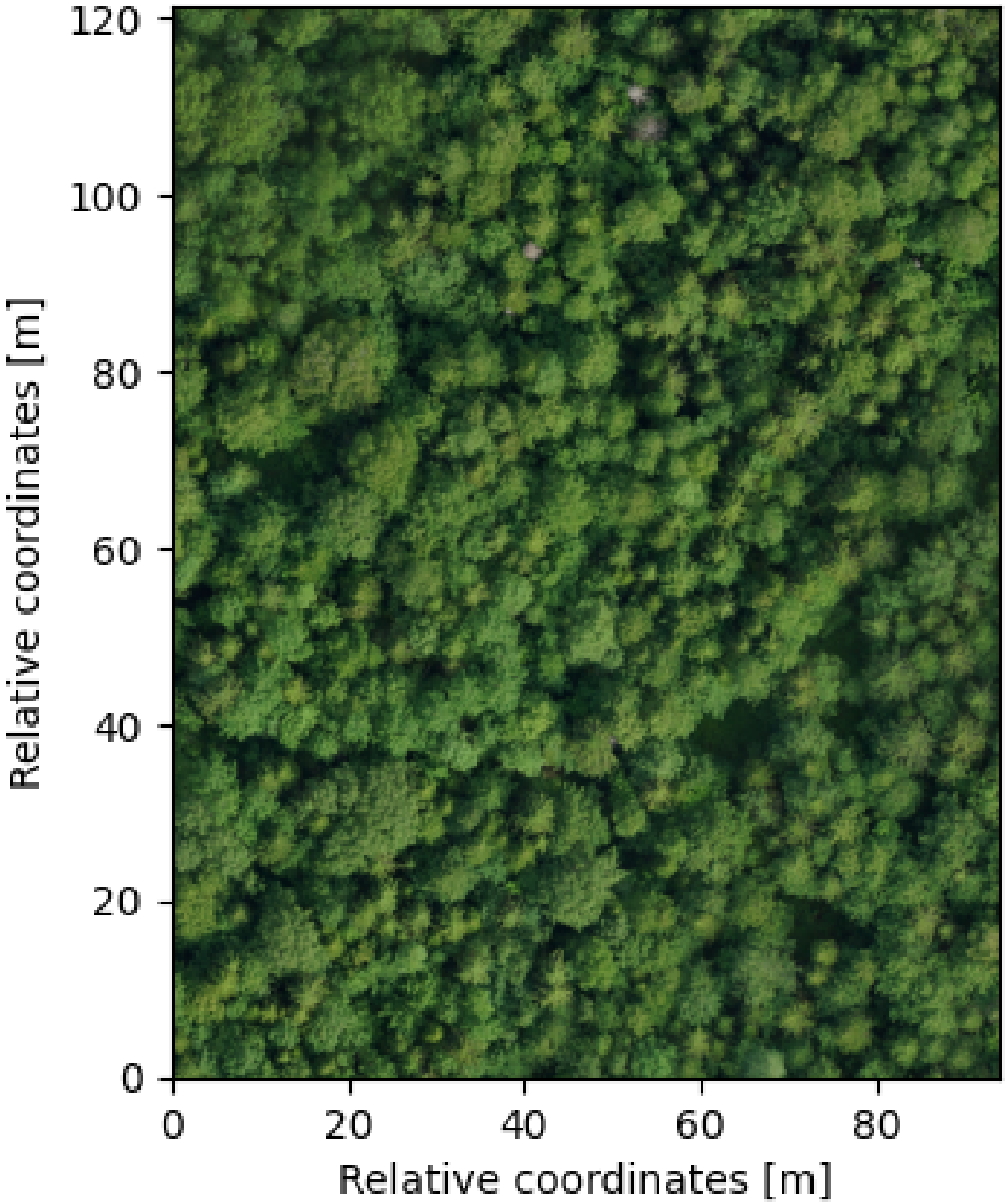


Figure 3-4: A visualization of the orthophoto of plot 10.

Figure 3-5: UAVs used for data collection (photos from vendor websites). **Left**: DJI Matrice 600 Pro that was used for the LiDAR survey. **Right**: Geoscan 201 Geodesy that was used for the orthophoto survey.

Table 3.3: Statistics for the plots in the Lysva dataset.

Plot	Tree count	Point density	Dominant type
1.0	420	31.7	Deciduous
2.0	365	47.9	Deciduous
3.0	332	40.3	Deciduous
4.0	261	33.5	Coniferous
5.0	208	14.2	Coniferous
6.0	290	39.1	Coniferous
7.0	408	41.9	Deciduous
8.0	341	35.5	Coniferous
9.0	459	42.1	Coniferous
10.0	518	42.9	Deciduous

Table 3.3 shows some descriptive statistics for each plot in the field inventory: the number of trees in the plot, average LiDAR point density in points per square meter, and dominant species type. The overall average point density is 37 points per square meter. Exactly half of the plots are predominantly coniferous and half are predominantly deciduous. Figure 3-6 shows two point clouds clipped by plot bounds in 3D, highlighting the differences in canopy structure complexity between predominantly deciduous and coniferous plots. The figure highlights that the forest is indeed dense and mixed, with non-uniform, complex canopy structure.

On using intensity-based features

Even though some sources report intensity-based features as some of the most important ones Shi et al. [2018], the features seem to me unreliable in the context of forestry because of the physics of light reflection. There are simply too many factors

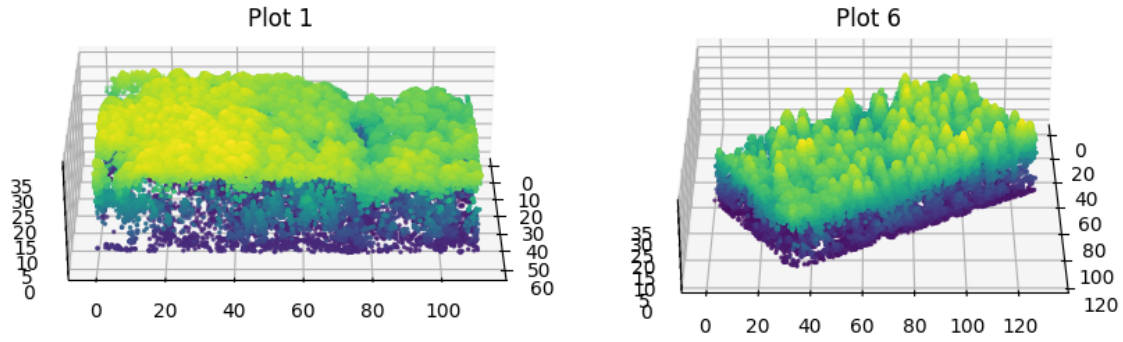
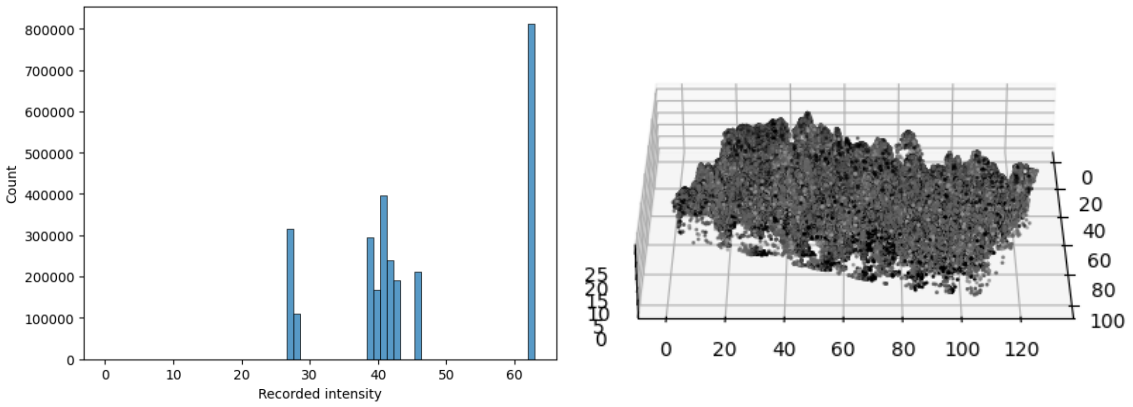


Figure 3-6: 3D visualizations of point clouds from plot 1 (predominantly deciduous) and plot 6 (predominantly coniferous). Note the difference in the canopy structure: it is relatively easy to tell conifers apart visually, while deciduous species do not have pronounced shapes and are hard to discriminate. Figure reused from (Dubrovin and Fortin 2024).

that affect the amplitude of the reflected signal, which might be useful when imaging stable targets such as urban environments but become completely unpredictable on highly unstable targets such as trees: they move in the wind in the time span of a single survey, they grow in the time span between repeated surveys, the leaves and branches are angled in every possible way. Moreover, the quality of the recorded intensities highly depends on the used sensor. As an example, Figure 3-7 shows the distribution of intensity values for every point in the Lysva dataset, and shows plot number 10 in 3D with points colored by their recorded intensity. The distribution of intensities seems like an artifact of faulty quantization (the fact that the maximum overall value is 63 makes me suspect it is stored by the hardware using a 6-bit unsigned integer, probably an ad hoc optimization by the sensor vendor). With this distribution in mind, it is not surprising that coloring points by their intensity values results in images that look like noise, and there is no signal to be exploited for predictive modeling.

Some sensors do provide more consistent values. However, relying on intensity-based features limits the applicability of developed models and methods, as any such model would surely fail on data like this.



(a) Distribution of intensity over all points in the Lysva dataset.

(b) A point cloud of plot 10 with points colored by intensity.

Figure 3-7: An example of the unreliability of the intensity attribute.

Comparison to other datasets

Our dataset is in many ways similar to the NewFor benchmark Eysn et al. [2015]. It serves the same purpose and also offers overlapping field survey ground plots and UAV LiDAR point clouds. There are, however, many notable differences. The NewFor benchmark covers much more diverse regions, including ground plots from France, Italy, Switzerland, Austria, and Slovenia, while all our data comes from the same area. The tree species covered by the datasets are also different: both contain spruce and fir, but the NewFor data also has beech, Scots pine, larch, sycamore, and poplar, while ours also has birch, aspen, tilia, alder, and willow. Our dataset has more than twice as many individual trees as the Alpine benchmark, and the forest is denser and more complex, making it more complicated to detect trees in. Our data contains very mild terrain variations, while the slopes of the terrain in Alpine data are very steep, which plays a role during height normalization, since subtraction of steep terrain introduces artificial slope to the points in the canopy, changing the overall shape of the tree. Our dataset has an additional information source – an RGB orthophoto that allows development of algorithms that fuse multimodal data, which, we believe, is a key to success in such complex environments. Our dataset

has species labels for every surveyed tree, but only partial coverage of tree heights and no timber volume information at all.

Another similar dataset is the NeonTreeEvaluation Benchmark Weinstein et al., which offers bounding box annotation for tree detection across a wide range of different forest types. It offers coregistered RGB, LiDAR, and hyperspectral images over 31,000 individual trees. The main difference in the reference data between the NeonTreeEvaluation Benchmark and our dataset is the source: our data comes from a field inventory and thus has additional tree information that can be used in downstream tasks, such as species classification or timber volume prediction, while the NeonTreeEvaluation Benchmark annotations are created from the RGB photo and thus only offer the positions and sized of trees.

Another dataset is the IDTReeS 2020 Competition Data Graves and Marconi [2020] aimed to develop algorithms for delineation and species classification of individual tree crowns in RGB, LiDAR, and hyperspectral data. It offers bounding box annotations for 1200 individual trees covered by RGB, LiDAR, and hyperspectral images in 3 national forests in the USA. Similarly, the source of the data is annotation of images, not a field inventory.

There are also datasets available that do not have LiDAR point cloud coverage, or use terrestrial LiDAR instead of UAV LiDAR, or use photogrammetric point clouds instead of LiDAR point clouds. We do not mention them here because we specifically focus on UAV LiDAR.

3.2 Individual tree point clouds dataset

The main dataset used for training the models is a collection of point clouds of individual trees, sometimes referred to in this text as tree clouds, extracted manually from larger UAV LiDAR point clouds. The dataset is released into open access, and was originally presented in [Dubrovin and Fortin, 2024], although it has been expanded since and now contains twice as many individual trees. It consists of 394 trees, 192 of which are extracted from the previously described Lysva survey, and 202 from other surveys in Perm Krai. The distinction between the parts is important

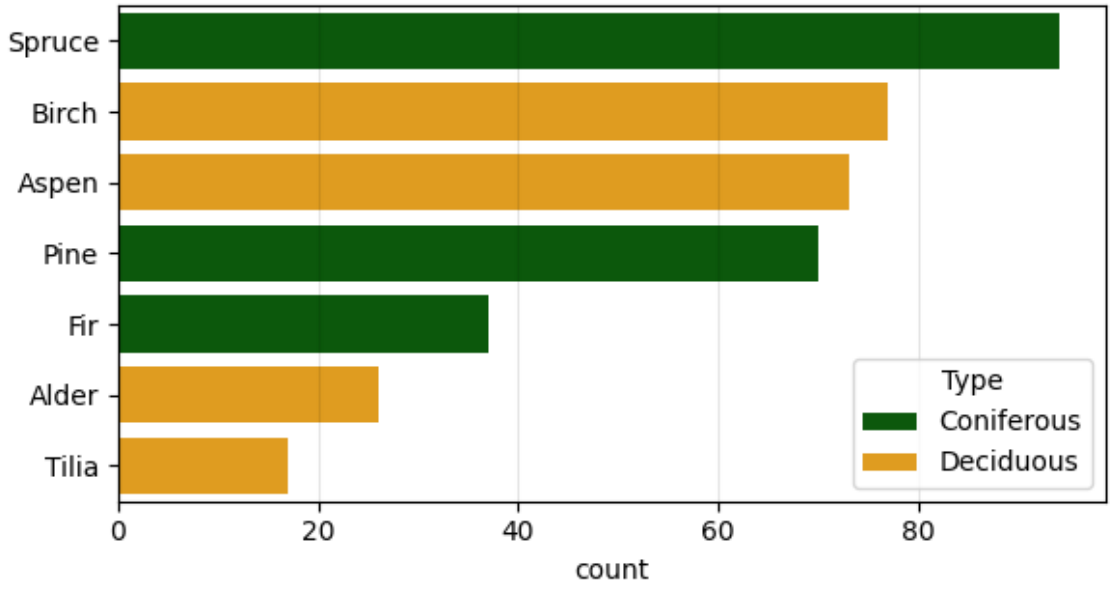
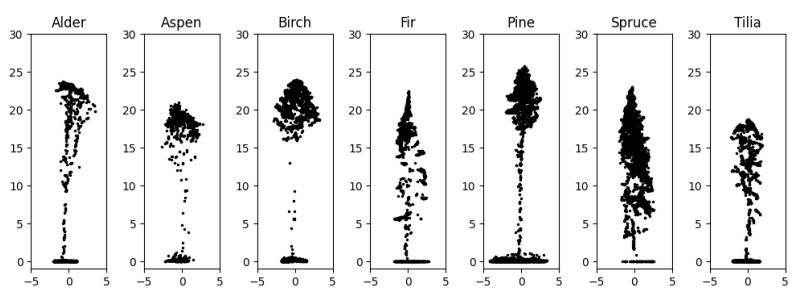


Figure 3-8: Distribution of tree species in the individual trees dataset. It contains 394 point clouds of individual trees: 201 coniferous and 193 deciduous. Note the presence of pine trees, as there are no pine trees in the Lysva field inventory – all of the pines come from other field surveys.

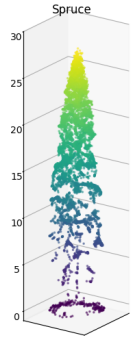
because the Lysva dataset has RGB orthophoto coverage, making it possible to infuse the tree clouds with orthophoto-based features. Thus, for training the tree segmentation networks that rely on these features, the effective size of the dataset is 192 tree clouds, as only the former part is used. However, the whole dataset is used for training regression and classification models that process segmented trees.

Figure 3-8 shows the distribution of species in the individual tree point clouds dataset. The split between coniferous and deciduous trees is almost even: there are 202 coniferous and 193 deciduous trees. There are seven species in total: spruce, birch, aspen, pine, fir, alder, and tilia, with most focus on 4 most important species listed first. Note the presence of pine trees, which are not present in the Lysva field inventory data, and the absence of willow trees. All the pine trees are from other field surveys. Willows are skipped intentionally, since they are of little interest in terms of timber harvesting: they are considered low quality, and their ripening cycle is mismatched with main timber species – when the overall plot is ready to be harvested, the willows are already rotten.

Figure 3-9 is a visualization of the data. It shows a random tree of every species



(a) Cross-sections of random trees of every species (ignoring the Y dimension).



(b) A spruce in 3D.

Figure 3-9: Visualizations of the individual tree point clouds in the dataset. Most of the tree clouds are top-heavy because of the observation from above, and some are artificially tilted because of slight terrain slopes and height normalization. The ground points are present.

as a 2D scatter plot and a single spruce as a 3D scatter plot with points colored by height. Because the observations are made from above, many trees have the highest concentrations of points at the top of their canopy and a very limited number of points along the trunk. Additionally, slight slopes of the terrain manifest as artificial tilt in some of the trees because of the height normalization of the original point cloud.

An important note is that the extracted trees were not chosen for extraction randomly but were selected by humans based on whether it was possible and relatively easy to separate from the surrounding trees. So there is a selection bias in there favoring the trees that are easily separable, standing outside of large dense clusters. Because of that the trees do not exactly represent what a tree closely surrounded by other trees is like in a point cloud. It is especially apparent in very pronounced trunks of all the visualized trees. In a dense forest, such as the one visualized in Figure 3-6, there is hardly ever enough penetration for such detailed trunk coverage. In fact, as mentioned in the literature review, good coverage of trunks is an immensely useful feature of terrestrial LiDAR surveys, which consistently show very good results using algorithm that segment trees from the trunk up.

3.3 Synthetic forests generated from individual trees

To train tree segmentation models, training data where every point is assigned to an individual tree is required. Data like that are very labor-intensive to label. An alternative way to create such data is by generating it from a set of tree clouds, as the per point labels arise naturally in this case. I used the dataset of individual tree point clouds described in the previous section to generate a synthetic forest to train a tree segmentation PointNet++.

Synthetic forest is generated in patches of set height and width by sampling individual trees from the full set and placing them from left to right, until the set width is extended, then from bottom to top, until the set height is extended. A height threshold is applied to each tree before placing it to remove ground points and non-tree reflections. This results in patches that are slightly bigger than the set size, but that can be cropped into the exact size. This has an added benefit of mimicking cutoff trees at the edges of the patch that are inevitable when the model is applied in a sliding window. The trees can be sampled with or without replacement, but since a set of augmentations described in Section 3.4.2 is applied to each tree individually, exact same tree never occurs in the patch fed to the model even when sampling with replacement. When trees are placed, their planar bounding boxes are tracked to avoid overlap. However, since some overlap might, in fact, be desired to better represent actual forests, a parameter that controls the amount of overlap is added to the dataset generation process. Each tree is also assigned a label, which simply tracks its ordinal number in the patch. An example of a 20 by 20 meter synthetic forest patch with 0.75 meter overlap and 2 meter height threshold is shown in Figure 3-10 from two different perspectives and using two color schemes: the label, i.e. the ordinal ID of the tree within a patch, and RGB color sampled from the orthophoto.

For comparison, a patch of the same size from a real point cloud over one of the plots of the Lysva survey is shown in Figure 3-11.

An alternative approach to creating patches of synthetic forest can be to use a fixed number of trees instead of fixed patch size. However, it poorly correlates

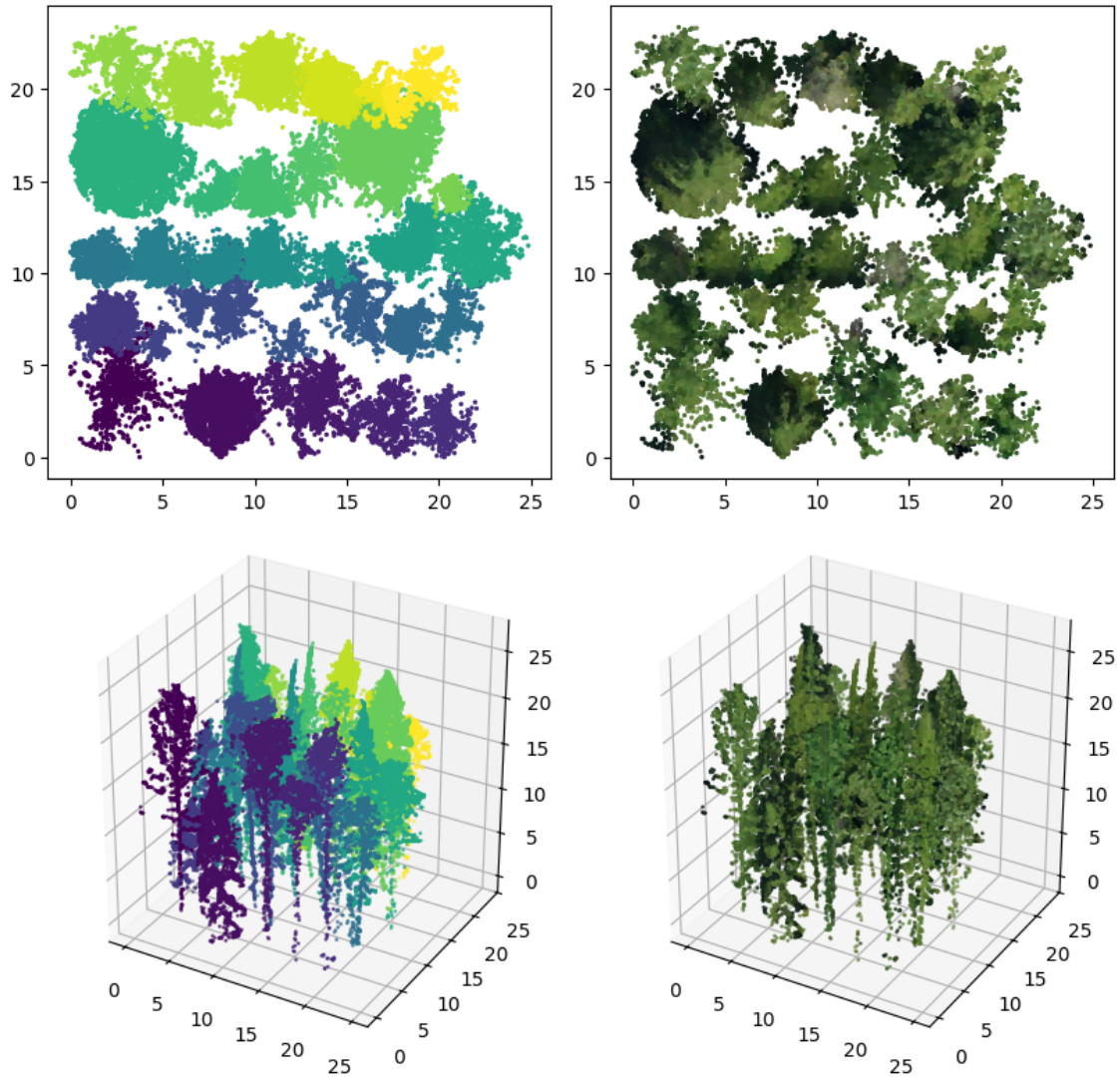


Figure 3-10: A visualization of a synthetic forest patch used for training the tree segmentation network. Patch width and height are set to 20 meters, overlap is set to 0.75. **Top:** Top-down view of the patch. **Bottom:** 3D view of the same patch. **Left:** Points colored by label: unique tree ID within the patch. **Right:** Points colored by RGB color sampled from the orthophoto.

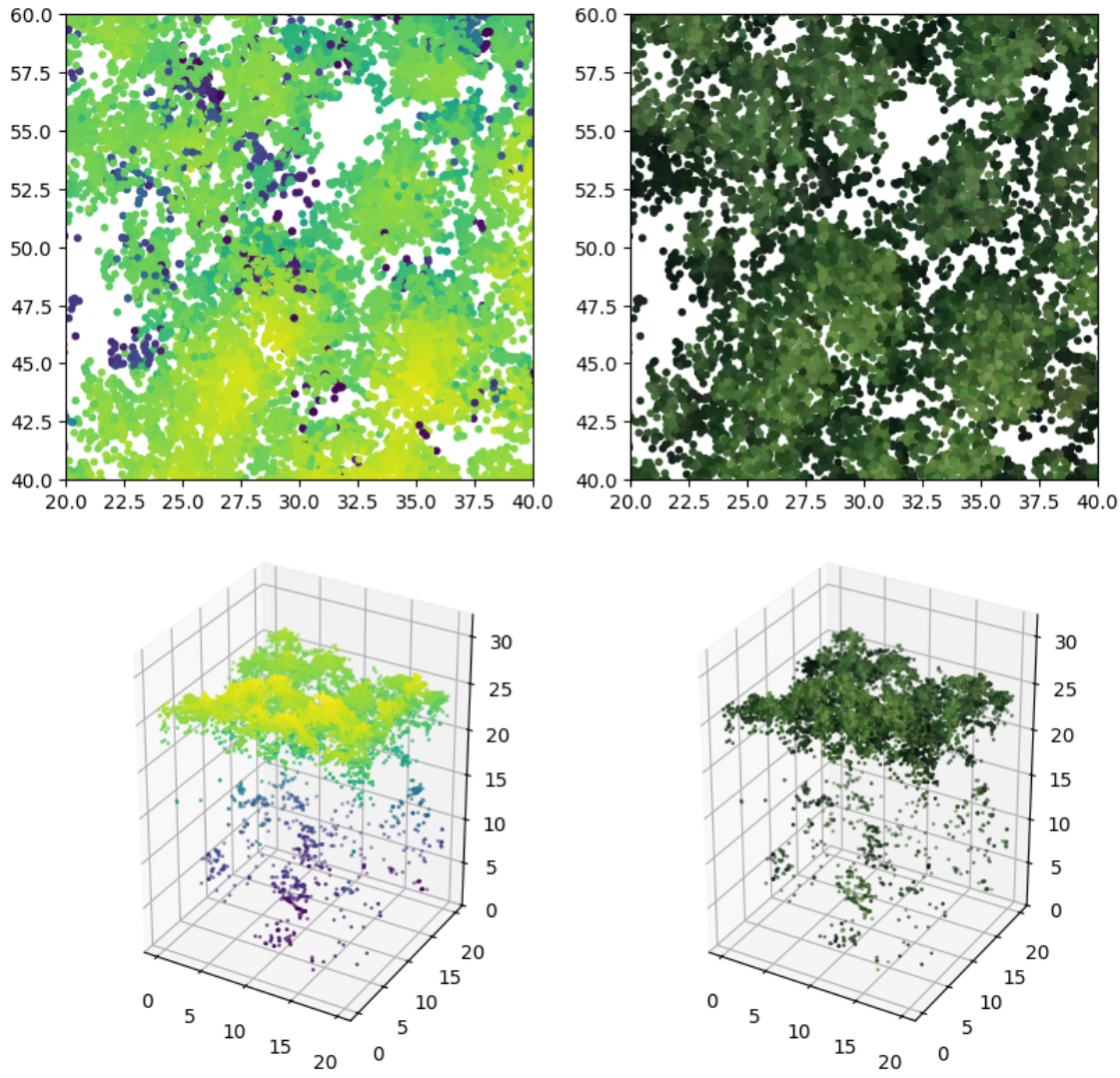


Figure 3-11: A piece of the Lysva LiDAR point cloud over plot 10 in within a 20 by 20 meter window. **Top:** Top-down view. **Bottom:** The same patch in 3D view. **Left:** Points colored by height. **Right:** Points colored by RGB color sampled from the orthophoto.

with how the final model will be applied: in windows of fixed size, not windows with varying size but fixed number of trees. And size of the patch the model sees is important because the scale is normalized, and using a different patch sizes changes the scaled coordinates and confuses the model that learned to rely on them.

Enriching point clouds using only color information from RGB orthophotos provides limited utility because it adds only local information to the points. As was mentioned in the introduction, one of the aspects of the complementary nature of the two data sources is that images provide continuous representations and capture textures. To enrich the image features with context, they can be preprocessed with feature extractors that encode context into pixel values and thus provide more information for the segmentation network. The feature extractors can be simple algorithms or specialized convolutional network feature extractors. For the same reason a simple PointNet++ is used as the architecture for the tree segmentation network, a simple collection of multi-scale features are used as orthophoto features, including intensity, edge, and texture features, extracted from the orthophotos using the `scikit-image` Python package. The features are calculated on different scales by applying Gaussian smoothing with varying parameters before calculation. Figure 3-12 shows example features from every mentioned group on a very fine scale. The top left image is the original orthophoto of plot 10 from the Lysva survey, the top right image is the intensity calculated from it, and bottom images are examples of an edge feature and a texture feature. The same features but on a coarser scale are shown in Figure 3-13.

It is worth mentioning that using synthetic data for training comes with potential biases that are important to keep in mind. The biases present in the source dataset can easily become exaggerated. In this case, the selection bias mentioned in Section 3.2 is the main potential culprit. The limited size of the source dataset also limits the variability of the examples, which can easily lead to complex overfitting. In this case, the range of shapes of the trees within each species is small, making it hard to generalize well. Moreover, different species will most likely have different accuracies associated with them. Models trained entirely on synthetic datasets are also a lot harder to properly evaluate. These drawbacks are partially addressed by

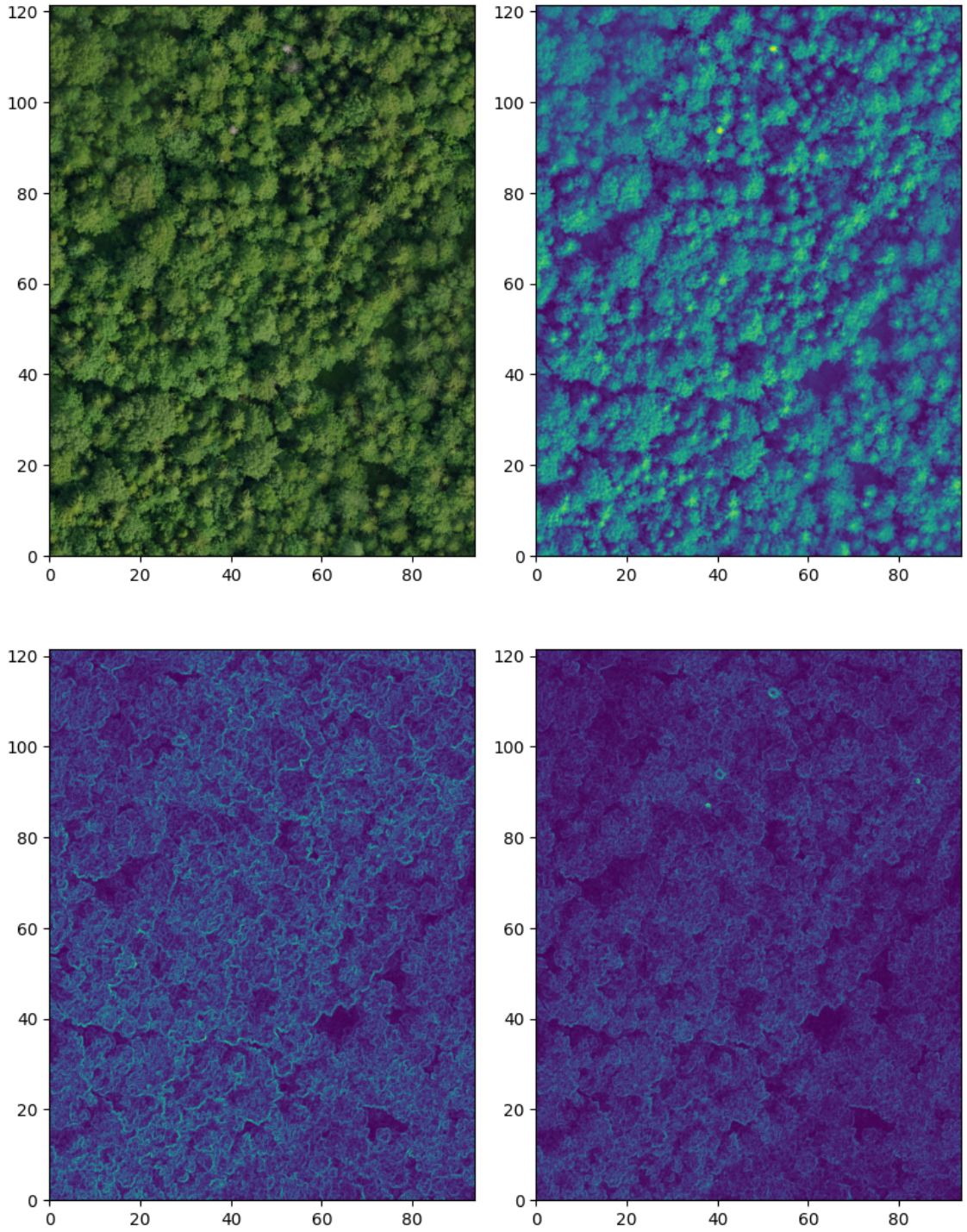


Figure 3-12: Basic orthophoto-based features used (on a single scale with sigma=1, actual features are across multiple increasing scales). **Top left**: The original orthophoto for plot 10. **Top right**: Intensity feature. **Bottom left**: Edges feature. **Bottom right**: Texture feature.

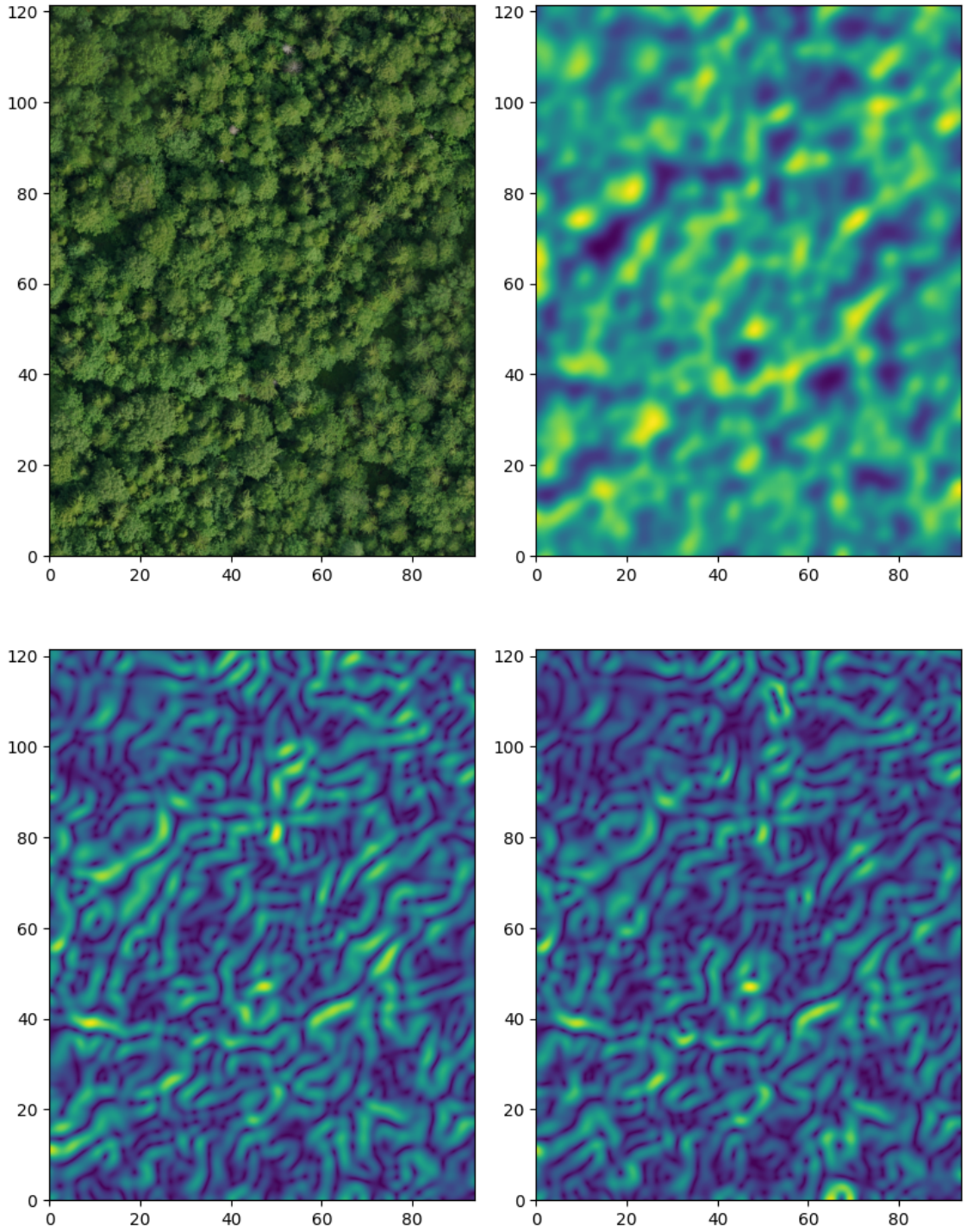


Figure 3-13: Basic orthophoto-based features used (on a single scale with sigma=16, actual features are across multiple increasing scales). **Top left**: The original orthophoto for plot 10. **Top right**: Intensity feature. **Bottom left**: Edges feature. **Bottom right**: Texture feature.

carefully choosing the augmentations for the training procedure, but they cannot be completely circumvented.

3.4 Training tree segmentation neural networks

The architecture chosen to serve as the tree segmentation network is the PointNet++ [Qi et al., 2017b], described in detail in Section 2.2. It is a relatively simple architecture, and further potential quality improvements might be achieved by using more modern and advanced architectures instead. However, the main goal of this thesis was to develop and verify an overall framework, and thus the choice was set on a model that is simple to implement and work with to allow easy experimentation with other parts of the proposed system.

The architecture of the used PointNet++ is similar to the segmentation architecture shown in Figure 2-3. The main differences from the shown architecture are the use of one more stacked set abstraction layer to make the network deeper, and the use of a regression head that predicts a continuous value for each point instead of a classification head that predicts per-point class scores, since the model needs to assign a unique ID to every tree in the patch. The three stacked set abstraction layers have the proportions of points sampled set to 0.75, 0.5, and 0.5, and neighborhood radii for feature aggregation set to 0.1, 0.2, and 0.4. Note that the scale of the input is normalized (see next subsection for details), so the radii are not in meters. The model has 30 million trainable parameters.

3.4.1 Coordinate and feature normalization

It's a well established practice to scale the inputs to neural networks that is known to improve the speed and accuracy of gradient descent convergence [Bishop, 2006]. Before going through the network, a set of augmentations and transformations is applied to each synthetic forest patch. The augmentations are described in detail with visualized examples in Section 3.4.2. The transformations include scale and feature normalization – the coordinates are centered and normalized to the interval $(-1, 1)$ and the features are normalized to the interval $(-1, 1)$ without centering.

3.4.2 Data augmentation

The primary objective of data augmentation is to enhance the quantity, quality, and variety of data used for training [Mumuni and Mumuni, 2022]. This is especially important when the sizes of the available datasets are limited, like in the case of the individual trees data that is the base of the synthetic forest datasets. Carefully selected augmentations allow increasing the effective dataset size. However, it is important to pay attention that the chosen augmentations keep the transformed examples semantically equivalent to the original. A readily understandable example of a bad augmentation in the task of digit classification, which is often used as the "hello world" of deep learning with the MNIST dataset of handwritten digits [Deng, 2012], is a vertical flip, as most digits lose meaning when upside down. A flip is also a bad augmentation example for a synthetic forest patch described in Section 3.3, as it changes the order of the trees within a patch, thus breaking the labels that are assumed to increase in a specific pattern.

In the scenario when the data for is created by combining several smaller inputs, there is a possibility of applying augmentations on two different scales. The most simple approach is to treat a synthetic forest patch as a whole and apply augmentations directly to it. However, it is also possible to apply per-tree augmentations, effectively increasing the size of the underlying tree set from which synthetic forest patches constructed. Only the latter kind are used for training the tree segmentation network.

The first transformation that changes the shape but does not affect any semantics for an individual tree is a random rotation around the vertical axis. A tree remains completely the same when rotated, but the coordinates of all points change. Figure 3-14 shows the effect of applying random rotation transformation of different magnitudes to a single aspen tree. For visualization purposes, the rotation is forced to apply with full magnitude for every parameter. During training, the angle is uniformly sampled from the specified range. For the final tree segmentation model, the range is set to $[-180, 180]$ degrees, as no amount of rotation breaks the semantics.

Another transformation that keeps the tree the same but changes the coordinates of the points is random scaling. It simply multiplies the coordinates by a scaling

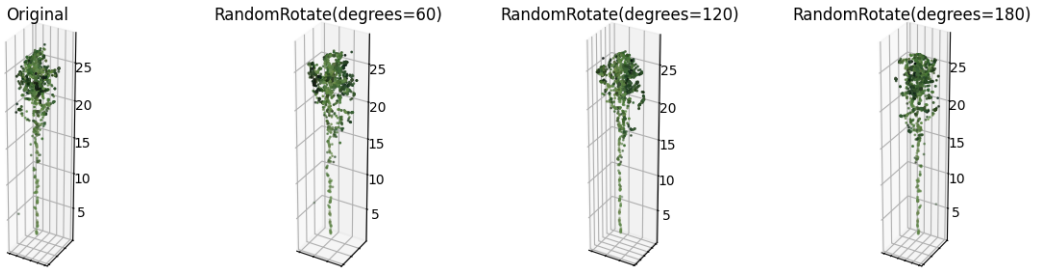


Figure 3-14: Visualization of the random rotation around Z axis augmentation on a single aspen tree. The effect is forced to happen with full amplitude for visualization purposes, during training a rotation angle is uniformly sampled from a set range.

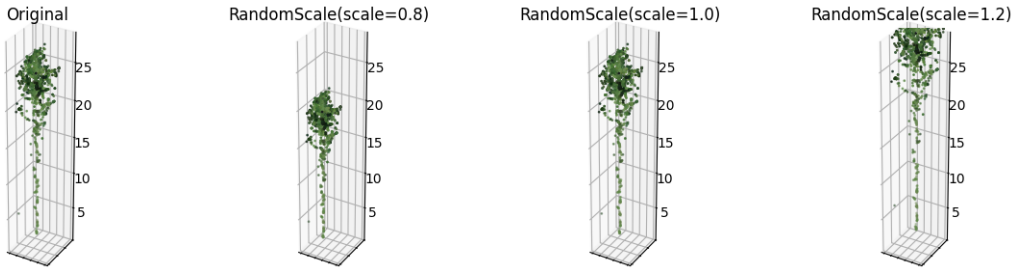


Figure 3-15: Visualization of the random scale augmentation on a single aspen tree. The effect is forced to happen with full amplitude for visualization purposes, during training a scale factor is uniformly sampled from a set range.

factor, making the tree larger or smaller in all directions. Unlike random rotation around the vertical axis, however, the range needs to be chosen much more carefully, as there is a possibility of making unrealistically large or small trees that would confuse the model during training. Figure 3-15 shows the effect of applying random scale transformation to a single aspen tree. Again, for purposes of visualization, the scale is forced to apply with full magnitude. During training, the scale is uniformly sampled from the specified range. For the final tree segmentation model, the range is set to $[0.8, 1.2]$.

Another way to change the positions is to slightly translate each point for a random distance in a random direction. That transformation is commonly referred to as random jitter. Since LiDAR sensors have limited spatial accuracy, introducing random translations within the accuracy range should not have any effect on the result of tree segmentation. Going further, small translations even outside the accu-

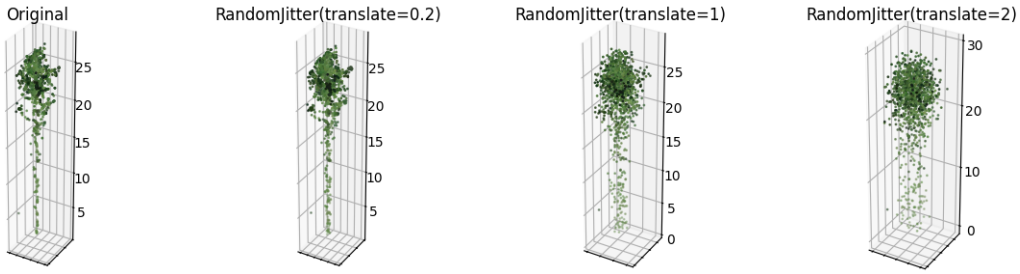


Figure 3-16: Visualization of the random jitter augmentation on a single aspen tree. The translation magnitude is uniformly sampled from a set range for every point.

racy range make sense, since they have very limited effect on the overall shape of the tree. Figure 3-16 shows this effect on a single aspen tree. The amount of translation is uniformly sampled independently for each point from a set range. Note that the random jitter augmentation is applied before the scale normalization. It matters because the augmentation's only parameter is the translation range, which depends on the scale. The parameters on the figure are thus in the original coordinate units – meters. Note how the shape of the tree almost does not change when the maximum range is set to 20 centimeters, and starts to become fuzzy and loose shape at 1 meter and higher. For the final tree segmentation model, the maximum translation is set to 30 centimeters.

Another useful effect augmentations can provide is to make synthetic data look more like real data. As was mentioned in Section 3.2, there is a selection bias in the dataset of individual trees: the trees that are easiest to manually separate are exponentially more likely to end up in the data. As such, the tree clouds in the individual tree dataset are significantly different from trees of the same species with similar sizes and shapes, but standing in dense clusters. One of the most important differences is that almost all the tree clouds have trunks, while in actual forest, dense canopy cover blocks most pulses and the trunk representation is very poor. One way to mitigate that issue is to apply a height-dependent dropout function to each tree cloud. For that purpose a probability threshold function that is around zero for the highest points of the tree and quickly ramps up to almost one for the lowest points is needed. An example function that satisfies these criteria is a modified sigmoid:

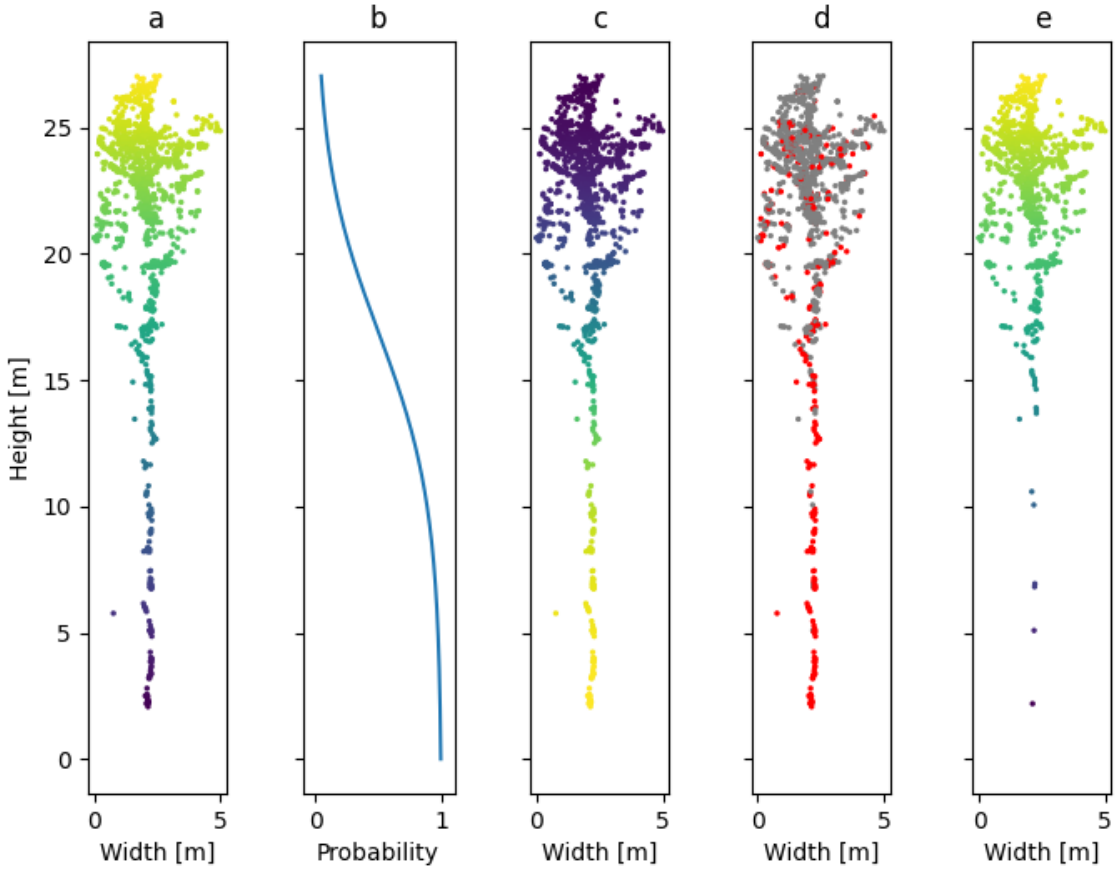


Figure 3-17: Effect of the height-dependent modified sigmoid dropout on a single aspen tree. Scale is set to 8, shift is set to 3. **a)** A single aspen tree with points colored by height. **b)** Height-dependent probability of dropout (a modified sigmoid). **c)** The same aspen with points colored by probability of dropout. **d)** The same aspen with points that will be dropped marked red. **e)** The same aspen after the dropout is appleid with point colored by height.

$$\text{threshold}(z) = \left[1 + e^{z \times \text{scale} + \text{shift}}\right]^{-1},$$

where z is the height normalized to $[0, 1]$ and reversed by subtraction from 1, scale and shift are hyperparameters that control the shape of the curve. Changing scale controls how steep is the climb from 0 to 1: the larger, the steeper. Changing shift controls the position of the climb: the larger, the lower. Figure 3-17 shows an example of applying such dropout function to a single aspen tree, and Figure 3-18 show the same tree with more aggressive parameters, resulting in much more points being dropped from the tree cloud.

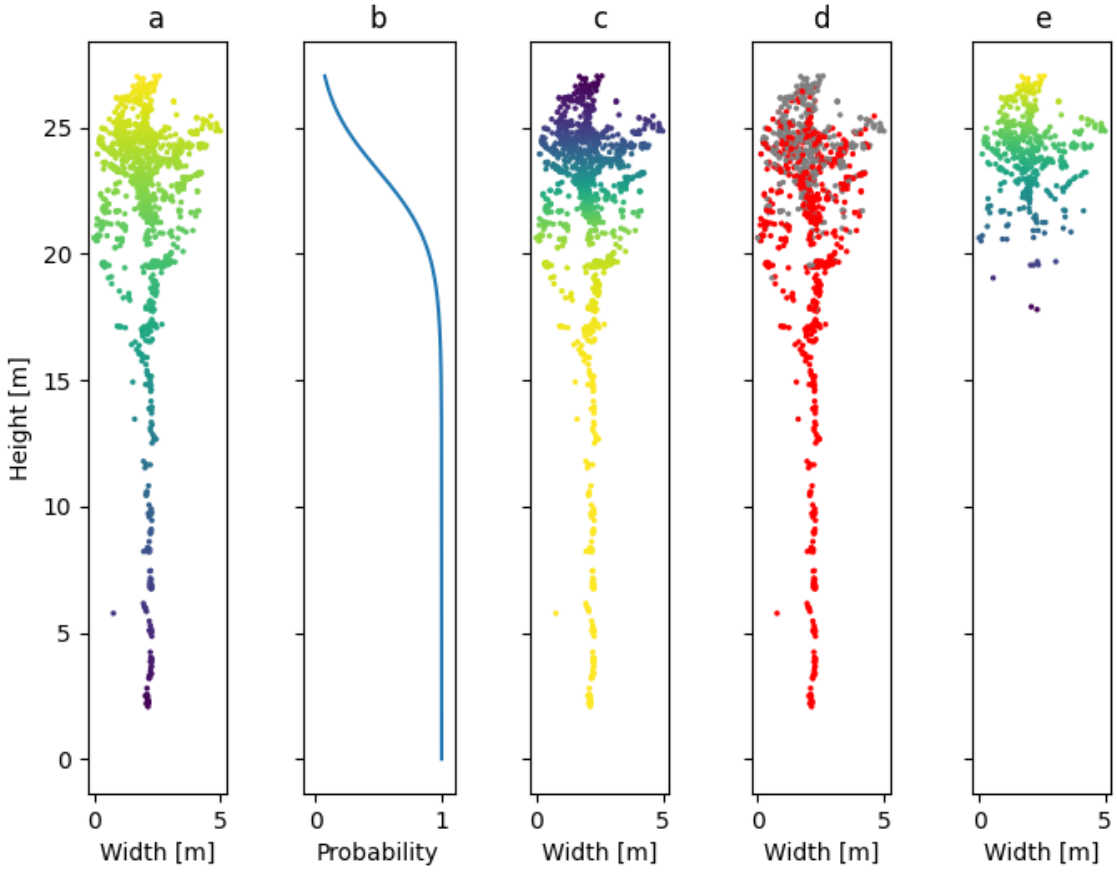


Figure 3-18: Same as Figure 3-17, but with a more aggressive dropout threshold function. Scale is set to 18, shift is set to 2.5.

All the augmentation are applied on the fly right before the examples are loaded into GPU memory and passed through the network.

3.4.3 Other training parameters

Mean absolute error is used as a loss function. Some experiments have shown improvements when using a custom loss function that modifies the mean absolute error loss by weighting it reversely proportional to the distance of the tree centroid. The idea of the modification is to make the network focus more on points closer to the center of each tree, as these points are much less likely to be overlapping with the crowns of adjacent trees.

The batch size is limited by the available GPU device memory, and in the described setup competes for memory space with the synthetic forest patch dimensions used in the training dataset. The preference is given to the size of the patch, so the batch size is set to 1, but is compensated for by using gradient accumulation steps – updates to the model parameters are made after the gradient is accumulated for a set number of iterations. This slows down the training, but enables usage of larger batches when there is not enough memory to fit them, making the training procedure overall more stable.

Early stopping [Prechelt, 1998] is set up to terminate training early if there is no improvement in average validation loss in a set number of epochs. This makes sure that valuable GPU time is not wasted on continuing training models that are likely to have started overfitting.

Model checkpointing is set up as well. After each training epoch, when the average validation loss and accuracy are calculated, the model state is saved to disk if it's current accuracy is better than the last saved one, where accuracy is the proportion of points for which the rounded integer label is correct. This makes sure that the best performing model can always be recovered, even if the training process was run for too long and the latest model is not the best one.

The learning rate schedule is set to follow a fast linear warm up and slow linear decay. Learning rate is set to ramp up from 0.001 to 0.01 in a span of 2 epochs, and then decay back to 0.001 in a span of 30 epochs. Adam optimizer is used [Kingma

and Ba, 2014].

Training was performed on an NVIDIA A100 GPU with 80 Gb of memory. Inference, depending on the point density of the input, might work even on a much smaller GPU like Tesla T4 with 16 Gb of memory. GPUs of this size are available with limits on usage for free on services like Google Colab and Kaggle.

3.4.4 On implementation of PointNet++

As mentioned in the introduction, the deep learning code, including the code for PointNet++, is implemented using the PyTorch Geometric library designed for writing and training graph neural networks. This makes the implementation not exactly the same as described in the PointNet papers. The main ideas, namely local feature learning in a k-nearest neighbors neighborhood and max pooling for permutation-invariant aggregation, are there. Input transform and feature transform networks are not defined explicitly, but networks that process features learn to perform a similar function.

3.5 Training segmented trees processing models

To predict per-tree forest attributes from a segmented point cloud, a set of specialized regression and classification models is trained on the tree clouds. These specialized models are classic machine learning models that operate on most common metrics described in the literature overview chapter. As mentioned in the literature overview, common features used for machine learning on point clouds are based on the eigenvalues of the covariance matrix of the point coordinates, calculated either for an entire cloud, or per-point in a neighborhood around it. These features include linearity, planarity, scatter, that aim to indicate the presence of linear, planar, or volumetric structures, and also omnivariance, anisotropy, eigentropy, the sum of eigenvalues, and curvature. The features are defined as follows, with eigenvalues sorted in descending order such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$:

$$\text{linearity} = \frac{\lambda_1 - \lambda_2}{\lambda_1}$$
$$\text{planarity} = \frac{\lambda_2 - \lambda_3}{\lambda_1}$$
$$\text{scatter} = \frac{\lambda_3}{\lambda_1}$$
$$\text{omnivariance} = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$$
$$\text{anisotropy} = \frac{\lambda_1 - \lambda_3}{\lambda_1}$$
$$\text{eigentropy} = -\sum_{i=1}^{3} \lambda_i \ln(\lambda_i)$$
$$\text{sum of eigenvalues} = \lambda_1 + \lambda_2 + \lambda_3$$
$$\text{curvature} = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

Another common set of features, especially popular in forestry applications, are various statistics that describe the height distribution of points within the neighborhood or the cloud. They include maximum and average height, standard deviation, kurtosis, skew, and entropy of the height distribution, percentage of points above the mean and each of the deciles of height and the deciles of height themselves. The features are calculated for the entire tree cloud, effectively reducing each tree to a collection of metrics, resulting in a tabular dataset. In this form, the dataset is used to train the models.

In Dubrovin and Fortin [2024], we propose a way to help build intuition into the meaning of some of the less obvious features by visualizing individual trees on a different end of the range of the feature's values. Figure 3-19 shows is an example of such visualization, showing the effect of the shape of spruce on omnivariance and the effect of the shape of aspen on percent of points above mean height. Note the presence of ground points, which are filtered out before calculating features for the models.

The feature set is thinned by using sequential feature selection: a greedy approach that selects the best feature according to 5-fold cross-validation accuracy on every iteration until a set number of features is selected. The models are trained on 20 best features selected this way.

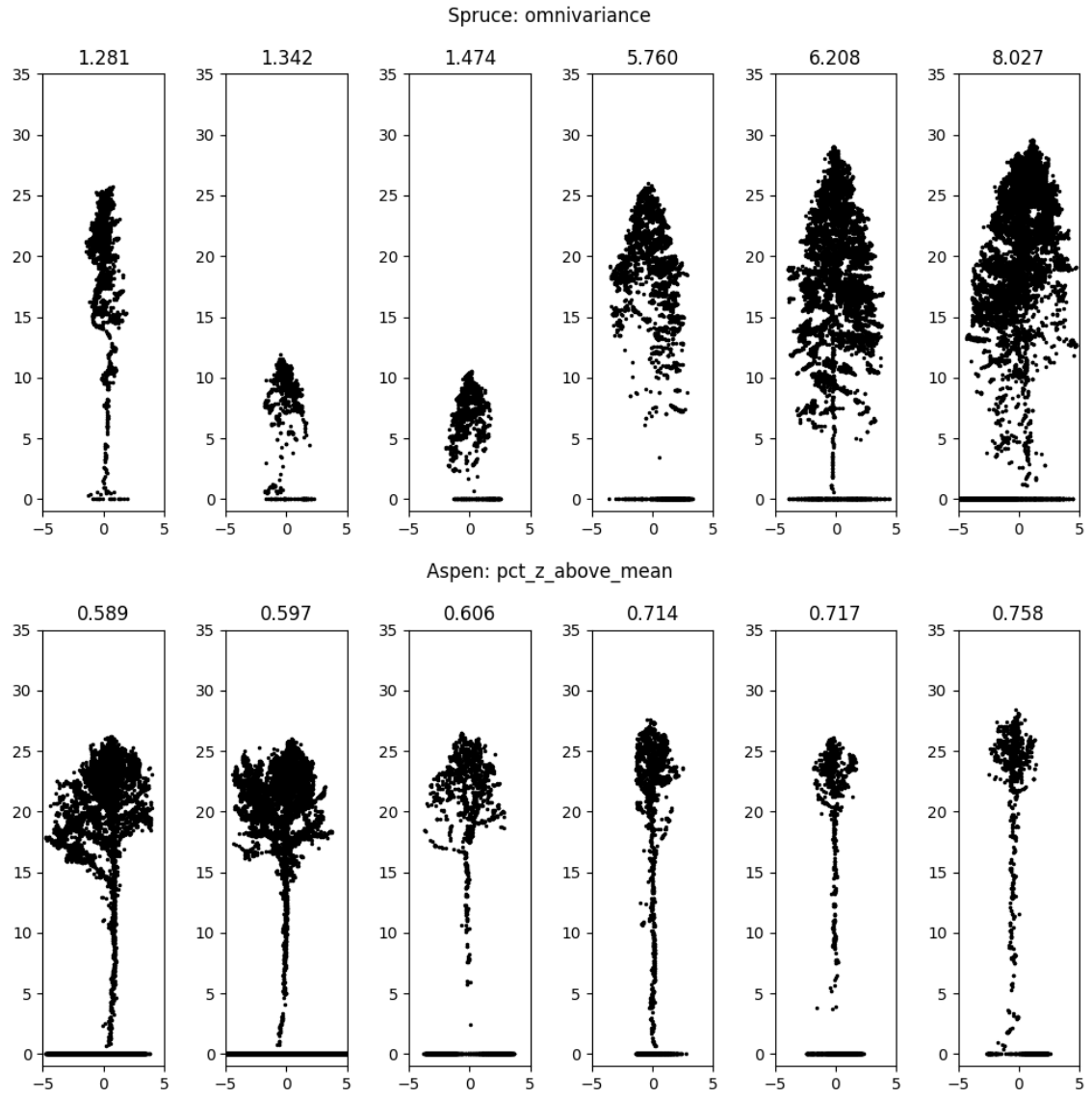


Figure 3-19: Visualizations of how values of commonly used features calculated for the whole tree cloud map to the shapes of individual trees. **Top**: Effect of the shape of spruce on omnivariance. **Bottom**: Effect of on the shape of aspen on percent of points higher than mean height. Figure reused from Dubrovin and Fortin (2024). Note the presence of ground points, which are filtered out before calculating features for the models.

To make the trees more closely resemble the trees standing in dense clusters, the same height-dependent dropout function that is used for augmenting synthetic forests is used. This dropout, together with a small amount of random jitter, is also used to create additional samples for training the models, as the original dataset size is very small. Great care is taken to only create additional samples within the training set, to avoid any data leakage. Every training set sample is repeated 3 times with random jitter with maximum amplitude of 20 cm and a dropout applied.

Two models are used as proof of concept for the proposed framework: a classifier that predicts the tree species, and a regressor that estimates a tree's diameter at breast height. Random Forest models are used in both cases. The number of trees is set to 200. For the classification model, class weights are assigned to every example that are inversely proportional to the class frequency in the data during model fitting.

3.6 Matching detected trees to ground truth

To evaluate the results of any tree detection system against field inventory data, an automated and deterministic method for matching detected tree candidates to the ground truth trees is needed. It should determine which ground truth trees, if any, the detected candidates correspond to. It should also classify both the trees and the candidates as either a true positive, meaning that the ground truth tree was successfully detected, a false negative, meaning that a ground truth tree was not detected, or a false positive, meaning a tree was detected when there is none.

I use the same matching procedure that is described and implemented in Dubrovin et al. [2024]. It considers the locations and heights of the trees, and falls back to using only locations when the height is not available for the ground truth tree. It is parametrized by the maximum allowed 2D distance and the maximum height difference between a detected and a ground truth tree to consider them a match. First, it constructs a distance matrix between the ground truth tree locations and the detected candidates and filters out the pairs for which the distance is larger than the allowed maximum. It then iterates over the remaining pairs in the order of

increasing distance, and marks the pairs with suitable heights in which both trees are not yet assigned a class as a true positive. The pairs in which one of the trees is already assigned a class are marked as either a false positive or a false negative, and the pairs in which both trees are assigned a class are skipped. Finally, it marks all unmatched ground truth trees as false negatives, and all unmatched candidate trees as false positives.

Based on the results of the matching algorithm, a set of metrics that are often used to evaluate the results of tree detection can be calculated. The same metrics are usually used for evaluation in classification problems and thus are well-known. The first commonly used metric is recall, which is the proportion of the ground truth trees that were detected. The second metric is precision, which is the proportion of candidates that are correct. Both are important, and describe a detection system from different perspectives. To combine them into a single metric, F_1-score is often used, which is the harmonic mean of precision and recall that provide a balanced measure of the system performance. The formulas for the metrics are as follows:

$$\text{recall} = \frac{N_{\text{true positives}}}{N_{\text{trees}}} = \frac{N_{\text{true positives}}}{N_{\text{true positives}} + N_{\text{false negatives}}}$$

$$\text{precision} = \frac{N_{\text{true positives}}}{N_{\text{candidates}}} = \frac{N_{\text{true positives}}}{N_{\text{true positives}} + N_{\text{false positive}}}$$

$$F_1\text{-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Another commonly used metric to evaluate the results of tree detection approaches is the average distance between a detected and a ground truth tree. This metric, however, has limitations that are important to keep in mind when analyzing the results. The first important limitation is that its value is limited in how low it can get by the difference in nature of the compared detections and ground truth field inventory. During the inventory, the trees' coordinates are recorded at their trunks, approximately at breast height, which is almost at the ground level compared to the height of the trees. The detections, on the other hand, happen with the perspective from above, and correspond to tree tops. Planar distance between the bottom of

the trunk and the top of the canopy can be significant, especially if trees are even slightly tilted, or for broadleaf species with wide and irregular canopies, or both. So the average distance between predicted trees and their ground truth counterparts will rarely be zero, even for a perfect detection system. The other limitation is that that distance is also limited from above by the parameters used for the matching algorithm. As mentioned, one of the parameters is the maximum distance allowed between a candidate and an actual tree to consider them a match. Thus, the average distance metric is limited from both directions, but in ways that are not immediately obvious. I do calculate and report it in the results section, but the reader is advised to keep the limitations in mind.

3.7 Application of the framework

The framework is applied in two steps. First, the tree segmentation network is applied in a sliding window of the same size that was used for patch generation during its training, with overlap to be able to combine the results into a single prediction. The continuous predictiond of the regression model are rounded to get the integer labels for every point. The results are merged to create a single point cloud, and a postprocessing routine aimed to transform per-window tree IDs to global tree IDs is run. It traces the overlapping points to find the labels that need to be updated in the later window, and adds a large number to predictions with remaining labels in the later window. Second, the segments identified by predicted integer labels are extracted, selected feature sets are calculated for each one, and each one is passed through the collection of attribute prediction models to get per-tree predictions.

Chapter 4

Results

This chapter describes the results of each stage of the framework preparation and the validation approach used to verify its applicability and effectiveness on realistic data.

4.1 Tree segmentation network training results

The results of training a tree segmentation PointNet++ are challenging to evaluate on their own because there are no inherently good metrics that would put the performance into easily understandable numbers. As was mentioned in Section 3.4.3, during training model checkpointing is set up to track average validation accuracy – the proportion of points for which the final predicted label is correct. However, the predictions are quite noisy, resulting in low accuracies even for results that are not bad. Figure 4-1 shows the prediction on a sample from the validation dataset used to monitor the training process, and Figure 4-2 shows the same prediction in 3D. The accuracy of the model on the figures is 33.6%, and it is around the best accuracy of all the experiments I ran, excluding ones with simpler data.

Overall, the final model was selected less based on raw numbers such as maximum average validation accuracy or minimum average validation loss, but by expecting the visualizations of the predictions on a subset of samples from the validation dataset.

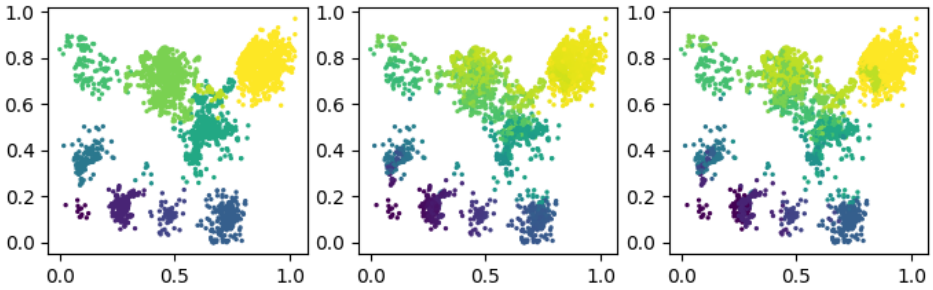


Figure 4-1: A top-down view of an example prediction of the tree segmentation PointNet++ on a sample from the validation dataset. **Left:** Points colored by label – tree ID within the patch. **Middle:** Points colored by the raw prediction of the network (the regression head outputs a continuous prediction for every point). **Right:** Points colored by the rounded prediction – converted from the continuous to the final integer tree ID.

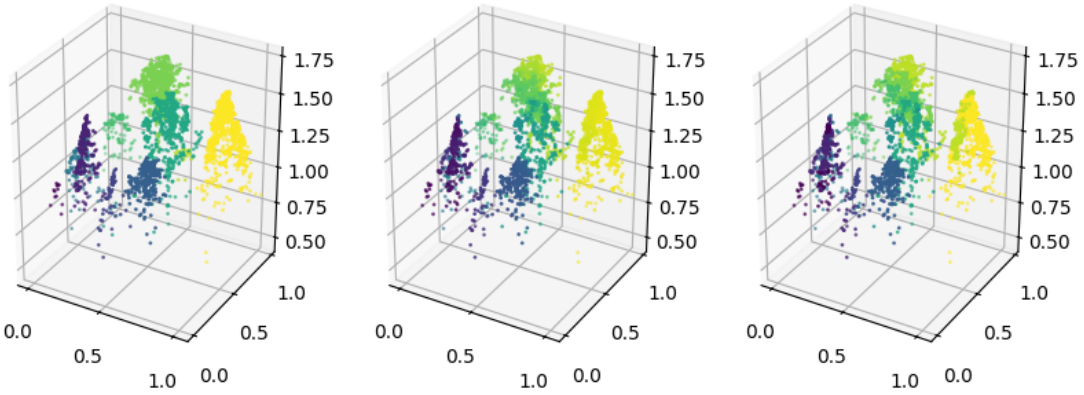


Figure 4-2: A 3D view of an example prediction of the tree segmentation PointNet++ on a sample from the validation dataset. **Left:** Points colored by label – tree ID within the patch. **Middle:** Points colored by the raw prediction of the network (the regression head outputs a continuous prediction for every point). **Right:** Points colored by the rounded prediction – converted from the continuous to the final integer tree ID.

4.1.1 Effect of dropout augmentation

An essential part of the framework is training of the tree segmentation network on synthetic forest patches, which requires heavy use of augmentations to make the data look more realistic and make the model applicable to real-world data. The most important augmentation that addresses the selection bias present in the dataset of individual tree point clouds is the height-dependent dropout described in Section 3.4.2. It would undoubtedly be beneficial to conduct a comprehensive exploration of the effect this augmentation itself and its parameters have on the results of the tree segmentation task specifically and the framework overall. Unfortunately, this thesis does not go into that much detail on that topic because of the time constraints. Still, despite not being properly recorded for the purpose of comprehensive qualitative evaluation, there were many experiments ran with various parameters, and their results are summarized qualitatively below.

Not using the dropout augmentation at all makes the task of segmenting the patches significantly easier – almost all the trees are complete, with a clearly separable trunks. Average validation accuracies in the setup without the augmentation are significantly higher, and validation loss is significantly lower. However, this does not translate to improved results when applying the network to real data, since the training data looks very different when the augmentation is not applied. In such a setup, during inference the network essentially operates completely outside the distribution of its training dataset, making the task impossible and resulting in almost random assignment of IDs to the points.

The sigmoid used to calculate the probability threshold for the dropout is parametrized by scale, which controls how steep the increase in the probability is with height, and shift, which controls the position of the incline. Examples of threshold curves for different values of the parameters are shown in Figure 3-17 and Figure 3-18. Milder variations, which are characterized by lower positions of the incline and more gradual slopes, as in Figure 3-17, have similar effects to not using the augmentations at all. When not enough points are removed, the task remains simple, leading to higher training metrics and worse performance on real data. As the dropout becomes more aggressive, with the position of the incline higher and slopes steeper,

as in Figure 3-18, the training data becomes more similar to the real data. It leads to the decrease in the average validation training accuracy and increase in values of the loss function during training, but the performance translates much better to the real data. At some point, the dropout becomes too aggressive, almost completely destroying the individual trees and making their shapes almost unrecognizable. The training metrics continue to decline, as the task becomes even more complicated, but so does the performance on real data, since the augmentation no longer makes the training data look similar to the real data.

The optimal parameters for the augmentation were selected by manual search. Automating their selection is not immediately available, since there are no direct metrics that tie the performance of the network during training to its performance on the validation data.

4.2 Attribute prediction model training results

To evaluate the performance of the attribute prediction models, 10-fold stratified cross validation was used, as well as a separate 40% hold-out set. For the tree species classification model, the stratification was performed using the labels directly. For the diameter at breast height regression model, the stratification was performed on the label split into 5 equal-width buckets across the range.

Tree species classification

The tree species classification model was evaluated using accuracy and macro F_1-score (an average of F_1-scores across all the classes). Figure 4-3 shows out-of-fold metrics across all ten folds of cross-validation. The average accuracy is 0.71, with a standard deviation of 0.06. The average macro F_1-score is 0.70, with a standard deviation of 0.07. The model is overall relatively consistent across all ten folds, showing that the cross-validation metrics estimates are reliable.

The holdout set was used to take a more in-depth look at the classification results. Figure 4-4 shows the confusion matrix for the holdout set, with values normalized by row. The set is relatively small – 158 examples, but enough to see some patterns

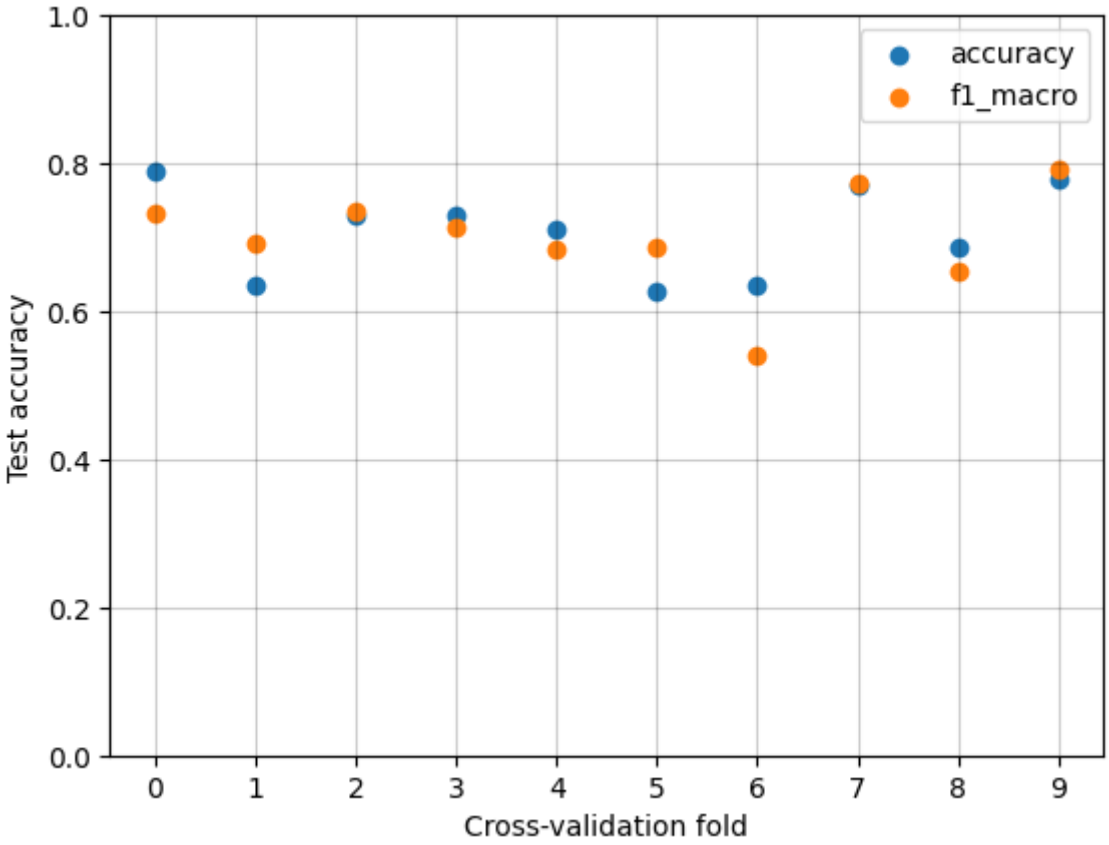


Figure 4-3: 10-fold cross-validation results of the species classification Random Forest model. The average accuracy is 0.71 with a standard deviation of 0.06. The average macro F-score is 0.70 with a standard deviation of 0.07. The model is overall stable across all of the folds.

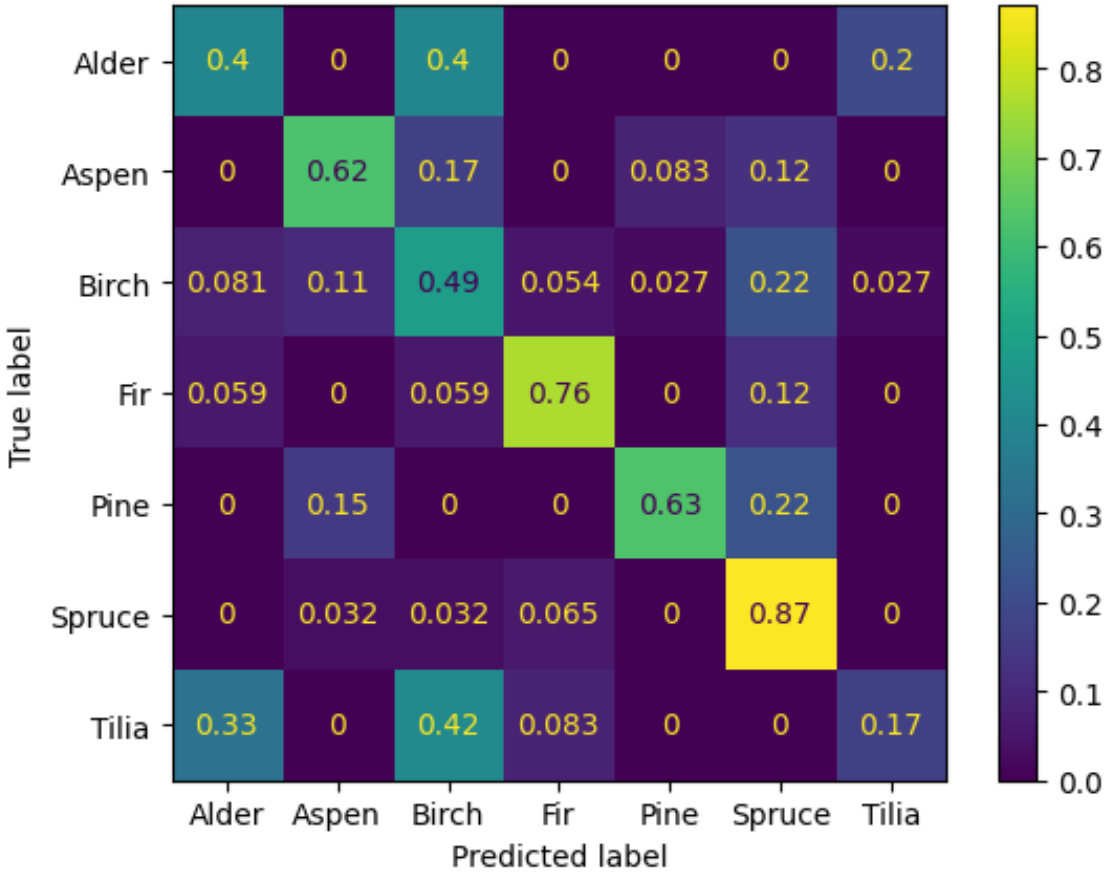


Figure 4-4: Confusion matrix for the tree species classification Random Forest. Calculated on the 40% holdout set of 158 examples. Values normalized by row (represent producer accuracy). The model confuses deciduous species a lot more than coniferous ones.

emerge in the predictions. The best overall species is spruce, having the larges accuracy. All coniferous species have higher accuracies than all deciduous species. The model often confuses deciduous species, classifying alder as birch or tilia, aspen as birch, tilia as alder or birch. This behavior is to be expected, as the shapes of deciduous species are very similar to each other and are complex in general. The overall accuracy in the holdout set is 61%.

Diameter at breast height regression

Diameter at breast height regression model was evaluated using root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination R^2. Formulas for the metrics are provided below, with y_i representing the true dbh

value, $\hat{y}_i$ – the corresponding prediction, and $\overline{y}$ – the average dbh. The dataset here is smaller than the one for tree species classification because diameter at breast height is only available for trees from the Lysva field inventory, while species are available for all trees in the individual tree point cloud dataset, some of which are from other surveys in the region.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}$$

Figure 4-5 shows out-of-fold metrics across all ten folds of cross-validation. Both the mean squared error and the mean absolute error are stable across the folds, with average RMSE 4.55 centimeters with a standard deviation of 0.19 and average MAE 3.49 centimeters with a standard deviation of 0.38. Coefficient of determination R^2 is less stable. Even though most folds have it around 0.6 to 0.8, there are two folds where it drops to below 0.2 and even close to 0, meaning that the model there is no better than predicting the mean dbh for every input. This drops the average R^2 to 0.53 with a standard deviation of 0.24.

Similar to the classification model, the holdout set was used to take a more in-depth look at the regression results. Here the holdout set is even smaller than for regression – 68 samples. Figure 4-6 offers a collection of diagnostic visualizations aimed to help better understand the regression performance. On the top left is the residual plot, where the differences between the actual dbh and the predicted dbh are plotted against the actual dbh values. Ideally, the points should be around a horizontal line at zero, shown in black, without any patterns. This would mean that the size of the error does not depend on the value of dbh. The resulting dbh regression model shows a trend, which means it tends to overestimate low dbh values and underestimate high dbh values. A related plot is on the top right, showing the distribution of the residuals across the holdout set. It should be centered around

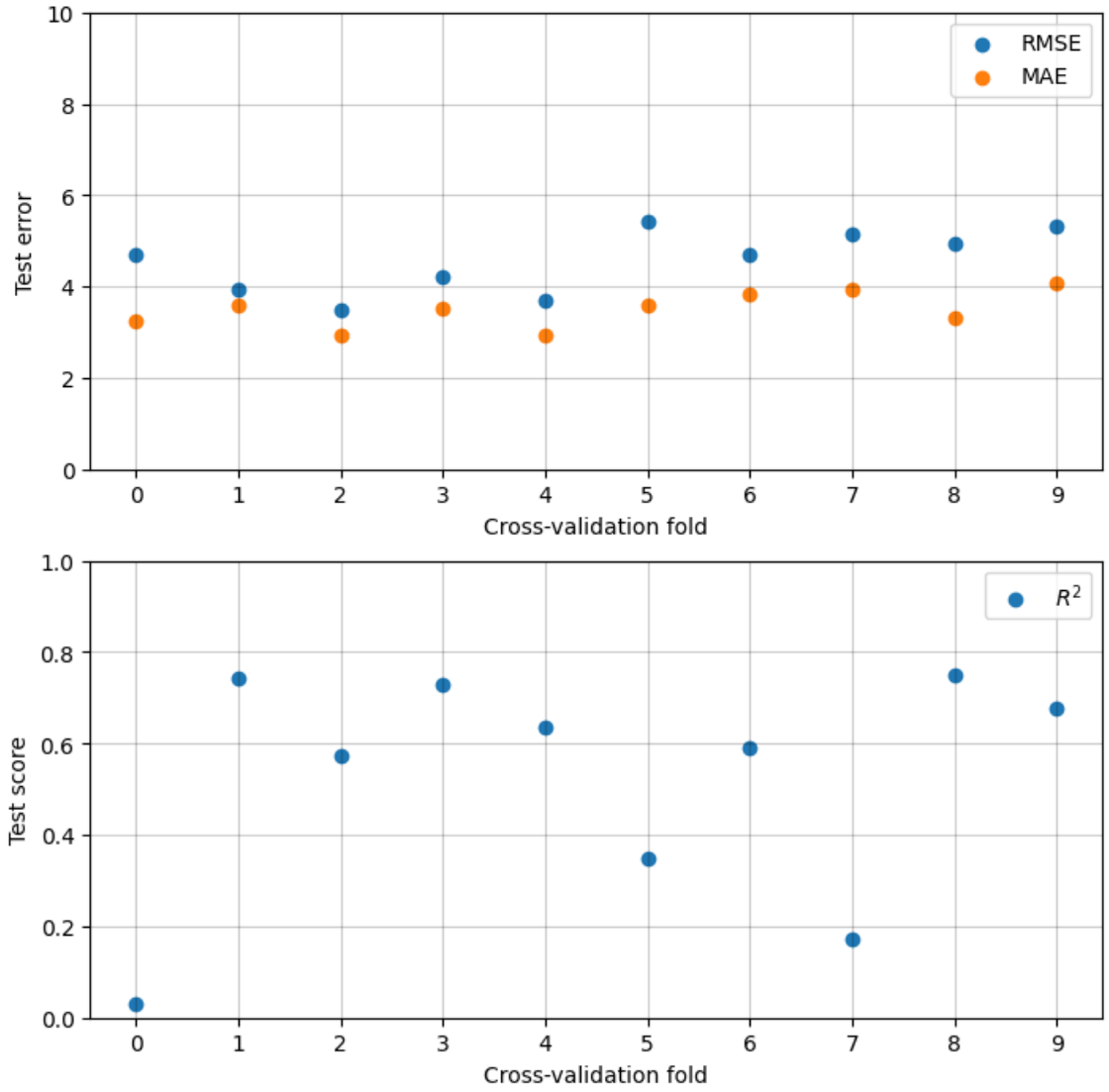


Figure 4-5: 10-fold cross-validation results of the dbh regression Random Forest model. The average RMSE is 4.55 centimeters with a standard deviation of 0.19. The average MAE is 3.49 centimeters with a standard deviation of 0.38. The average coefficient of determination R^2 is 0.53 with a standard deviation of 0.24. The model is relatively stable in terms of the errors, but the coefficient of determination is all over the place.

zero and as narrow as possible. On the bottom left is a scatter plot of predicted dbh against the actual dbh. Ideally, this should be a 45-degree line, shown in black. Because of the underestimation on the high end and overestimation in the low end, the points seem to lie on a line with a smaller slope. Although most points are in the middle, where the line is followed a lot more closely. And, finally, a related plot on the bottom right, comparing distributions of the actual dbh and predicted dbh in the holdout set. The same lack of extreme can be observed. The final holdout set RMSE is 5.76 cm, MAE is 4.11, R^2 is 0.52.

Figure 4-7 show the distributions of the field measured dbh in the full Lysva dataset for reference. On the top it show a boxplot with every individual observation plotted as a point, with random jitter along the Y axis to reduce overlap. On the bottom, a histogram and an empirical cumulative density function are shown, along with the mean, median, and standard deviation.

4.3 Validation on the Lysva field inventory data

To validate the framework, the results were evaluated on the Lysva field inventory dataset, described in Section 3.1. The mean X and Y coordinates of each segmented tree were calculated to convert the cloud into a set of points, where each point represents a candidate detected tree. The maximum height of the points in the cluster was used as the height of the detected tree. Then the algorithm described in Section 3.6 was applied to match the detected trees their corresponding the ground truth trees, if any. The matching algorithm was run with maximum distance between a detected tree candidate and a ground truth tree of 5 meters and maximum height difference of 3 meters (ignored for ground truth trees that do not have measured height). Then, precision, recall, and F_1-score were calculated for the detection results, and for the true positive matches the average distance between the detected tree and the actual tree, species classification accuracy and macro F_1-score, and diameter at breast height errors are calculated.

Table 4.1 shows the tree detection metrics. When interpreting the metrics, the reader is advised to keep in mind the limitations of the distance metric mentioned

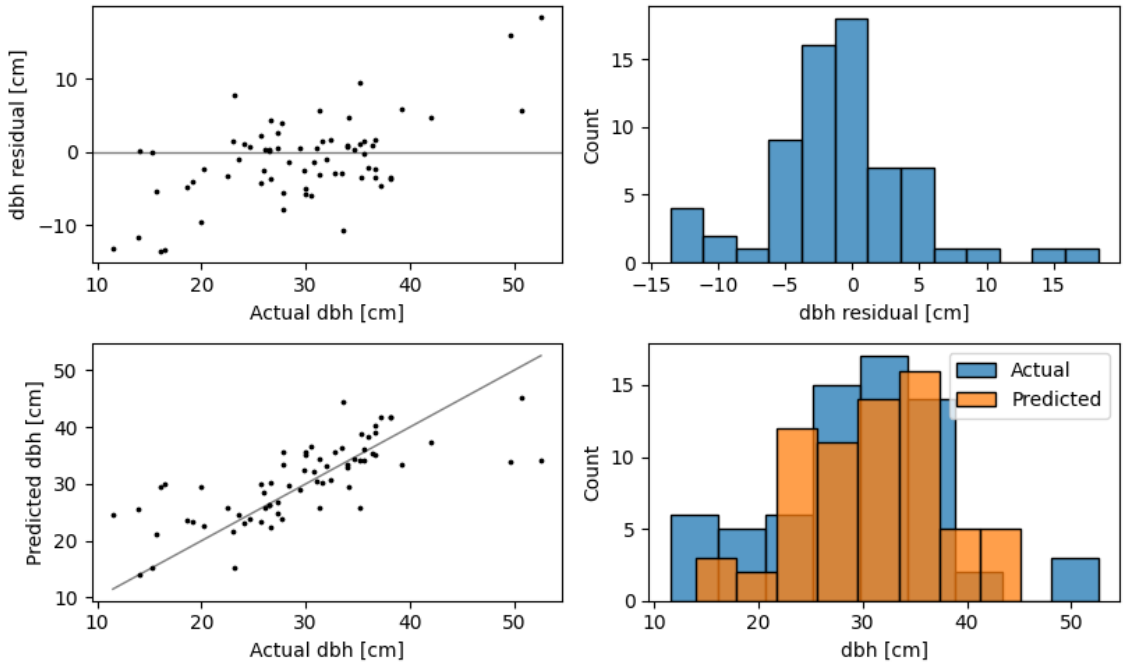


Figure 4-6: Quality assessment plots for the diameter at breast height regression Random Forest. Calculated on a 40% holdout set of 68 examples. **Top Left:** Residual scatter plot: difference between the prediction and the true value vs. the true value. Ideally the spread should be equal across the valu range. **Bottom Left:** Predicted vs. actual scatter plot. Ideally should be as close to a line as possible. **Top right:** Distribution of the residuals. **Bottom right:** Distribution of the predicted and actual values of dbh.

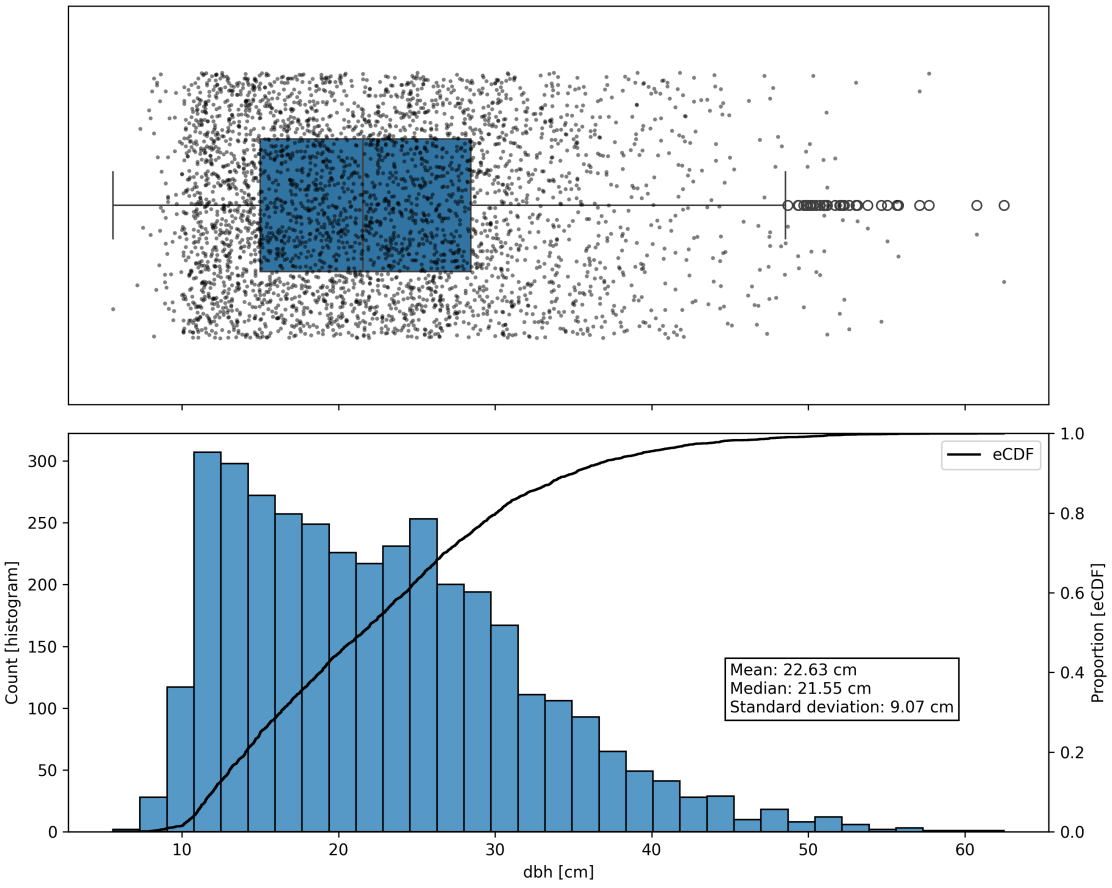


Figure 4-7: Distributions of the field measured dbh in the Lysva dataset. **Top:** Boxplot with individual observations marked by points (jittered along the Y axis to reduce overlap). **Bottom:** Histogram and the empirical cumulative density function of field measured dbh in the Lysva dataset.

Table 4.1: Results of the tree detection across all plots in the Lysva field survey.

Plot	Trees	Dominant type	Point density	Recall	Precision	F_1	Distance
1	420	Deciduous	31.7	0.83	0.63	0.72	0.77
2	365	Deciduous	47.9	0.61	0.72	0.66	0.81
3	332	Deciduous	40.3	0.63	0.49	0.56	0.85
4	261	Coniferous	33.5	0.85	0.50	0.63	1.06
5	208	Coniferous	14.2	0.68	0.55	0.61	0.97
6	290	Coniferous	39.1	0.80	0.52	0.63	0.83
7	408	Deciduous	41.9	0.65	0.62	0.63	0.85
8	341	Coniferous	35.5	0.87	0.57	0.69	0.93
9	459	Coniferous	42.1	0.56	0.65	0.60	0.92
10	518	Deciduous	42.9	0.42	0.92	0.58	0.99
			Average:	0.69	0.62	0.63	0.90

in Section 3.6, and instead focus on the F_1-score as the most informative metric. The overall average F_1-score for tree detection by the described system is 0.63. This approach thus outperforms the basic individual local maximum filter in tree detection in the Lysva region as reported in Dubrovin et al. [2024] by 0.13.

Figure 4-8 shows the confusion matrix for species prediction calculated for true positive matches. Note that willow trees are not included, as the model has not seen any in the training data, and all pine predictions are counted as spruce instead. The pattern of the confusion matrix is similar to the evaluation on the holdout set. The overall accuracy is 66.1%. The overall macro F_1-score is 55.4%.

Figure 4-9 show the quality assessment plots for the results of diameter at breast height regression. It's obvious from the plots that the individual tree dataset is not representative of the entire Lysva dataset in terms of dbh, as the range of dbh in there is not wide enough. The distributions show that the individual trees dataset has higher mean dbh values than the overall distribution, which makes the model eager to overestimate a lot. This effect is no doubt connected to the sampling bias mentioned in the description of the individual trees dataset: trees were selected by humans based on how easy they are to extract from a large point cloud. As the result, the predictions outside the range seen by the Random Forest are constant. Within the covered range, however, the behavior is similar to what was observed on

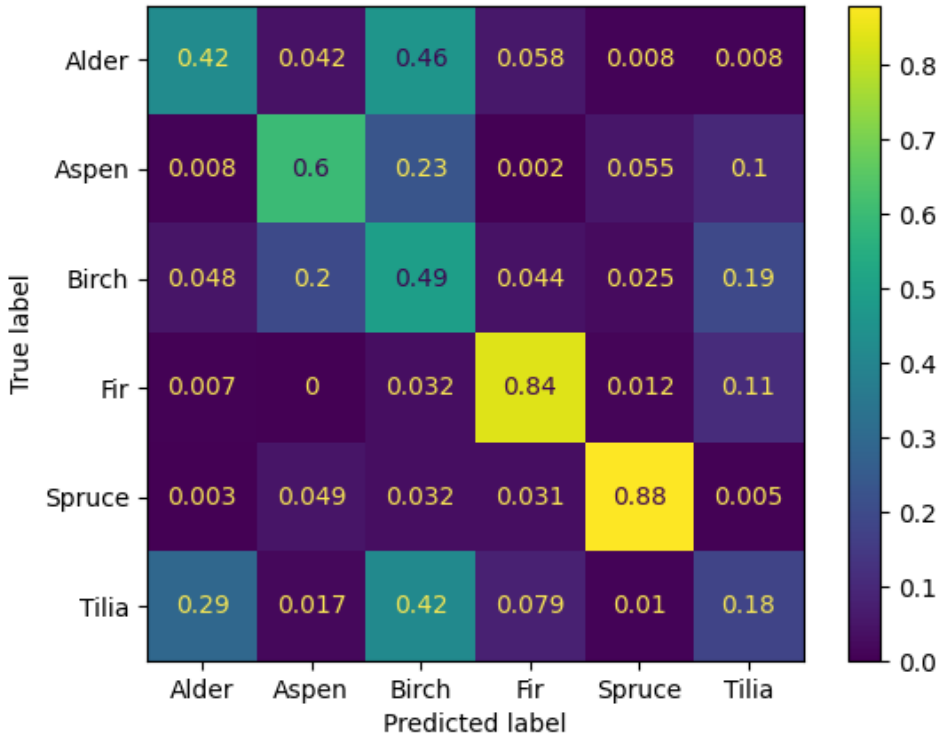


Figure 4-8: Confusion matrix for the tree species classification on true positive detected trees in the Lysva field inventory plots.

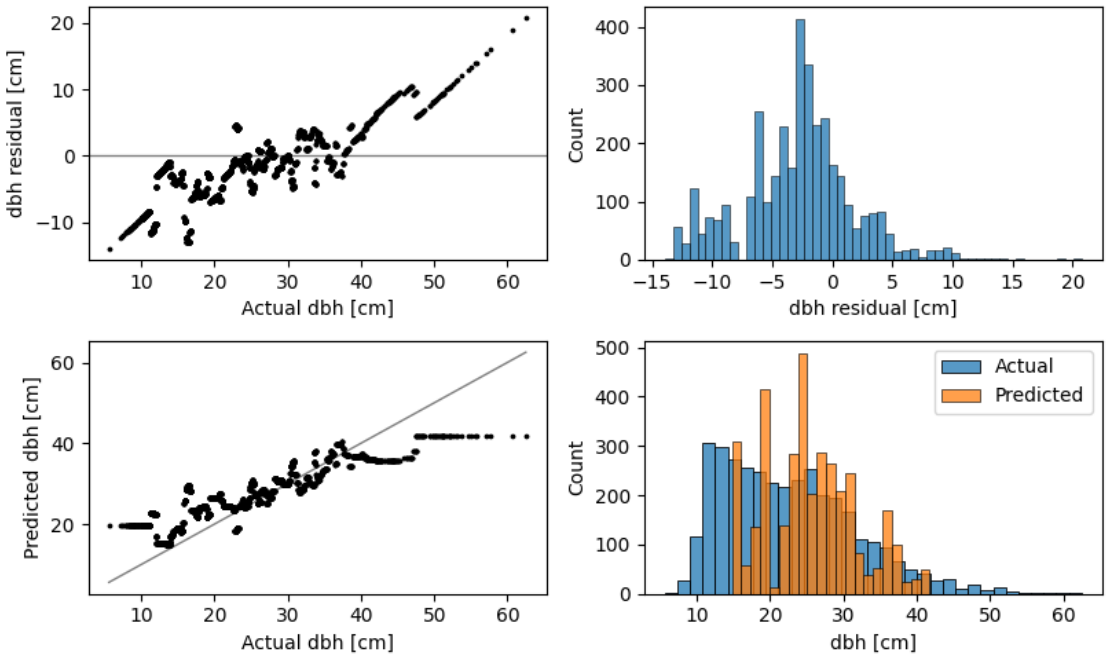


Figure 4-9: Regression quality assessment plots plots for the diameter at breast height regression on true positive detected trees in the Lysva field inventory plots.

the holdout set: a trend in the residual plot indicating over- and underestimation of low and high values, and errors close to zero near the middle of the range. The overall root mean squared error is 5.37 centimeters. The overall mean absolute error is 4.21 centimeters. The overall coefficient of determination R^2 is 0.65.

"A PhD research project is an academic exercise of given effort and length."

Dr. Ron Anderson, PhD adviser of prof. Clement Fortin

Chapter 5

Conclusion

This thesis is dedicated to designing, implementing, and testing of a framework for estimating forest parameters at the scale of individual trees. The proposed framework relies on fusion of UAV LiDAR point clouds and RGB orthophotos data, deep neural networks designed for processing 3D points clouds to segment the data into individual trees, and a collection of parameter-specific classic machine learning models to process the segmented trees and predict forest parameters of interest. An end-to-end implementation is described in detail, with parts of the system evaluated separately, and the overall system validated on a detailed manual field inventory dataset.

The main contributions of the thesis can be summarized as follows:

- Publication of two original open access datasets that will undoubtedly be useful to the research community with a rapidly growing interest in detailed digital forest inventories [Dubrovin and Fortin, 2024, Dubrovin et al., 2024]. I believe the release of data into open access is an important contribution, especially in the current era of commercial machine learning[1] When having good data is an advantage, fewer people are willing to make their data public, which makes it harder for researchers to explore, develop, and compare novel approaches.

- A novel approach to training individual tree segmentation neural networks on

[1] I am purposefully avoiding the term "artificial intelligence" as it seems to have been abused so much as to become practically meaningless, and thus not useful in technical context.

forest patches synthetically generated from individual trees with heavy use of augmentations to make the data more closely represent real data.

- Design, implementation, and experimental verification of the framework mentioned in the first paragraph. The framework is still in the early prototype stage, since the components are far behind the current state of the art. The focus was on creating a proof of concept, thus prioritizing ease of integration over potential performance to show that the end-to-end system has potential.

Coming back to the research question and the hypothesis stated in Section 1.2, I want to address the ways I believe the proposed framework answers it. As already mentioned there, the baselines are the manual forest inventory and the widely used area-based approach described in Section 2.3. The cost reduction mainly comes from the choice of the platform for remote sensing observations. Using UAVs is cheaper than purchasing high-resolution data from satellites or planes, and it offers a lot more control over the acquisition parameters, although it requires trained operators. The effort reduction mainly comes from a dramatic decrease in the amount of required field inventory data, which is very labor-intensive to collect. UAVs also allow covering huge areas easily by a few people, unlike fully manual inventories or those based on terrestrial LiDAR measurements, where a forester or an operator needs to cover the entire area of interest by foot.

The reported experimental results achieved by the implemented system are by no means state of the art, but they do serve as a proof of concept. The current state of the system, described in the previous chapter, can be seen as a baseline, since it prioritizes simple components to show that the overall system has potential. And I believe the resulting framework has potential, with specific steps I would try if I had unlimited time, listed in the next section.

5.1 Potential further improvements

I see numerous ways to potentially improve the results of the proposed system. Some of the most important are as follows:

- Use of more advanced tree segmentation network architectures. As mentioned, the PointNet++ is a pioneering model in the field of deep learning on 3D point clouds that was chosen because it is relatively easy to implement and work with. It is still competitive in many tasks, but the field of deep learning is evolving rapidly, and many more powerful architectures were suggested and proven since its release. The framework will definitely benefit from a better, more efficient tree segmentation network.

- Use of more advanced feature extraction for RGB orthophoto-based features. For the same reason the PointNet++ was chosen as a baseline for tree segmentation, a very basic approach was used to extract features from RGB RGBorthophotos. Despite being better than just RGB colors, these features still offer very limited representation power. Instead, convolutional neural networks specialized in creating good representations for downstream tasks can be used.

- Use of open-access datasets for pretraining and additional validation. At its current state, the tree segmentation network learns to perform a very hard task from scratch on a dataset of a very limited size. I believe it can benefit from first training on similar, but larger dataset, for example, the NeonTreeEvaluation Benchmark [Weinstein et al.]. It is briefly described in the comparison subsection of the Lysva field inventory section. The bounding boxes from the orthophoto can be used to label the point cloud for tree segmentation. The labels will be noisy, but pretraining on a large amount of noisy data often shows good results. Including other data for validation will also make it significantly more reliable.

- Use of more sophisticated techniques for creating synthetic forest patches. Both the sampling of the trees and the placement onto the patch can be improved. For example, sampling can take into consideration the species of the trees, to make the patches more realistic: some fully coniferous, some fully deciduous, some mixed in various proportions. Placing of the trees can use packing algorithms, or try to mimic the spatial distribution of trees observed

in the inventory, where the tree density is not constant.

- Use of more sophisticated augmentations to make the synthetic forest patches look as close as possible to real forest patches. For example, the height-depended dropout can be improved to take into account canopy density and canopy overlap, since dense and overlapping canopies are more likely to block the laser. A shorter tree almost completely covered from above by a larger tree should have fewer points overall, not just at low heights relative to its highest point. It will also be beneficial to generate patches slightly larger than the desired target size and crop them to it randomly as an augmentation. This will mimic the way the data will come to the model during inference, where trees on the edges of the patch are highly likely to be cropped.

- Use of a more careful approach to training attribute prediction models. The results of the second step can be further improved by adopting a more careful approach to modeling, including a more careful approach to feature engineering and model selection.

- Use of synthetic data for pretraining and fine-tuning on the real data. The most common way to enhance training deep learning models with synthetic data usually consist of pretraining on large amount of synthetic data and then careful fine-tuning on the small amount of real data. The current iteration of the dataset does not have labeling to enable direct fine-tuning, but it is possible to create such labels to enable this in the future. Release of the dataset into open access allows the community to enhance the it in many ways, and adding such labels would be one of the most useful ones.

Bibliography

Global Forest Resources Assessment. FAO, 2020. ISBN 978-92-5-132974-0. doi:10.4060/ca9825en. URL http://www.fao.org/documents/card/en/c/ca9825en.

The State of the World's Forests. FAO and UNEP, 2020. ISBN 978-92-5-132419-6. doi:10.4060/ca8642en. URL http://www.fao.org/documents/card/en/c/ca8642en.

Raf Aerts and Olivier Honnay. Forest restoration, biodiversity and ecosystem functioning. *BMC Ecology*, 11(1):29, 2011. ISSN 1472-6785. doi:10.1186/1472-6785-11-29. URL http://bmcecol.biomedcentral.com/articles/10.1186/1472-6785-11-29.

J.J. Allaire, Charles Teague, Carlos Scheidegger, Yihui Xie, and Christophe Dervieux. Quarto, February 2024. URL https://github.com/quarto-dev/quarto-cli.

Matthew J. Allen, Stuart W. D. Grieve, Harry J. F. Owen, and Emily R. Lines. Tree species classification from complex laser scanning data in Mediterranean forests using deep learning. *Methods in Ecology and Evolution*, September 2022. ISSN 2041-210X. doi:10.1111/2041-210X.13981. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13981.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, April 2024. doi:10.1145/3620665.3640366. URL https://pytorch.org/assets/pytorch2-2.pdf.

Isabel Arenas-Corraliza, Ana Nieto, and Gerardo Moreno. Automatic mapping of tree crowns in scattered-tree woodlands using low-density LiDAR data and infrared imagery. *Agroforestry Systems*, 94(5):1989–2002, October 2020. ISSN 0167-4366, 1572-9680. doi:10.1007/s10457-020-00517-2. URL https://link.springer.com/10.1007/s10457-020-00517-2.

Mélaine Aubry-Kientz, Anthony Laybros, Ben Weinstein, James G. C. Ball, Toby Jackson, David Coomes, and Grégoire Vincent. Multisensor Data Fusion for Improved Segmentation of Individual Tree Crowns in Dense Tropical Forests. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:3927–3936, 2021. ISSN 2151-1535. doi:10.1109/JSTARS.2021.3069159. URL https://ieeexplore.ieee.org/abstract/document/9387530.

Mattia Balestra, Suzanne Marselis, Temuulen Tsagaan Sankey, Carlos Cabo, Xinlian Liang, Martin Mokroš, Xi Peng, Arunima Singh, Krzysztof Stereńczak, Cedric Vega, Gregoire Vincent, and Markus Hollaus. LiDAR Data Fusion to Improve Forest Attribute Estimates: A Review. *Current Forestry Reports*, 10(4):281–297, June 2024. ISSN 2198-6436. doi:10.1007/s40725-024-00223-7. URL https://link.springer.com/10.1007/s40725-024-00223-7.

Saifullahi Aminu Bello, Shangshu Yu, Cheng Wang, Jibril Muhmmad Adam, and Jonathan Li. Review: Deep Learning on 3D Point Clouds. *Remote Sensing*, 12 (11):1729, May 2020. ISSN 2072-4292. doi:10.3390/rs12111729. URL https://www.mdpi.com/2072-4292/12/11/1729.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, January 2006. URL https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/.

Lucienne T. M. Blessing and Amaresh Chakrabarti. *DRM, a Design Research Methodology*. Springer, Dordrecht London, 2009. ISBN 978-1-84882-587-1.

Marc Bouvier, Sylvie Durrieu, Richard A. Fournier, and Jean-Pierre Renaud. Generalizing predictive models of forest inventory attributes using an area-based approach with airborne LiDAR data. *Remote Sensing of Environment*, 156:322–334, January 2015. ISSN 0034-4257. doi:10.1016/j.rse.2014.10.004. URL https://www.sciencedirect.com/science/article/pii/S0034425714004003.

Jeffery Burley, John Youngquist, and Julian Evans, editors. *Encyclopedia of Forest Sciences*. Elsevier, Oxford, 1st edition, 2004. ISBN 978-0-12-145160-8.

Andrew Burt, Mathias Disney, and Kim Calders. Extracting individual trees from lidar point clouds using *treeseg*. *Methods in Ecology and Evolution*, pages 2041–210X.13121, December 2018. ISSN 2041-210X, 2041-210X. doi:10.1111/2041-210X.13121. URL https://onlinelibrary.wiley.com/doi/10.1111/2041-210X.13121.

Ward W Carson, Hans-Erik Andersen, Stephen E Reutebuch, and Robert J McGaughey. LiDAR applications in forestry—An overview. In *ASPRS Annual Conference Proceedings*, page 4, 2004.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Rim Douss and Imed Riadh Farah. Extraction of individual trees based on Canopy Height Model to monitor the state of the forest. *Trees, Forests and People*, 8: 100257, June 2022. ISSN 26667193. doi:10.1016/j.tfp.2022.100257. URL https://linkinghub.elsevier.com/retrieve/pii/S2666719322000644.

Ivan Dubrovin and Clement Fortin. An Exploration of Properties of Point Clouds of Individual Trees Extracted From a Larger UAV Lidar Survey. In *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 4503–4506, Athens, Greece, July 2024. IEEE. ISBN 9798350360325. doi:10.1109/IGARSS53475.2024.10641061.

Ivan Dubrovin and Anton Ivanov. Remote Sensing Evidence for the Harmful Algal Bloom Explanation of the Ecological Situation in Kamchatka in Autumn of 2020. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 6764–6767, Kuala Lumpur, Malaysia, July 2022. IEEE. ISBN 978-1-66542-792-0. doi:10.1109/IGARSS46834.2022.9884841.

Ivan Dubrovin, Clement Fortin, and Alexander Kedrov. An open dataset for individual tree detection in UAV LiDAR point clouds and RGB orthophotos in dense mixed forests. *Scientific Reports*, 14(1):21938, September 2024. ISSN 2045-2322. doi:10.1038/s41598-024-72669-5.

Lothar Eysn, Markus Hollaus, Eva Lindberg, Frédéric Berger, Jean-Matthieu Monnet, Michele Dalponte, Milan Kobal, Marco Pellegrini, Emanuele Lingua, Domen Mongus, and Norbert Pfeifer. A Benchmark of Lidar-Based Single Tree Detection Methods Using Heterogeneous Forest Data from the Alpine Space. *Forests*, 6(5):1721–1747, May 2015. ISSN 1999-4907. doi:10.3390/f6051721. URL https://www.mdpi.com/1999-4907/6/5/1721.

Timothy J Fahey, Peter B Woodbury, John J Battles, Christine L Goodale, Steven P Hamburg, Scott V Ollinger, and Christopher W Woodall. Forest carbon storage: Ecology, management, and policy. *Frontiers in Ecology and the Environment*, 8(5):245–252, 2010. ISSN 1540-9309. doi:10.1890/080169. URL https://onlinelibrary.wiley.com/doi/abs/10.1890/080169.

William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL https://github.com/Lightning-AI/lightning.

Tom G. Farr and Mike Kobrick. Shuttle radar topography mission produces a wealth of data. *Eos, Transactions American Geophysical Union*, 81(48):583–585, November 2000. ISSN 0096-3941, 2324-9250. doi:10.1029/EO081i048p00583. URL https://agupubs.onlinelibrary.wiley.com/doi/10.1029/EO081i048p00583.

Felipe Ferrari, Matheus Pinheiro Ferreira, Cláudio Aparecido Almeida, and Raul Queiroz Feitosa. Fusing Sentinel-1 and Sentinel-2 Images for Deforestation Detection in the Brazilian Amazon Under Diverse Cloud Conditions. *IEEE*

Geoscience and Remote Sensing Letters, 20:1–5, 2023. ISSN 1545-598X, 1558-0571. doi:10.1109/LGRS.2023.3242430. URL https://ieeexplore.ieee.org/document/10046410/.

Matheus Pinheiro Ferreira, Daniel Rodrigues Dos Santos, Felipe Ferrari, Luiz Carlos Teixeira Coelho Filho, Gabriela Barbosa Martins, and Raul Queiroz Feitosa. Improving urban tree species classification by deep-learning based fusion of digital aerial images and LiDAR. *Urban Forestry & Urban Greening*, 94:128240, April 2024. ISSN 16188667. doi:10.1016/j.ufug.2024.128240. URL https://linkinghub.elsevier.com/retrieve/pii/S1618866724000384.

Matthias Fey and Jan Eric Lenssen. Fast Graph Representation Learning with PyTorch Geometric, May 2019. URL https://github.com/pyg-team/pytorch_geometric.

Piers M. Forster, Chris Smith, Tristram Walsh, William F. Lamb, Robin Lamboll, Bradley Hall, Mathias Hauser, Aurélien Ribes, Debbie Rosen, Nathan P. Gillett, Matthew D. Palmer, Joeri Rogelj, Karina Von Schuckmann, Blair Trewin, Myles Allen, Robbie Andrew, Richard A. Betts, Alex Borger, Tim Boyer, Jiddu A. Broersma, Carlo Buontempo, Samantha Burgess, Chiara Cagnazzo, Lijing Cheng, Pierre Friedlingstein, Andrew Gettelman, Johannes Gütschow, Masayoshi Ishii, Stuart Jenkins, Xin Lan, Colin Morice, Jens Mühle, Christopher Kadow, John Kennedy, Rachel E. Killick, Paul B. Krummel, Jan C. Minx, Gunnar Myhre, Vaishali Naik, Glen P. Peters, Anna Pirani, Julia Pongratz, Carl-Friedrich Schleussner, Sonia I. Seneviratne, Sophie Szopa, Peter Thorne, Mahesh V. M. Kovilakam, Elisa Majamäki, Jukka-Pekka Jalkanen, Margreet Van Marle, Rachel M. Hoesly, Robert Rohde, Dominik Schumacher, Guido Van Der Werf, Russell Vose, Kirsten Zickfeld, Xuebin Zhang, Valérie Masson-Delmotte, and Panmao Zhai. Indicators of Global Climate Change 2023: Annual update of key indicators of the state of the climate system and human influence. *Earth System Science Data*, 16(6):2625–2658, June 2024. ISSN 1866-3516. doi:10.5194/essd-16-2625-2024. URL https://essd.copernicus.org/articles/16/2625/2024/.

Yuwen Fu, Yifang Niu, Li Wang, and Wang Li. Individual-Tree Segmentation from UAV–LiDAR Data Using a Region-Growing Segmentation and Supervoxel-Weighted Fuzzy Clustering Approach. *Remote Sensing*, 16(4):608, February 2024. ISSN 2072-4292. doi:10.3390/rs16040608. URL https://www.mdpi.com/2072-4292/16/4/608.

Sean Gillies et al. Rasterio: Geospatial raster I/O for Python programmers. Mapbox, 2013/. URL https://github.com/rasterio/rasterio.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, 2016. ISBN 978-0-262-03561-3.

Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting Point Cloud Shape Classification with a Simple and Effective Baseline. In *Proceedings of*

the 38th International Conference on Machine Learning, pages 3809–3820. PMLR, July 2021. URL https://proceedings.mlr.press/v139/goyal21a.html.

Sarah Graves and Sergio Marconi. IDTReeS 2020 Competition Data, July 2020. URL https://zenodo.org/record/3934932.

Li Guo, Nesrine Chehata, Clément Mallet, and Samia Boukir. Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(1):56–66, January 2011. ISSN 09242716. doi:10.1016/j.isprsjprs.2010.08.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0924271610000705.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, December 2021. ISSN 0162-8828, 2160-9292, 1939-3539. doi:10.1109/TPAMI.2020.3005434. URL https://ieeexplore.ieee.org/document/9127813/.

Johannes N. Hansen, Edward T. A. Mitchard, and Stuart King. Assessing Forest/Non-Forest Separability Using Sentinel-1 C-Band Synthetic Aperture Radar. *Remote Sensing*, 12(11):1899, January 2020. ISSN 2072-4292. doi:10.3390/rs12111899. URL https://www.mdpi.com/2072-4292/12/11/1899.

M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. High-Resolution Global Maps of 21st-Century Forest Cover Change. *Science*, 342(6160):850–853, November 2013. ISSN 0036-8075, 1095-9203. doi:10.1126/science.1244693. URL https://www.science.org/doi/10.1126/science.1244693.

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020. doi:10.1038/s41586-020-2649-2.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.

Svetlana Illarionova, Alexey Trekin, Vladimir Ignatiev, and Ivan Oseledets. Neural-Based Hierarchical Approach for Detailed Dominant Forest Species Classification by Multispectral Satellite Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:1810–1820, 2021. ISSN 1939-1404, 2151-1535. doi:10.1109/JSTARS.2020.3048372. URL https://ieeexplore.ieee.org/document/9311828/.

Svetlana Illarionova, Dmitrii Shadrin, Vladimir Ignatiev, Sergey Shayakhmetov, Alexey Trekin, and Ivan Oseledets. Estimation of the Canopy Height Model From Multispectral Satellite Imagery With Convolutional Neural Networks. *IEEE Access*, 10:34116–34132, 2022. ISSN 2169-3536. doi:10.1109/ACCESS.2022.3161568. URL https://ieeexplore.ieee.org/document/9739688/.

Marek K. Jakubowski, Qinghua Guo, and Maggi Kelly. Tradeoffs between lidar pulse density and forest measurement accuracy. *Remote Sensing of Environment*, 130:245–253, March 2013. ISSN 0034-4257. doi:10.1016/j.rse.2012.11.024. URL https://www.sciencedirect.com/science/article/pii/S0034425712004567.

Yam Bahadur Kc, Qijing Liu, Pradip Saud, Damodar Gaire, and Hari Adhikari. Estimation of Above-Ground Forest Biomass in Nepal by the Use of Airborne LiDAR, and Forest Inventory Data. *Land*, 13(2):213, February 2024. ISSN 2073-445X. doi:10.3390/land13020213. URL https://www.mdpi.com/2073-445X/13/2/213.

Kent Kellogg, Pamela Hoffman, Shaun Standley, Scott Shaffer, Paul Rosen, Wendy Edelstein, Charles Dunn, Charles Baker, Phillip Barela, Yuhsyen Shen, Ana Maria Guerrero, Peter Xaypraseuth, V Raju Sagi, C V Sreekantha, Nandini Harinath, Raj Kumar, Rakesh Bhan, and C. V. H. S. Sarma. NASA-ISRO Synthetic Aperture Radar (NISAR) Mission. In *2020 IEEE Aerospace Conference*, pages 1–21, Big Sky, MT, USA, March 2020. IEEE. ISBN 978-1-72812-734-7. doi:10.1109/AERO47225.2020.9172638. URL https://ieeexplore.ieee.org/document/9172638/.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

Donald E. Knuth. Literate programming. *Comput. J.*, 27(2):97–111, May 1984. ISSN 0010-4620. doi:10.1093/comjnl/27.2.97. URL https://doi.org/10.1093/comjnl/27.2.97.

Hien Phu La, Yang Dam Eo, Anjin Chang, and Changjae Kim. Extraction of individual tree crown using hyperspectral image and LiDAR data. *KSCE Journal of Civil*

Engineering, 19(4):1078–1087, May 2015. ISSN 1976-3808. doi:10.1007/s12205-013-1178-z. URL https://doi.org/10.1007/s12205-013-1178-z.

Laura Elena Cué La Rosa, Camile Sothe, Raul Queiroz Feitosa, Cláudia Maria De Almeida, Marcos Benedito Schimalski, and Dário Augusto Borges Oliveira. Multi-task fully convolutional network for tree species mapping in dense forests using small training hyperspectral data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 179:35–49, September 2021. ISSN 09242716. doi:10.1016/j.isprsjprs.2021.07.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0924271621001787.

Guillaume Lassalle, Matheus Pinheiro Ferreira, Laura Elena Cué La Rosa, and Carlos Roberto de Souza Filho. Deep learning-based individual tree crown delineation in mangrove forests using very-high-resolution satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 189:220–235, July 2022. ISSN 0924-2716. doi:10.1016/j.isprsjprs.2022.05.002. URL https://www.sciencedirect.com/science/article/pii/S0924271622001411.

Qian Li, Baoxin Hu, Jiali Shang, and Hui Li. Fusion Approaches to Individual Tree Species Classification Using Multisource Remote Sensing Data. *Forests*, 14(7): 1392, July 2023. ISSN 1999-4907. doi:10.3390/f14071392. URL https://www.mdpi.com/1999-4907/14/7/1392.

Wenkai Li, Qinghua Guo, Marek K. Jakubowski, and Maggi Kelly. A New Method for Segmenting Individual Trees from the Lidar Point Cloud. *Photogrammetric Engineering & Remote Sensing*, 78(1):75–84, January 2012. ISSN 00991112. doi:10.14358/PERS.78.1.75. URL http://openurl.ingenta.com/content/xref?genre=article&issn=0099-1112&volume=78&issue=1&spage=75.

Xugang Lian, Hailang Zhang, Wu Xiao, Yunping Lei, Linlin Ge, Kai Qin, Yuanwen He, Quanyi Dong, Longfei Li, Yu Han, Haodi Fan, Yu Li, Lifan Shi, and Jiang Chang. Biomass Calculations of Individual Trees Based on Unmanned Aerial Vehicle Multispectral Imagery and Laser Scanning Combined with Terrestrial Laser Scanning in Complex Stands. *Remote Sensing*, 14(19):4715, September 2022. ISSN 2072-4292. doi:10.3390/rs14194715. URL https://www.mdpi.com/2072-4292/14/19/4715.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A Simple State Space Model for Point Cloud Analysis, November 2024. URL http://arxiv.org/abs/2402.10739.

Maciej Lisiewicz, Agnieszka Kamińska, Bartłomiej Kraszewski, and Krzysztof Stereńczak. Correcting the Results of CHM-Based Individual Tree Detection Algorithms to Improve Their Accuracy and Reliability. *Remote Sensing*, 14 (8):1822, April 2022. ISSN 2072-4292. doi:10.3390/rs14081822. URL https://www.mdpi.com/2072-4292/14/8/1822.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440,

2015. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html.

F. R. López Serrano, E. Rubio, F. A. García Morote, M. Andrés Abellán, M. I. Picazo Córdoba, F. García Saucedo, E. Martínez García, J. M. Sánchez García, J. Serena Innerarity, L. Carrasco Lucas, O. García González, and J. C. García González. Artificial intelligence-based software (AID-FOREST) for tree detection: A new framework for fast and accurate forest inventorying using LiDAR point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 113:103014, September 2022. ISSN 1569-8432. doi:10.1016/j.jag.2022.103014. URL https://www.sciencedirect.com/science/article/pii/S1569843222002023.

Chris Lucas, Willem Bouten, Zsófia Koma, W. Daniel Kissling, and Arie C. Seijmonsbergen. Identification of Linear Vegetation Elements in a Rural Landscape Using LiDAR Point Clouds. *Remote Sensing*, 11(3):292, January 2019. ISSN 2072-4292. doi:10.3390/rs11030292. URL https://www.mdpi.com/2072-4292/11/3/292.

Clément Mallet, Frédéric Bretar, Michel Roux, Uwe Soergel, and Christian Heipke. Relevance assessment of full-waveform lidar data for urban area classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6):S71–S84, December 2011. ISSN 09242716. doi:10.1016/j.isprsjprs.2011.09.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0924271611001055.

Gabriela Barbosa Martins, Laura Elena Cué La Rosa, Patrick Nigri Happ, Luiz Carlos Teixeira Coelho Filho, Celso Junius F. Santos, Raul Queiroz Feitosa, and Matheus Pinheiro Ferreira. Deep learning-based tree species mapping in a highly diverse tropical urban setting. *Urban Forestry & Urban Greening*, 64:127241, September 2021. ISSN 1618-8667. doi:10.1016/j.ufug.2021.127241. URL https://www.sciencedirect.com/science/article/pii/S1618866721002661.

Alhassan Mumuni and Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, 16:100258, December 2022. ISSN 2590-0056. doi:10.1016/j.array.2022.100258. URL https://www.sciencedirect.com/science/article/pii/S2590005622000911.

Erik Næsset. Determination of mean tree height of forest stands using airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 52(2):49–56, April 1997a. ISSN 0924-2716. doi:10.1016/S0924-2716(97)83000-6. URL https://www.sciencedirect.com/science/article/pii/S0924271697830006.

Erik Næsset. Estimating timber volume of forest stands using airborne laser scanner data. *Remote Sensing of Environment*, 61(2):246–253, August 1997b. ISSN 0034-4257. doi:10.1016/S0034-4257(97)00041-2. URL https://www.sciencedirect.com/science/article/pii/S0034425797000412.

Ross Nelson, William Krabill, and Gordon MacLean. Determining forest canopy characteristics using airborne laser data. *Remote Sensing of Environment*, 15(3): 201–212, June 1984. ISSN 0034-4257. doi:10.1016/0034-4257(84)90031-2. URL https://www.sciencedirect.com/science/article/pii/0034425784900312.

Mats Nilsson. Estimation of tree heights and stand volume using an airborne lidar system. *Remote Sensing of Environment*, 56(1):1–7, April 1996. ISSN 0034-4257. doi:10.1016/0034-4257(95)00224-3. URL https://www.sciencedirect.com/science/article/pii/0034425795002243.

Abdul Nurunnabi, Felicia Teferle, Debra F. Laefer, Meida Chen, and Mir Masoom Ali. Development of a Precise Tree Structure from LiDAR Point Clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2-2024:301–308, June 2024. ISSN 2194-9034. doi:10.5194/isprs-archives-XLVIII-2-2024-301-2024. URL https://isprs-archives.copernicus.org/articles/XLVIII-2-2024/301/2024/.

Lucas Prado Osco, Mauro Dos Santos De Arruda, José Marcato Junior, Neemias Buceli Da Silva, Ana Paula Marques Ramos, Érika Akemi Saito Moryia, Nilton Nobuhiro Imai, Danillo Roberto Pereira, José Eduardo Creste, Edson Takashi Matsubara, Jonathan Li, and Wesley Nunes Gonçalves. A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 160:97–106, February 2020. ISSN 09242716. doi:10.1016/j.isprsjprs.2019.12.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0924271619302989.

Emre Özdemir. *A Deep Learning Framework for Geospatial Point Cloud Classification*. PhD thesis, Skolkovo Institute of Science and Technology, Moscow, 2021. URL https://www.skoltech.ru/app/data/uploads/2021/12/thesis-2.pdf.

M. Pauly, M. Gross, and L.P. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization, 2002. VIS 2002.*, pages 163–170, Boston, MA, USA, 2002. IEEE. ISBN 978-0-7803-7498-0. doi:10.1109/VISUAL.2002.1183771. URL http://ieeexplore.ieee.org/document/1183771/.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Lutz Prechelt. Automatic early stopping using cross validation: Quantifying the criteria. *Neural Networks*, 11(4):761–767, June 1998. ISSN 0893-6080. doi:10.1016/S0893-6080(98)00010-0. URL https://www.sciencedirect.com/science/article/pii/S0893608098000100.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017a. URL https://arxiv.org/abs/1612.00593.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017b. URL https://proceedings.neurips.cc/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html.

Shaun Quegan, Thuy Le Toan, Jerome Chave, Jorgen Dall, Jean-François Exbrayat, Dinh Ho Tong Minh, Mark Lomas, Mauro Mariotti D'Alessandro, Philippe Paillou, Kostas Papathanassiou, Fabio Rocca, Sassan Saatchi, Klaus Scipal, Hank Shugart, T. Luke Smallman, Maciej J. Soja, Stefano Tebaldini, Lars Ulander, Ludovic Villard, and Mathew Williams. The European Space Agency BIOMASS mission: Measuring forest above-ground biomass from space. *Remote Sensing of Environment*, 227:44–60, June 2019. ISSN 00344257. doi:10.1016/j.rse.2019.03.032. URL https://linkinghub.elsevier.com/retrieve/pii/S0034425719301233.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. doi:10.1007/978-3-319-24574-4_28.

Jean-Romain Roussel, David Auty, Nicholas C. Coops, Piotr Tompalski, Tristan R. H. Goodbody, Andrew Sánchez Meador, Jean-François Bourdon, Florian de Boissieu, and Alexis Achim. lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. *Remote Sensing of Environment*, 251:112061, 2020. ISSN 0034-4257. doi:10.1016/j.rse.2020.112061. URL https://www.sciencedirect.com/science/article/pii/S0034425720304314.

Yifang Shi, Tiejun Wang, Andrew K. Skidmore, and Marco Heurich. Important LiDAR metrics for discriminating forest tree species in Central Europe. *ISPRS Journal of Photogrammetry and Remote Sensing*, 137:163–174, March 2018. ISSN 09242716. doi:10.1016/j.isprsjprs.2018.02.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0924271618300315.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014. URL https://arxiv.org/abs/1409.1556.

Juris Sinica-Sinavskis and Gunta Grube. Forest Stand Volume Estimation by Species from Sentinel-2 and LiDAR Data Using Regression Models. In *2022 18th Biennial Baltic Electronics Conference (BEC)*, pages 1–5, October 2022. doi:10.1109/BEC56180.2022.9935590. URL https://ieeexplore.ieee.org/abstract/document/9935590.

The pandas development team. Pandas-dev/pandas: Pandas. URL https://github.com/pandas-dev/pandas.

Paul Treitz, Kevin Lim, Murray Woods, Doug Pitt, Dave Nesbitt, and Dave Etheridge. LiDAR Sampling Density for Forest Resource Inventories in On-

tario, Canada. *Remote Sensing*, 4(4):830–848, April 2012. ISSN 2072-4292. doi:10.3390/rs4040830. URL https://www.mdpi.com/2072-4292/4/4/830.

Stéfan J. van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. Scikit-image: Image processing in Python. *PeerJ*, 2: e453, June 2014. doi:10.7717/peerj.453. URL https://doi.org/10.7717/peerj.453.

Jonathan Ventura, Camille Pawlak, Milo Honsberger, Cameron Gonsalves, Julian Rice, Natalie L.R. Love, Skyler Han, Viet Nguyen, Keilana Sugano, Jacqueline Doremus, G. Andrew Fricker, Jenn Yost, and Matt Ritter. Individual tree detection in large-scale urban environments using high-resolution multispectral imagery. *International Journal of Applied Earth Observation and Geoinformation*, 130:103848, June 2024. ISSN 15698432. doi:10.1016/j.jag.2024.103848. URL https://linkinghub.elsevier.com/retrieve/pii/S1569843224002024.

Martijn Vermeer, Jacob Alexander Hay, David Völgyes, Zsófia Koma, Johannes Breidenbach, and Daniele Stefano Maria Fantin. Lidar-based Norwegian tree species detection using deep learning, November 2023. URL http://arxiv.org/abs/2311.06066.

Aguida Beatriz Travaglia Viana, Carlos Moreira Miquelino Eleto Torres, Cibele Humel do Amaral, Elpídio Inácio Fernandes Filho, Carlos Pedro Boechat Soares, Felipe Carvalho Santana, Lucas Brandão Timo, and Samuel José Silva Soares da Rocha. Timber Volume Estimation By Using Terrestrial Laser Scanning: Method In Hyperdiverse Secondary Forests. *Revista Árvore*, 46, August 2022. ISSN 0100-6762, 1806-9088. doi:10.1590/1806-908820220000021. URL http://www.scielo.br/j/rarv/a/7zVyPTcnXJSFHkFLfMSp7pJ/?lang=en.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272, 2020. doi:10.1038/s41592-019-0686-2. URL https://doi.org/10.1038/s41592-019-0686-2.

Xizhao Wang, Yanxia Zhao, and Farhad Pourpanah. Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics*, 11(4):747–750, April 2020. ISSN 1868-8071, 1868-808X. doi:10.1007/s13042-020-01096-5. URL http://link.springer.com/10.1007/s13042-020-01096-5.

Zhen Wang, Pu Li, Yuancheng Cui, Shuowen Lei, and Zhizhong Kang. Automatic Detection of Individual Trees in Forests Based on Airborne LiDAR Data with a

Tree Region-Based Convolutional Neural Network (RCNN). *Remote Sensing*, 15 (4):1024, February 2023. ISSN 2072-4292. doi:10.3390/rs15041024. URL https://www.mdpi.com/2072-4292/15/4/1024.

Michael Waskom. Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60):3021, April 2021. ISSN 2475-9066. doi:10.21105/joss.03021. URL https://joss.theoj.org/papers/10.21105/joss.03021.

M. Weinmann, B. Jutzi, and C. Mallet. Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-5-W2:313–318, October 2013. ISSN 2194-9042. doi:10.5194/isprsannals-II-5-W2-313-2013. URL https://isprs-annals.copernicus.org/articles/II-5-W2/313/2013/.

Ben Weinstein, Sergio Marconi, and Ethan White. Data for the NeonTreeEvaluation Benchmark. URL https://zenodo.org/record/5914554.

Ben G. Weinstein, Sergio Marconi, Stephanie Bohlman, Alina Zare, and Ethan White. Individual Tree-Crown Detection in RGB Imagery Using Semi-Supervised Deep Learning Neural Networks. *Remote Sensing*, 11(11):1309, June 2019. ISSN 2072-4292. doi:10.3390/rs11111309. URL https://www.mdpi.com/2072-4292/11/11/1309.

Ben G. Weinstein, Sergio Marconi, Mélaine Aubry-Kientz, Gregoire Vincent, Henry Senyondo, and Ethan P. White. DeepForest: A PYTHON package for RGB deep learning tree crown delineation. *Methods in Ecology and Evolution*, 11 (12):1743–1751, December 2020. ISSN 2041-210X, 2041-210X. doi:10.1111/2041-210X.13472. URL https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.13472.

Karen F. West, Brian N. Webb, James R. Lersch, Steven Pothier, Joseph M. Triscari, and A. E. Iverson. Context-driven automated target detection in 3D data. In Firooz A. Sadjadi, editor, *Defense and Security*, pages 133–143, Orlando, FL, September 2004. doi:10.1117/12.542536. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=844327.

Joanne C. White, Michael A. Wulder, Andrés Varhola, Mikko Vastaranta, Nicholas C. Coops, Bruce D. Cook, Doug Pitt, and Murray Woods. *A Best Practices Guide for Generating Forest Inventory Attributes from Airborne Laser Scanning Data Using the Area-Based Approach*. Canadian Forest Service, Victoria, British Columbia, 2013. ISBN 978-1-100-22385-8.

Phil Wilkes, Mathias Disney, John Armston, Harm Bartholomeus, Lisa Bentley, Benjamin Brede, Andrew Burt, Kim Calders, Cecilia Chavana-Bryant, Daniel Clewley, Laura Duncanson, Brieanne Forbes, Sean Krisanski, Yadvinder Malhi, David Moffat, Niall Origo, Alexander Shenkin, and Wanxin Yang. *TLS2trees* : A scalable tree segmentation pipeline for TLS data. *Methods in Ecology and Evolution*, 14(12):3083–3099, December 2023. ISSN 2041-210X, 2041-210X. doi:10.1111/2041-210X.14233. URL https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233.

Jingru Wu, Qixia Man, Xinming Yang, Pinliang Dong, Xiaotong Ma, Chunhui Liu, and Changyin Han. Fine Classification of Urban Tree Species Based on UAV-Based RGB Imagery and LiDAR Data. *Forests*, 15(2):390, February 2024a. ISSN 1999-4907. doi:10.3390/f15020390. URL https://www.mdpi.com/1999-4907/15/2/390.

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer V2: Grouped Vector Attention and Partition-based Pooling. In *NeurIPS*, 2022.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler Faster Stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wu_Point_Transformer_V3_Simpler_Faster_Stronger_CVPR_2024_paper.html.

Zhengnan Zhang, Tiejun Wang, Andrew K. Skidmore, Fuliang Cao, Guanghui She, and Lin Cao. An improved area-based approach for estimating plot-level tree DBH from airborne LiDAR data. *Forest Ecosystems*, 10:100089, January 2023. ISSN 2197-5620. doi:10.1016/j.fecs.2023.100089. URL https://www.sciencedirect.com/science/article/pii/S2197562023000040.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Zhao_Point_Transformer_ICCV_2021_paper.html?ref=https://githubhelp.com.

Zhen Zhen, Lindi Quackenbush, and Lianjun Zhang. Impact of Tree-Oriented Growth Order in Marker-Controlled Region Growing for Individual Tree Crown Delineation Using Airborne Laser Scanner (ALS) Data. *Remote Sensing*, 6 (1):555–579, January 2014. ISSN 2072-4292. doi:10.3390/rs6010555. URL https://www.mdpi.com/2072-4292/6/1/555.

Liang Zhu, Zhijian Zhao, Chao Lu, Yining Lin, Yao Peng, and Tangren Yao. Dual Path Multi-Scale Fusion Networks with Attention for Crowd Counting, 2019. URL https://arxiv.org/abs/1902.01115.