

# **Hoge Bomen Vangen Veel Wind: Automated Tree Species Classification and Geometry Reconstruction for Urban CFD Modelling**

Noah Petri Alting

March 27, 2025

student number      4828968

1st supervisor      Hugo Ledoux  
2nd supervisor      Clara García-Sánchez



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related work</b>	<b>3</b>
2.1	Vegetation in Urban CFD Modelling . . . . .	3
2.1.1	Vegetation in Urban Digital Twins . . . . .	3
2.1.2	Level of Detail and Vegetation Semantics in CFD . . . . .	3
2.2	Filtering Vegetation from Aerial Lidar . . . . .	3
2.2.1	Generation of Multispectral Features . . . . .	3
2.3	Tree Instance Segmentation . . . . .	4
2.3.1	Pretrained Model Using Aerial Imagery . . . . .	4
2.3.2	Cuboid-Based Tree Delineation . . . . .	5
2.3.3	Tree Delineation from Lidar and Hyper-Spectral Imagery . . . . .	6
2.4	Inferring Species from Airborne Lidar . . . . .	6
2.4.1	Getting training data . . . . .	6
2.4.2	Machine Learning Methods for Tree Delineation . . . . .	6
2.4.3	Tree-level Features for Model Training . . . . .	7
2.4.4	Leaf Area Index and Leaf Area Density . . . . .	7
2.5	Urban Tree Reconstruction . . . . .	7
2.5.1	Level of Details for Vegetation . . . . .	7
2.5.2	Alpha Shapes . . . . .	8
2.5.3	Voxelisation . . . . .	8
2.5.4	Trunk Reconstruction . . . . .	9
<b>3</b>	<b>Research Goal and Questions</b>	<b>10</b>
<b>4</b>	<b>Methodology</b>	<b>11</b>
4.1	Overview of the Pipeline . . . . .	11
4.2	Data Sources and Preprocessing . . . . .	12
4.3	Tree Instance Segmentation . . . . .	14
4.4	Feature Extraction and Data Matching . . . . .	16
4.5	Species Classification . . . . .	20
4.6	3D Geometry Reconstruction for CFD . . . . .	21
<b>5</b>	<b>Time planning</b>	<b>23</b>
5.1	Supervision . . . . .	23
<b>6</b>	<b>Tools and datasets used</b>	<b>23</b>
<b>7</b>	<b>Appendix</b>	<b>25</b>

# 1 Introduction

We live in a world where urbanisation continues to increase, with more people moving into cities and living in denser environments. Since 2023, over 93% of the population in the Netherlands resides in urban areas (WorldBank, 2023). Urban analysis plays a key role in helping cities adapt to this growth in healthy, sustainable, and inclusive ways.

Within the human ecosystem, a strong dependency on biodiversity and vegetation is rooted in our biophysical foundation Pickett et al. (2011). As more people move into urban environments, it becomes increasingly important to ensure that this fundamental need for vegetation can be met. Monitoring urban greenery using a digital twin that includes trees can help support this effort. In addition, global warming is leading to rising temperatures worldwide, and urban environments are no exception (ICPP, 2023). The urban heat island (UHI) effect—caused by dense built environments retaining heat—is a well-known challenge in climate adaptation. As shown by Park et al. (2024), Computational Fluid Dynamics (CFD) simulations are increasingly used to analyse and optimise strategies for UHI mitigation, such as urban greening systems that maximise latent heat dissipation.

CFD modelling is widely used to assess wind patterns, thermal comfort, and pollutant dispersion in urban environments—particularly as cities adapt to climate change. Recent literature underlines that vegetation, especially trees, plays an active aerodynamic role in such simulations. Trees can both help or hinder pollutant dispersion in street canyons, depending on their placement and configuration (Buccolieri et al., 2018). Moreover, Rodriguez et al. (2024) show that not just the presence, but also the geometric detail of tree foliage significantly affects local thermal predictions.

Although vegetation is increasingly recognised as important in urban CFD simulations, trees are still often neglected or overly simplified. Specifically, the Level of Detail (LoD) in which tree geometries are modelled, affects local flow patterns (García-Sánchez et al., 2021). This oversimplification—or even exclusion—of trees in CFD simulations is often due to current workflows falling short in integrating vegetation in an end-to-end manner (Somanath et al., 2024).

This brings us to the central goal of this research. The goal of this thesis is to develop a scalable, automated method to reconstruct CFD-ready models of individual urban trees using open-source airborne lidar enriched with RGB and NIR data. The proposed pipeline will produce geometrically realistic tree models, each assigned a species-based porosity value, suitable for integration into CFD simulations. This work bridges a gap between urban-scale tree modelling and aerodynamic applications by combining 3D reconstruction, species classification, and geometry simplification in one end-to-end approach.

The research addresses four main questions: (1) How can individual trees be segmented from an unstructured urban point cloud? (2) Which features support accurate species classification from airborne lidar-derived geometry and reflectance? (3) What machine learning methods perform best for this task? and (4) How can segmented trees be converted into CFD-compatible models with assigned porosity? The approach focuses on Dutch urban areas using AHN4 (aerial) lidar data and public tree inventories, under the assumption that RGB and NIR channels are available. CFD simulations themselves are out of scope; only CFD-ready geometries are delivered.

The remainder of this report is structured as follows: Chapter 2 reviews relevant literature on urban CFD modelling, point cloud processing and segmentation, species classification, and 3D tree geometry reconstruction. Chapter 3 presents the research goal and questions. Chapter

4 outlines the proposed methodology and pipeline. Chapter 5 provides a time planning for the thesis, and Chapter 6 describes the tools and datasets to be used.

This report is part of the MSc Geomatics graduation process and serves as the P2 research proposal submitted in pursuit of the Master of Science degree.

## 2 Related work

### 2.1 Vegetation in Urban CFD Modelling

#### 2.1.1 Vegetation in Urban Digital Twins

Vegetation is increasingly being acknowledged as a relevant factor in urban digital twin development, particularly for applications involving airflow and microclimate modelling. Somanath et al. (2024) present a procedural pipeline for generating 3D city models from geospatial data, aiming to bridge GIS formats with simulation-ready geometry.

Although their focus lies primarily on built structures, the paper highlights that current workflows fall short in integrating vegetation in an end-to-end manner. Their case study is based on a coniferous forest and does not directly address urban vegetation types or species-specific structure. Still, it reinforces that vegetation has aerodynamic relevance and cites earlier work by García Sánchez that shows the importance of semantic differentiation—including vegetation—for reliable CFD outcomes. This underlines the need for more detailed and application-aware representations of trees in urban modelling pipelines.

#### 2.1.2 Level of Detail and Vegetation Semantics in CFD

The impact of level of detail (LoD) and semantics in urban CFD simulations is the main focus of a study by García-Sánchez et al. (2021). They simulate airflow across a part of the TU Delft campus using city models at two levels of geometric detail (LoD 1.3 and 2.2), with and without semantic surfaces for vegetation and water. Their results confirm that both geometry and surface classification affect local wind patterns at pedestrian level, especially near tall buildings. While the study focuses on wind comfort, the authors note that these effects could also be relevant for pollutant or pathogen dispersion—highlighting a broader role for vegetation modelling.

These findings build on earlier work by Ricci et al. (2017), who showed how geometric simplification can distort flow behaviour in dense urban environments. Interestingly, this links back to Buccolieri et al. (2018), where the need for detailed Leaf Area Density (LAD) inputs in CFD is also stressed. All three papers converge on the same idea: vegetation structure matters, but is often oversimplified.

The importance of distinguishing between different parts of a tree—such as treating trunks as solid obstacles and crowns as porous media—has also been demonstrated in CFD-based pollutant dispersion studies. Hong et al. (2017), for example, explicitly model tree trunks as impermeable objects, showing that such structural separation improves accuracy in near-ground wind flow predictions.

Taken together, these studies underscore the importance of semantic and geometric detail in vegetation modelling. They also highlight the need for a robust, scalable pipeline that translates lidar-based tree data into CFD-suitable representations.

### 2.2 Filtering Vegetation from Aerial Lidar

#### 2.2.1 Generation of Multispectral Features

Chen et al. (2023) describes species classification using deep learning on high resolution UAV RGB and multispectral satellite images. The robustness of the model is questioned still, but the features used to train the model look interesting enough to apply in my research. Table

three provides these metrics and the metrics. The best performing features as indicated in the section 3.2 of Chen et al. (2023) are **Norm\_G**, **ARVI** and **MTVI2**. I provide their definition in table 1.

Metric	Equation	Reference
NDVI	$\frac{\rho_{\text{nir}} - \rho_{\text{red}}}{\rho_{\text{nir}} + \rho_{\text{red}}}$	Haboudane et al. (2004)
Norm G	$\frac{\rho_{\text{green}}}{\rho_{\text{green}} + \rho_{\text{red}} + \rho_{\text{blue}}}$	Fraser et al. (2017)
ARVI	$\frac{\rho_{\text{nir}} - [\rho_{\text{red}} - \gamma(\rho_{\text{blue}} - \rho_{\text{red}})]}{\rho_{\text{nir}} + [\rho_{\text{red}} - \gamma(\rho_{\text{blue}} - \rho_{\text{red}})]} \quad (\gamma = 0.5)$	Kaufman and Tanré (1996)
MTVI2	$\frac{1.5(1.2(\rho_{\text{nir}} - \rho_{\text{green}}) - 2.5(\rho_{\text{red}} - \rho_{\text{green}}))}{\sqrt{(2\rho_{\text{nir}} + 1)^2 - (6\rho_{\text{nir}} - 5\rho_{\text{red}})} - 0.5}$	Haboudane et al. (2004)

Table 1: Vegetation indices that can be used to identify, separate and classify vegetation.

The Normalized Difference Vegetation Index (NDVI) is a widely used spectral indicator of vegetation health and density based on the difference in red and near-infrared (NIR) reflectance. Healthy green vegetation reflects relatively more NIR light and absorbs more red light. Higher NDVI values therefore indicate active, healthy plant canopies, while lower values suggest sparse or stressed vegetation.

In remote sensing or UAV-based imaging, the normalised green value (norm G) refers to the ratio of the green light reflectance and the total reflected light across the three-band (RGB) image. By normalizing the green band in this way, *normG* adjusts the green band intensity relative to the overall RGB intensity. This approach helps to mitigate the effects of shade, illumination changes and varying exposure, which proves useful in studies examining burn severity, vegetation health or land cover changes Fraser et al. (2017).

Kaufman and Tanré (1996) introduces the Atmospherically Resistant Vegetation Index (ARVI), a vegetation index designed to correct for aerosol effects in satellite images. While this correction is important for satellite imagery, it is less relevant for aerial imagery. Since I will not be using satellite imagery due to its lower resolution, I will disregard this otherwise effective parameter.

The Modified Triangular Vegetation Index (MTVI2) can be calculated from the reflectance in the near-infrared, red and green band. The MTVI2 value is used for detecting chlorophyll content at the canopy scale while being relatively insensitive to leaf area index Haboudane et al. (2004). For more details check Table 1.

## 2.3 Tree Instance Segmentation

### 2.3.1 Pretrained Model Using Aerial Imagery

DeepForest is a deep learning framework for detecting individual trees from RGB aerial imagery using a pretrained RetinaNet model Weinstein et al. (2020). It outputs 2D bounding boxes of tree crowns and offers pretrained models for immediate use, as well as fine-tuning options through annotation and custom training. While DeepForest has proven effective for forest canopy detection in RGB data, it is not directly applicable to my research, which focuses entirely on airborne lidar point clouds. Moreover, vertical information from the tree point

clouds is essential for later steps such as species classification. Additionally, DeepForest's 2D bounding boxes are less informative than the 2D projections of convex hulls derived from 3D tree instances, which I plan to use as a more geometry-aware representation of the crown footprint. For these reasons, DeepForest will not be incorporated into this project, although it remains a valuable reference for deep learning-based tree detection workflows.

### 2.3.2 Cuboid-Based Tree Delineation

In their 2018 paper, Wang et al. (2018) proposed a tree segmentation method designed to efficiently handle large-scale point clouds while remaining adaptable across different lidar sources, including airborne data. A key idea in their approach is the use of cuboid-shaped cells rather than voxels. The difference lies in changing the shape of a cell from a cube to a cuboid. By stretching the cells vertically (larger z) and refining them horizontally (smaller x and y), the algorithm achieves finer separation between adjacent tree crowns without a proportional increase in memory or processing load.

As illustrated in Figure 1, once tree points are isolated from the raw cloud, these cuboids are clustered and used to identify seed regions for potential tree tops ( $S_1, S_2, S_3, S_4$ ). Tree labels are then propagated throughout the cuboid structure in either top-down or bottom-up passes, allowing the algorithm to handle occlusion and overlapping crowns. Rather than assigning cells based purely on distance, the method introduces a connectivity coefficient that accounts for both spatial proximity and the type of connection between cells (face, edge, etc.), resulting in more nuanced and accurate tree segmentation. The approach was validated on a dataset of 11 densely packed trees, where manually segmented ground truth was used as a reference. The method achieved a Cohen's kappa of 94%, outperforming Wu et al. (2013), which had 89%. This improvement highlights its ability to handle overlapping crowns more effectively while maintaining computational efficiency Wang et al. (2018).

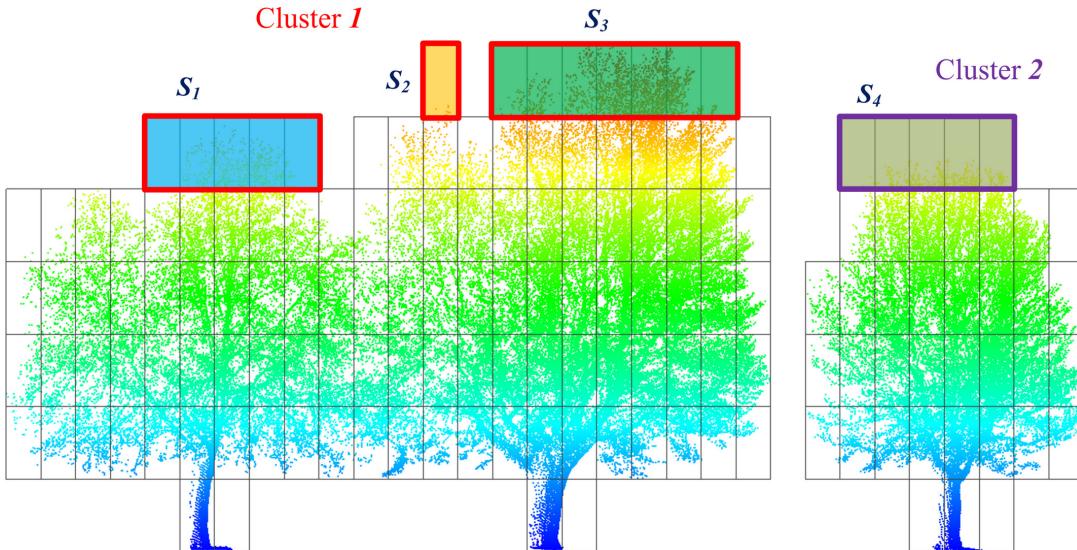


Figure 1: Tree instance segmentation process from Wang et al. (2018), showing the use of cuboid-based clustering for separating individual trees in a lidar point cloud. Seed regions ( $S_1, S_2, S_3, S_4$ ) are identified at the canopy level and used to propagate labels downward, forming distinct tree instances (Cluster 1 and Cluster 2).

### 2.3.3 Tree Delineation from Lidar and Hyper-Spectral Imagery

While exploring methods for individual tree segmentation, I reviewed the work of Lee et al. (2017), which applies a graph-cut-based approach to delineate trees using airborne lidar and hyper-spectral imagery. The algorithm constructs an energy function that balances spectral similarity and spatial coherence, optimizing the segmentation through a min-cut/max-flow solution. This method is particularly effective for isolating tall trees in forested areas, as it leverages both height and spectral contrast to segment tree canopy instances. However, the authors note that its performance degrades significantly for trees smaller than 20 meters, which makes it unsuitable for my dataset, as it includes a variety of smaller urban trees. While the paper provides valuable insights into energy function optimization for tree segmentation, it is not directly applicable to this work.

## 2.4 Inferring Species from Airborne Lidar

### 2.4.1 Getting training data

The MSc thesis by Geert Jan de Groot (2020) describes how public tree species data can be collected from the municipality of Rotterdam. Since Dutch municipalities maintain records of their trees, data is available on their locations, species, height, and other attributes. While information about size and pruning requirements may become outdated, only the location and species name are necessary for linking tree instances to their corresponding species. Both of these attributes are invariant and, therefore, less susceptible to errors due to an older dataset.

### 2.4.2 Machine Learning Methods for Tree Delineation

Deep learning has been explored for tree species classification from laser data, with various methods attempting to improve accuracy and robustness. One approach, the Point Cloud Tree Species Classification Network (PCTSCN), was developed to address challenges in this domain. Proposed by Chen et al. (2021), this method focuses on distinguishing between two tree species: white birch (broadleaf deciduous) and larch (coniferous). While limited in scope, it represents a valuable step toward broader tree species classification.

To delineate four tree types—coniferous, deciduous, dead tree with crown, and snag—deep neural networks (DNNs) have been applied with promising results. Hell et al. (2022) evaluated two such networks, PointCNN and 3DmFV-Net, achieving overall accuracies (OA) of 87.0% and 73.2%, respectively, using a point cloud density of 80 pts/m<sup>2</sup>. Prior to classification, a total of 2,721 individual trees were delineated using a normalized cut segmentation method. These trees were manually labelled into four tree classes, providing the ground truth for training and evaluation. The inclusion of features derived from multispectral orthophotos led to accuracy improvements of up to 16.3%.

Beyond deep learning models, spatial metrics can further refine tree species classification. Slavík et al. (2023) analyzed 1,045 trees using generalized linear models (GLM) and random forest (RF) classifiers to distinguish coniferous from deciduous trees. Their study introduced the Clark-Evans spatial aggregation index (CE) (Equation 1) to assess point cloud clustering. Incorporating the CE index improved classification accuracy, raising GLM performance from 92.6% to 94.8% and RF performance from 93.8% to 95.1%.

$$CE = \frac{\frac{1}{n} \sum_{i=1}^n r}{\frac{1}{2} \times \sqrt{\frac{A}{n}}} \quad (1)$$

### 2.4.3 Tree-level Features for Model Training

Effective species classification models rely heavily on the quality and relevance of extracted features from single tree point clouds. These features may include geometric properties (e.g., height, crown width), vertical structure profiles, and foliage-related metrics such as LAI or LAD. Chi et al. (2025) provides a well-documented set of extractable features from tree point clouds, which are summarized in Appendix Table 4.

In addition to static geometric descriptors, more dynamic signatures can be derived from the distribution of leaf area along the vertical axis of the tree. Kamoske et al. (2019), also referenced in Parker (2020), proposes a method to generate LAI-height curves by summing LAD values per height bin. This results in a vertical “signature” of leaf density that may offer a discriminative feature for species classification. I find this approach promising, as it could help distinguish species based on typical crown architectures and foliage distribution.

### 2.4.4 Leaf Area Index and Leaf Area Density

The Leaf Area Index (LAI) is a widely used metric for analysing urban canopy structure, providing insights into the total amount of leaf surface relative to the ground area beneath a tree. It is typically defined as the total one-sided leaf area per unit of ground surface area ( $m^2 / m^2$ ), though multiple definitions exist, as discussed in detail by Fang et al. (2019). The MSc thesis by Hermann (2024) also provides a thorough literature review on LAI, while Parker (2020) discusses various methods for its calculation, highlighting their respective benefits and limitations.

While LAI provides a general measure of leaf abundance, it does not account for the vertical distribution of leaves within a tree. Leaf Area Density (LAD), on the other hand, is a three-dimensional extension of LAI, representing the total leaf area per unit volume ( $m^2 / m^3$ ). This distinction is particularly relevant when considering tree porosity in CFD simulations, where understanding how air flows through different layers of a canopy is crucial Buccolieri et al. (2018). Since LAD captures the spatial distribution of leaves, it provides a more appropriate metric for modelling airflow within tree canopies.

One study that stood out to me is Kamoske et al. (2019), which was recommended to me by phd candidate Ivan Paden and is also referenced in Parker (2020). The algorithm calculates the LAD per voxel and considers the LAI as the vertical sum of LAD as visualized in Figure 2. This work moreover introduces an open-source R package that can extract LAD from aerial point cloud data, making it a useful tool for my research.

## 2.5 Urban Tree Reconstruction

### 2.5.1 Level of Details for Vegetation

Levels of Detail (LoDs) are well established for buildings in 3D city models, particularly through the CityGML standard, which formalises LoD0 to LoD4 for both geometric and semantic representations Gröger and Plümer (2012). In contrast, equivalent frameworks for vegetation have remained underdeveloped. In her MSc thesis, Lessie M Ortega-Córdova (2018) addresses this gap by proposing a structured set of 14 refined LoDs for Single Vegetation Objects (SVOs), along with four additional levels for root representation, (Figure 3). These specifications account for geometric, semantic, and application-driven aspects of vegetation modelling and are tailored to urban environments. The definitions introduced in this work will serve as a reference framework for describing tree geometries throughout this thesis.

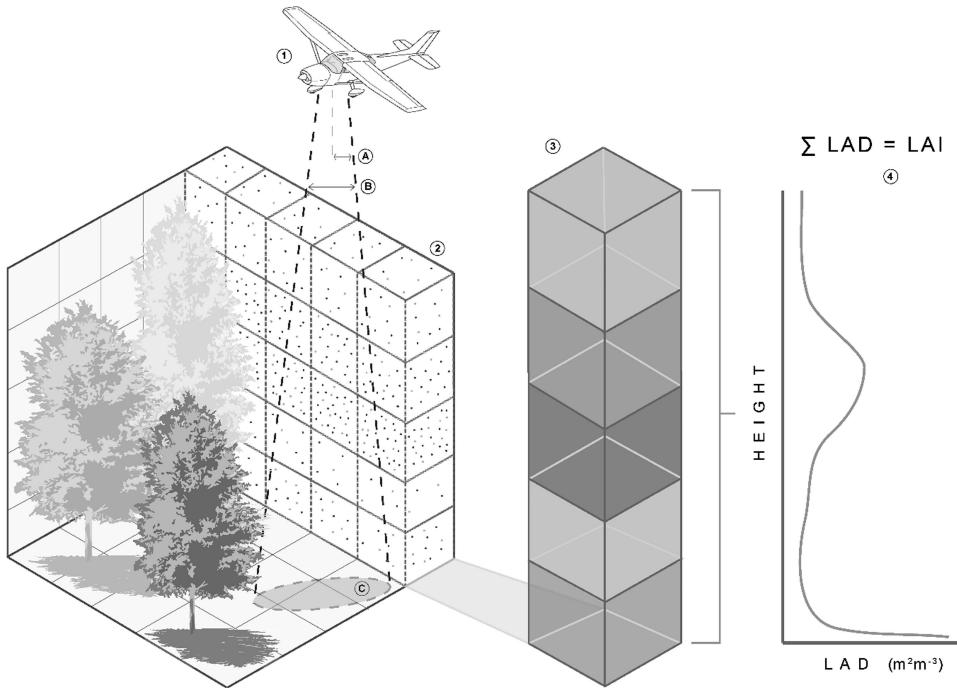


Figure 2: Voxelisation method to retrieve LAD and LAI from aerial lidar as described in Kamoske et al. (2019).

### 2.5.2 Alpha Shapes

Alpha shapes offer a computationally efficient way to generate watertight surface approximations of point-based geometries. This technique has been used by Geert Jan de Groot (2020) to delineate individual tree crowns from airborne lidar data. The approach produces a smoothed representation of the tree's outer structure and is well suited for generating LoD3 models, which include a generalised representation of crown volume. Given the findings of García-Sánchez et al. (2021), who show that higher geometric detail has a measurable impact on CFD wind flow simulations, alpha shapes provide a pragmatic step beyond simplified models (e.g. LoD1) without introducing significant computational overhead. This makes them a promising candidate for geometry reconstruction in this research.

### 2.5.3 Voxelisation

Voxelisation is a common strategy for representing 3D vegetation structure, particularly in contexts where physical accuracy outweighs visual realism. By subdividing space into regular volumetric units, trees can be described in a resolution-aware manner that facilitates the direct calculation of metrics such as leaf area density (LAD).

Keerthinathan et al. (2025) emphasise that voxel-based representations of trees provide sufficient detail for CFD modelling of heat transfer. In such applications, voxel grids can serve as simplified proxies for foliage porosity and aerodynamic resistance—especially in simulations where detailed surface meshing is either infeasible or unnecessary. Although voxelised trees lack smooth outer geometry, they align well with grid-based CFD solvers and can be generated directly from point cloud data with minimal pre-processing. As stated by Keerthinathan et al. (2025), “LiDAR’s high spatial resolution often results in greater detail of the vegetation structure, but for CFD purposes, simplified voxel grids have shown to be effective and computationally practical for meshing and heat transfer analysis.”

	LODx.A	LODx.B	LODx.C	LODx.D
LOD0.x	• or LOD0.A	LOD0.B	LOD0.C	
LOD1.x	LOD1.A	LOD1.B	LOD1.C	LOD1.D
LOD2.x	LOD2.A	LOD2.B	LOD2.C	
LOD3.x	LOD3.A	LOD3.B	LOD3.C	LOD3.D
ROOT				
Optional LOD	ROOT.sprd	ROOT.vol	ROOT.vtype	ROOT.realistic

LOD1.D, LOD2.A and LOD2.B and some roots are library models (ESRI)

Figure 3: Basic geometric Levels of Detail for Single Vegetation Objects (SVOs) as proposed by Lessie M Ortega-Córdova (2018).

#### 2.5.4 Trunk Reconstruction

So far, only the tree foliage has been discussed. For urban CFD modelling, however, the trunk is also a critical structural element. Hong et al. (2017) used a Reynolds-averaged Navier–Stokes (RANS) model to simulate pollutant dispersion in urban street canyons, explicitly differentiating between porous tree crowns and solid trunks. While the crowns were modelled as porous volumes based on LAD values, the trunks were treated as impenetrable solid objects. This distinction was shown to significantly affect airflow patterns, especially near ground level, where pedestrian wind comfort and pollutant concentrations are most sensitive to obstructions.

In terms of geometry reconstruction, Fu et al. (2020) propose a method to estimate tree diameter at breast height (DBH) from airborne lidar. Their model uses crown-level features—such as crown area, tree height, and LAD—alongside species information. A non-linear mixed-effects model is applied to account for both fixed effects (tree morphology) and random effects (e.g., site-specific variation due to terrain or microclimate). This approach is particularly relevant in urban settings, where only aerial point clouds are available and field-based DBH measurements are often missing.

Still, direct DBH estimation from airborne lidar remains a challenge. As highlighted by Mao et al. (2023), occlusion from the tree canopy limits visibility of the lower trunk, reducing estimation accuracy. Despite this, indirect methods using crown and species data—such as that of Fu et al. (2020)—offer a practical alternative for reconstructing trunk geometry when no ground truth is available.

### 3 Research Goal and Questions

The goal of this MSc thesis is to develop a scalable, automated method to reconstruct CFD-ready models of individual urban trees using open-source airborne lidar datasets enriched with RGB and NIR reflectance. The output of the pipeline will be a set of geometrically realistic tree models, each assigned a species-based porosity value that can be integrated into CFD simulations.

This goal can be divided into two research objectives:

1. **Segment and reconstruct 3D tree geometries** from dense point cloud data in a way that preserves crown structure while remaining computationally efficient for city-scale datasets, enabling integration into larger urban digital twin frameworks.
2. **Classify tree species** based on structural and spectral features extracted from the point cloud, using machine learning models trained on public municipal data.

These objectives address gaps in current literature. Most lidar-based species classification studies focus on binary problems (typically conifer vs. deciduous) and small-scale forest plots. Urban environments—with diverse species, irregular planting, and complex crowns—are under-represented. Moreover, few approaches link tree modelling to CFD applications. Existing digital twin pipelines prioritise buildings, with limited attention to vegetation. This thesis proposes a scalable pipeline that integrates segmentation, classification, and geometry simplification for CFD-ready trees. These goals require the generation of tree geometries that preserve both foliage volume and trunk structure, with simplifications appropriate for downstream CFD meshing and simulation.

The following research questions guide the work:

- *How can individual trees be segmented from an unstructured airborne point cloud in an urban context?*
- *What features can be extracted from segmented tree instances to enable accurate species classification?*
- *Which machine learning methods are most effective for species classification using lidar-derived geometric and spectral features?*
- *How can tree instances be converted into simplified, CFD-compatible geometries with assigned properties such as porosity?*

These questions are investigated under the following assumptions:

- My focus is limited to urban areas in the Netherlands, using AHN4 as the lidar source.
- RGB and NIR information is assumed available in the point cloud.
- No image-based methods (e.g., RGB crown detection) are included.
- Public tree inventories provide species labels and are assumed sufficiently accurate.
- The method must scale — i.e., be computationally viable for large datasets.
- CFD simulations are out of scope; only CFD-ready outputs are delivered.
- If time permits, generalisability may be tested outside urban or Dutch areas.

## 4 Methodology

### 4.1 Overview of the Pipeline

Figure 4 presents the proposed pipeline for generating CFD-compatible tree models from airborne lidar. It consists of five stages: data preprocessing, instance segmentation, feature extraction and data matching, species classification, and 3D geometry reconstruction. Each stage addresses one or more research questions from Chapter 3. The process begins by filtering and downsampling AHN4 point clouds to isolate vegetation (RQ1, RQ2), followed by cuboid-based tree segmentation with label propagation and post-processing (RQ1). Segmented trees are matched to public inventory data and enriched with structural features (RQ2). A Random Forest classifier is used initially, with deep learning considered if needed (RQ3). Finally, species-based trunk shapes and porosity-aware crowns are assembled into CFD-ready geometries (RQ4). This chapter outlines the methodology and presents preliminary results where applicable to evaluate feasibility and guide improvements.

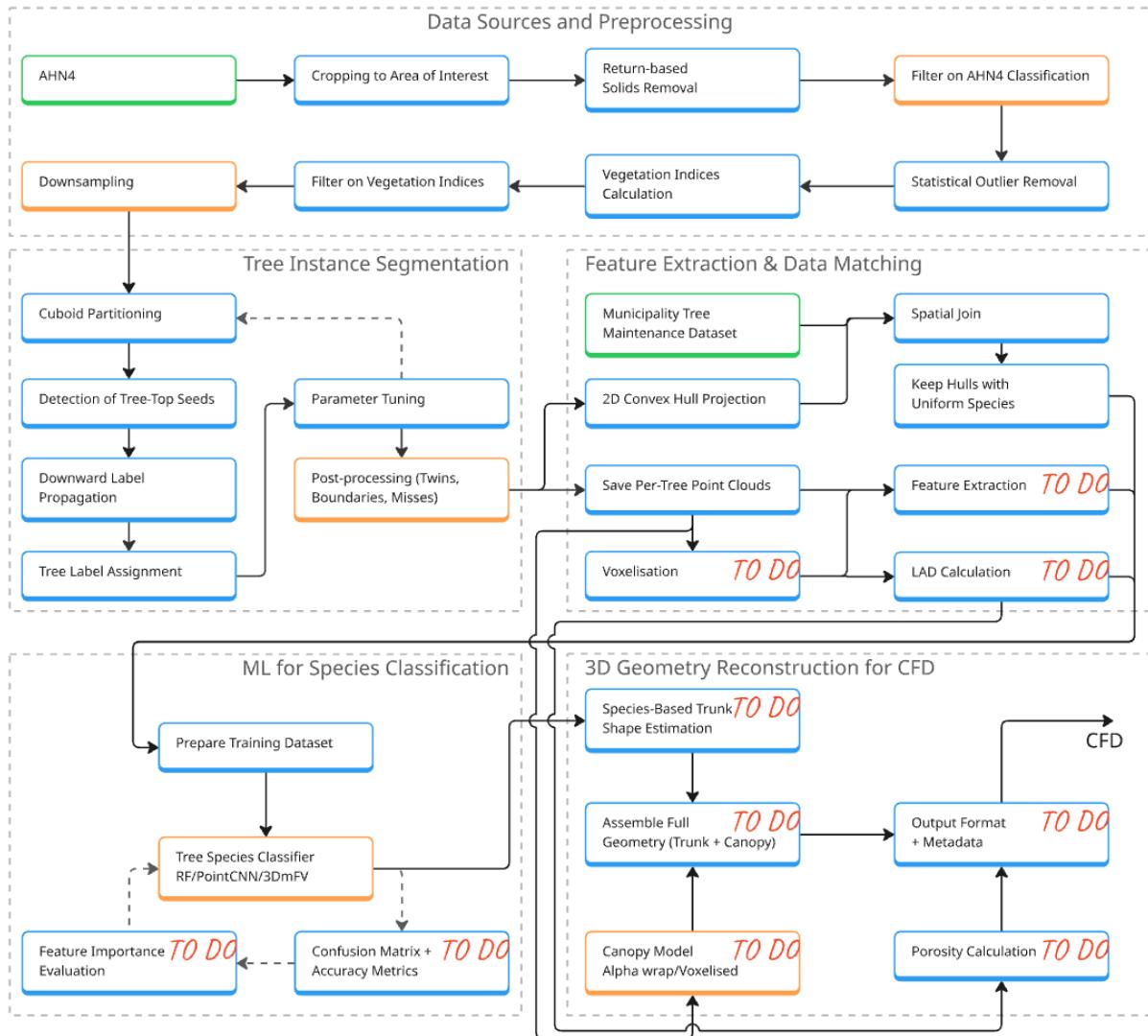


Figure 4: Proposed methodological pipeline for tree segmentation, classification, and CFD-ready geometry generation. Blue boxes indicate implementation steps; orange requires future decision points or refinement; green represents data sources. Boxes that contain 'TO DO' remain to be done while boxes without it have been carried out.

## 4.2 Data Sources and Preprocessing

The first stage of the pipeline prepares raw point cloud data for tree segmentation and species classification. As shown in Figure 4, this stage includes data acquisition, spatial and attribute filtering, outlier removal, vegetation index calculation, and optional downsampling. These steps support RQ1 and RQ2 by progressively filtering the full urban point cloud to isolate vegetation and suppress structural or reflective noise.

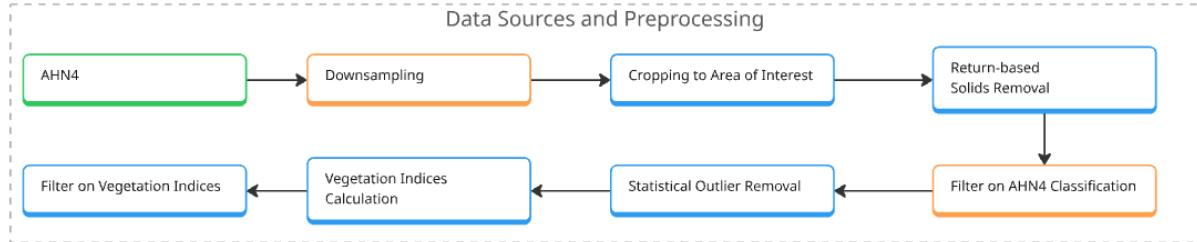


Figure 5: Isolated section of the pipeline, dedicated to the preprocessing of the AHN point cloud.

### AHN4 Dataset and Area of Interest

The primary input dataset is AHN4, the latest nationwide airborne lidar survey of the Netherlands. Tiles are acquired from the *Geotiles* platform and contain all classified points within a given extent. For this study, a 100 m square region within Wilhelmina Park in Delft is selected as the test site. This location is chosen due to proximity and familiarity, which facilitates intuitive validation during pipeline development. The spatial extent is cropped from the larger AHN tile to reduce memory and runtime overhead. Future scaling to larger areas will require parallelized processing or tiled streaming workflows.

**Note:** Cropping a rectangular area of interest from the AHN4 tile can result in partially included trees near the edges. These trees may propagate through the segmentation, matching, and modelling stages as incomplete instances. Strategies to mitigate this effect, such as extending the crop bounds or buffering tile edges, may be explored in future work.

### Optional Downsampling

The AHN4 dataset typically provides a point density of 10–14 points/m<sup>2</sup>, which is well-suited for detailed tree-level analysis. While this density could become computationally intensive at large scales, downsampling is unnecessary at this stage due to the small size of the current test area (100 m × 100 m). Therefore, the original point density is retained throughout early development and validation. Downsampling strategies may be explored later when scaling the pipeline to city-wide datasets.

### Return-Based Solids Removal

The raw point cloud includes ground, buildings, and other non-vegetation structures. To isolate potential vegetation, a filter is applied based on lidar return attributes. Specifically, only points that are *not* the last return of a laser pulse—and not part of single-return pulses—are retained. This logic is based on the assumption that vegetation typically causes partial occlusions, resulting in multiple returns per pulse.

While this filter is effective at removing most solid surfaces such as roofs and the ground, it does not exclusively isolate vegetation. Reflective artifacts and certain structural features (e.g., fences, poles, or façades behind vegetation) may also pass this filter due to multiple surface reflections. As such, return-based filtering is considered a *first-pass* method, and is later refined using vegetation-specific indicators and statistical outlier detection. Figure 6 illustrates the filtering effect at this stage.

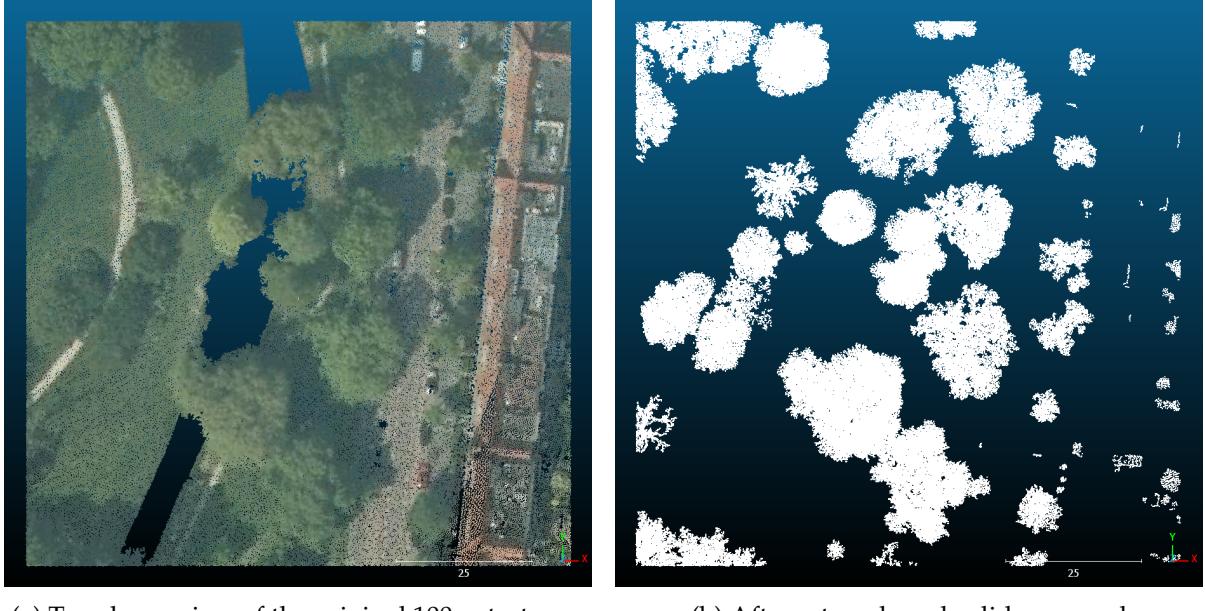


Figure 6: Example of return-based filtering. Points classified as last returns or from single-return pulses are excluded to remove buildings and ground surfaces. The point size is increased for visual clarity.

### Classification-Based Filtering (Optional)

Although AHN4 provides a classification label per point, this information is deliberately excluded from the filtering step. This design decision ensures that the pipeline remains applicable to datasets lacking semantic classification, such as lidar collections in other countries. Avoiding reliance on labelled classes enhances the portability and generalizability of the methodology.

### Statistical Outlier Removal

To reduce noise, a statistical outlier removal algorithm is applied using the Open3D Python library. Each point is evaluated based on its distance to  $nb\_neighbors = 20$  nearest neighbours. Points whose mean distance exceeds two standard deviations ( $std\_ratio = 2.0$ ) from the neighbourhood average are discarded. This step removes sparse outliers without affecting dense crown structures.

### Vegetation Indices Calculation

Three spectral indices are calculated per point to further isolate vegetation:

- **NDVI** (Normalized Difference Vegetation Index), based on red and NIR reflectance.

- **NormG** (Normalized Green), which adjusts for lighting variability in RGB.
- **MTVI2** (Modified Triangular Vegetation Index), indicative of chlorophyll content.

These indices follow definitions in Chen et al. (2023) and Haboudane et al. (2004). While calculated, no threshold-based filtering has yet been applied. NDVI thresholds proved unstable in initial testing, likely due to illumination differences (e.g., shadowed vs. sunlit canopies). Further investigation is needed to determine robust thresholds for each index.

### 4.3 Tree Instance Segmentation

The second stage of the pipeline isolates individual tree crowns from the filtered vegetation point cloud. This step applies a scalable instance segmentation algorithm developed by Wang et al. (2018), which partitions the space into vertically stacked cuboid cells and propagates labels downward from detected treetop seed regions. Each resulting cluster is assigned a unique tree instance ID.

#### Algorithm Overview

The algorithm first partitions the point cloud into cuboid cells using an anisotropic voxel grid. Each cell is binary-labelled based on whether it contains vegetation points. Treetop “seed” regions are detected by clustering neighbouring filled cells in the uppermost layers. Labels are then propagated downward, assigning each point to an instance based on spatial and vertical connectivity. The method is scalable and does not depend on canopy models or peak detection, making it well-suited for diverse and heterogeneous urban environments.

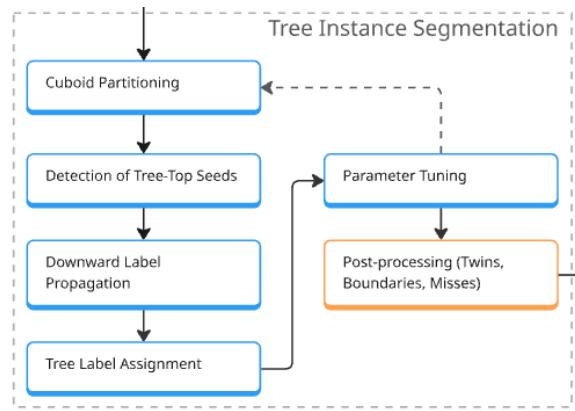


Figure 7: Isolated section of the pipeline, dedicated to the instance segmentation of trees.

#### Parameter Tuning and Evaluation

The algorithm includes several tunable parameters that significantly affect segmentation performance: the horizontal search radius, vertical resolution of the voxel grid, and the minimum number of points required to form a cluster. To determine suitable parameters for the test site, a small grid of parameter combinations was evaluated. Segmentation results were assessed based on two criteria: (1) the number of distinct tree instances detected, and (2) the total number of retained points after segmentation. The first criterion ensures correspondence with the expected number of trees, while the second ensures that the segmentations preserve the completeness of crown geometry.

#### Validation Against Reference Tree Count

The test area, a  $100\text{ m} \times 100\text{ m}$  square at the edge of Wilhelmina Park in Delft, contains approximately 25 mature trees based on visual inspection. Figure 8 shows two segmentation outcomes: one with significantly fewer trees and one with a count that more closely matches the expected reference. While visual inspection helps tune the parameters, some ambiguity

remains—particularly for trees with bifurcating stems near ground level, which can be split into multiple instances.

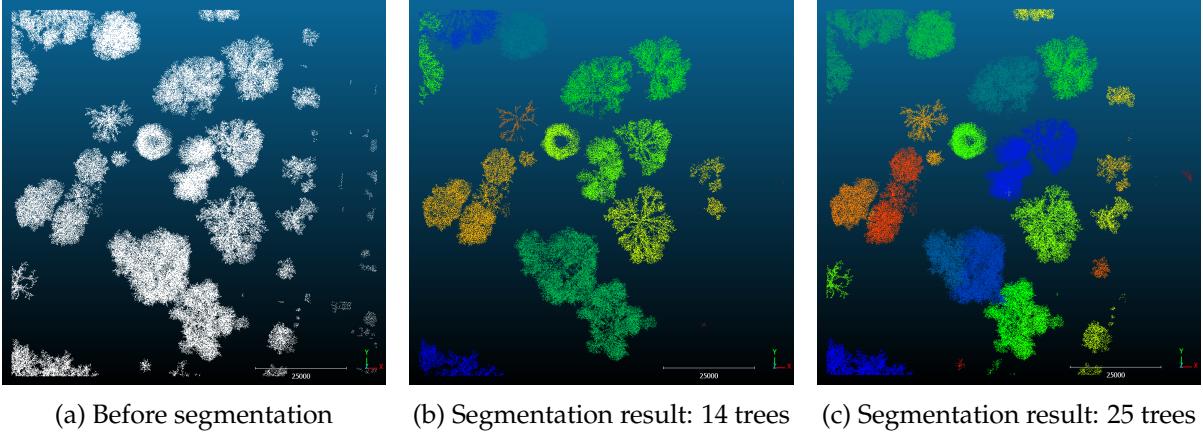


Figure 8: Segmentation tuning comparison. Left: the original test area. Middle: poor segmentation result with under-segmentation. Right: improved segmentation with tree count aligned to ground reference.

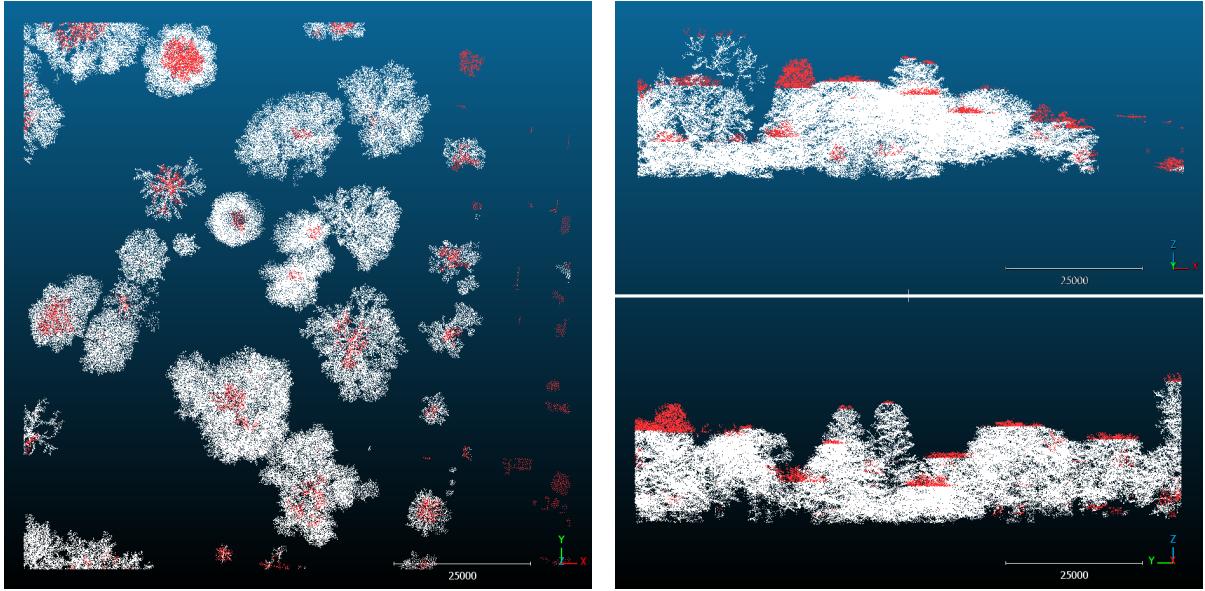
### Known Issues and Post-Processing

A consistent limitation observed during testing is the exclusion of the uppermost portions of tree crowns from the segmentation output (Figure 9). This truncation is substantial enough to affect crown completeness and would introduce significant bias in downstream applications that rely on crown shape, such as biomass estimation or CFD modelling. While the current pipeline can proceed despite this, a complete solution is required before final deployment.

Initial analysis suggests this issue arises from a mismatch between the resolution of the point cloud and the voxel grid used in the segmentation algorithm. In particular, cuboids may be too coarse relative to the point density, or misaligned with the vertical structure of the canopy. Supporting this hypothesis, it was observed that increasing the `min_points` threshold above 3 resulted in no tree detections, indicating a failure to form valid voxel clusters.

This problem will be addressed after the P2 submission. Candidate solutions include adapting the voxel grid size to the point cloud resolution, rescaling the data, or modifying the vertical propagation algorithm to capture sparse canopy regions more effectively.

Following segmentation, additional post-processing may be applied to refine the results. This includes merging erroneously split twin-trees, removing undersized fragments, or filtering incomplete crowns near the boundary of the test area. These steps aim to ensure that only valid, fully segmented trees proceed to the feature extraction and matching stages.



(a) Top view of missing crown points

(b) Side view of crown cutoff

Figure 9: Example of tree crown points excluded from segmentation. These unlabelled points are likely caused by misaligned voxel grid resolution or propagation thresholds.

## 4.4 Feature Extraction and Data Matching

### Stage Overview

This stage of the pipeline enriches each segmented tree with geometric features and links them to known species data from the municipality. As shown in Figure 10, segmented trees are re-associated with their original point cloud attributes and stored as individual instances. Next, 2D convex hulls are computed to spatially match tree clusters with public tree records. Matching logic ensures that only instances with reliable species labels are retained for classification. These steps support RQ2 by producing a clean, labelled dataset for downstream learning tasks.

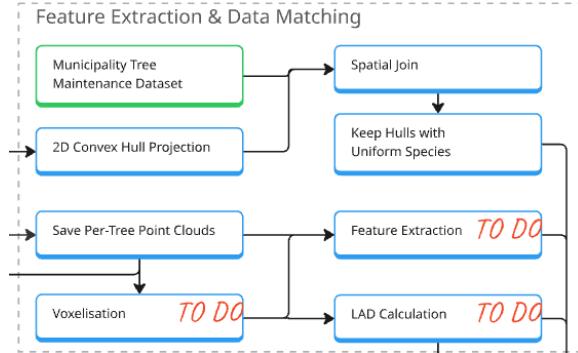


Figure 10: Isolated section of the pipeline, dedicated to feature extraction and species matching.

### Reconnecting Tree IDs to the Point Cloud

The instance segmentation algorithm, based on Wang et al. (2018), operates on simple XYZ input files and produces cluster-labelled XYZ outputs. This means that any additional attributes from the original point cloud—such as intensity, colour, or return number—are lost during processing. However, since the segmentation algorithm does not alter the point coordinates, the tree instance ID (`tid`) assigned to each point can be reattached to the original point cloud by matching coordinates.

This is done by using the segmented XYZ output as a reference to add a new `tid` attribute to the LAS point cloud after preprocessing. This step ensures that the full set of point attributes

remains available for later stages such as feature extraction and visualisation.

### Matching with Public Tree Records

To link segmented trees to known species information, the public tree inventory provided by the municipality of Delft is used. This dataset is available in both CSV and GeoJSON formats and contains metadata such as species label ('boomsortiment') and a unique tree ID. The coordinates are provided in the WGS84 (World Geodetic System 1984) projection and are first reprojected to EPSG:28992 (Amersfoort / RD New) to match the coordinate system used in the lidar processing workflow.

Each segmented tree instance is assigned a 2D convex hull, derived from the projected ( $X, Y$ ) coordinates of its associated points. These hulls serve as approximations of the crown footprint. To associate species labels, a spatial join is performed using the geopandas library. For each hull, the join operation identifies all public tree records whose coordinates fall within its boundary. The result is that every tree instance receives a list of potentially matching municipal tree entries, which is used in a later step to determine training labels.



Figure 11: Spatial join between segmented tree instances and the public tree dataset. Green polygons show 2D convex hulls derived from tree segmentation, each annotated with its corresponding `tid`. Red points mark public tree locations from the municipality dataset. In the zoomed view, species labels are displayed. The aerial basemap is PDOK's WMTS layer — Luchtfoto Actueel Ortho 8cm RGB.

### Species Matching Cases and Filtering Logic

After performing the spatial join between each segmented tree's convex hull and the municipal tree dataset, four distinct cases can be identified based on the number and configuration of matched trees:

- 1. No public trees matched inside the convex hull ( $|t_i \cap H_2| = 0$ ):**  
In this case, the tree instance has no corresponding entry in the municipality dataset.

These instances are excluded from the training set, as no species label can be verified. Such trees may be located on private property or may result from segmentation artefacts or edge effects caused by bounding box cropping.

2. **Exactly one public tree matched** ( $|t_i \cap H_2| = 1$ ):

The tree species from the matched point is assigned to the convex hull as ground truth. These instances are retained as labelled training data for species classification.

3. **Multiple public trees matched** ( $|t_i \cap H_2| > 1$ ):

This indicates that the segmentation algorithm has grouped multiple trees into a single cluster. If all matched species are the same, the hull is retained and labelled accordingly. If the matched species are heterogeneous, the instance is excluded to avoid introducing ambiguity into the training set.

4. **One public tree is matched by multiple convex hulls** ( $|t_i| \cap H_2 = \frac{1}{n}$ ):

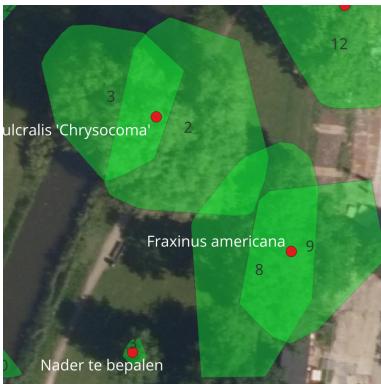
This typically occurs when a single tree is segmented into multiple crowns due to bifurcation near the base (e.g., twin trees). These segments are assigned the same species label, as they originate from the same tree. However, if left unmerged, they may later be modelled as separate trunks. Detecting and merging such twin crowns is therefore an important post-processing step planned for future implementation.

## Special Cases

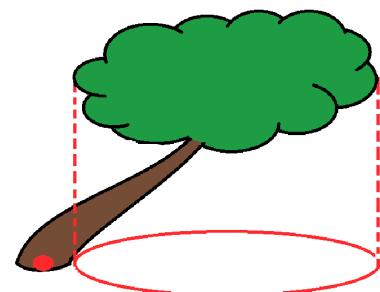
Certain cases fall outside the main matching logic and require special attention. These include geometry mismatches or segmentation artefacts that could affect either training quality or final model geometry.



(a) Tree crown cut by test area boundary



(b) Twin crowns from a single trunk



(c) Tilted tree with displaced crown

Figure 12: Examples of edge cases that influence automatic training data extraction.

Figure 12a shows a case where the crown is included in the segmentation, but the tree base lies outside the cropped point cloud due to the area of interest selection. Since no tree base is detected, matching fails even though the crown is (partly) present.

Figure 12b illustrates cases where a single tree has been segmented into two crowns due to bifurcation near the base. These twins are currently treated as separate instances but should ideally be merged prior to classification and trunk modelling, as they could otherwise result in multiple trunk geometries for the same tree.

Figure 12c depicts a theoretical case (not observed in the test area) in which a tree grows at

an angle due to wind or soil pressure. In such cases, the trunk base may fall outside the 2D projection of the crown. These instances are currently excluded from training due to spatial mismatch, but further investigation is needed if such growth patterns appear species-specific.

## Filtering for Training Data

After evaluating the hull-to-tree matches, only those instances with a clear and unique species label are retained for classifier training. This includes convex hulls with one public tree inside, or multiple trees of the same species. All ambiguous cases—such as overlapping hulls, mixed-species matches, or spatial outliers—are excluded to ensure clean training data. The resulting filtered set is passed to the next stage of the pipeline: feature extraction and species classification.

## Feature Extraction

Once tree instances have been segmented and matched with species labels, a variety of structural and geometric features are extracted to describe each tree in numerical terms. These features serve as input for the species classification stage (RQ3), and some are also relevant for downstream tasks such as geometry reconstruction and porosity estimation for CFD (RQ4). Feature extraction enables learning algorithms to detect meaningful patterns in shape, density, or spatial structure that may relate to species.

At this stage, the aim is to extract as many potentially informative features as possible. This allows for flexibility in later stages, such as performing feature importance evaluation and pruning redundant or noisy variables to improve classifier accuracy. Features are derived from both the spatial arrangement of points and their associated attributes. These include return number, intensity, and vegetation indices (e.g., NDVI, NormG, MTVI2), which were previously computed during preprocessing.

Some features are calculated over the entire tree, such as total point count, crown width, and height percentiles. Others are derived from spatial subdivisions of the tree, such as horizontal layers (percentile slabs) or vertical bins. For example, one feature may represent the number of points below the 25th percentile height, while another could be the average NDVI in that same layer. The full list of metrics includes both geometric descriptors and vegetation indices, as summarised in Table 1 and Table 4, the latter of which is based on the feature set proposed by Chi et al. (2025).

Since the RGB values used to derive these indices originate from aerial orthophotos rather than directly from the lidar sensor, their reliability may be limited to the topmost layers of the crown. This potential bias will be evaluated in future analysis.

Additional spatial metrics, such as the Clark–Evans aggregation index (Equation 1), may also be evaluated as part of a feature importance study. This index has been shown to improve classification accuracy in previous work Slavík et al. (2023), and could offer insight into species-specific crown clustering patterns.

In addition to these horizontal aggregations, radial structure may also be informative. Therefore, features based on concentric slices from the crown centroid are under consideration. For instance, the point intensity or density could be computed at 25%, 50%, and 75% of the average crown radius. This radial drop-off may help distinguish between species with denser central canopies and those with broader, more open growth patterns.

It is also important to extract Leaf Area Density (LAD) from each tree, as this is used both in species classification and later in canopy porosity estimation. LAD will be derived either by adapting the method from Kamoske et al. (2019), or by voxelising the tree point cloud and computing LAD directly within those voxel bins. The latter has the advantage of supporting additional voxel-level features and reduces computational redundancy by consolidating LAD extraction and voxelisation into one step.

Some implementation details—such as voxel resolution or binning strategy—remain to be decided and will be finalised after the P2 submission. The output of this stage will be a set of segmented tree instances, each associated with a species label and a feature vector. These form the basis for training a Random Forest or deep learning classifier and later enable per-voxel porosity calculation in CFD preprocessing.

## 4.5 Species Classification

This stage of the pipeline addresses RQ3 by training a supervised machine learning model to classify each segmented tree instance by species. The classifier takes as input the geometric, structural, and spectral features extracted in the previous stage—such as crown width, height percentiles, NDVI, LAD, and radial or vertical density patterns. The goal is to learn discriminative patterns in these features that are predictive of tree species, enabling automated annotation of unseen tree instances.

### Stage Overview

Figure 13 shows the relevant section of the pipeline, including optional classifier types and validation steps.

Random Forest is selected as the initial classifier due to its strong performance in prior studies—achieving up to 95.1% accuracy when combined with spatial features such as the Clark–Evans index Slavík et al. (2023)—as well as its interpretability and compatibility with structured feature vectors.

If Random Forest performance proves insufficient—e.g., due to point cloud sparsity, skewed class distributions, or non-linear feature relationships—deep learning alternatives such as PointCNN or 3DmFV will be explored. These methods can learn directly from point clouds, though this comes with the trade-off of reduced interpretability.

The output of this stage is a predicted species label per tree instance, which will be used to guide downstream tasks such as trunk reconstruction and canopy porosity estimation for CFD.

At the time of writing, only limited testing has been performed due to the small size of the initial test area (25 trees). This is insufficient for training a reliable classifier, but the general workflow and evaluation pipeline have been implemented and are ready for scaling up once more training data is extracted.

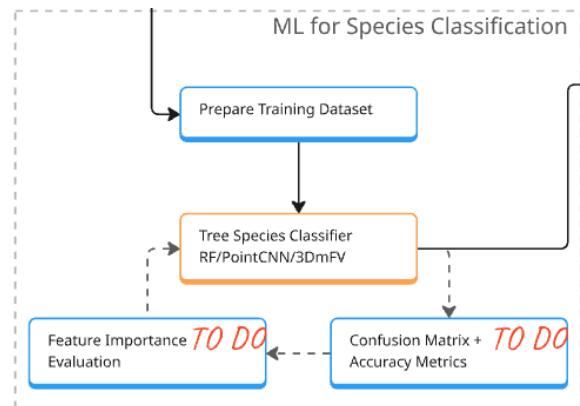


Figure 13: Isolated section of the pipeline, dedicated to the species classification stage. The classifier may be a Random Forest, PointCNN, or 3DmFV model. Evaluation is based on accuracy metrics and feature importance.

Each labelled tree is represented by a structured feature vector. These features will be analysed using feature importance ranking—initially via the `feature_importances_` attribute of the Random Forest classifier in `scikit-learn`. Dimensionality reduction techniques such as Principle Component Analysis (PCA) may also be considered to identify low-discriminative features and reduce overfitting risk.

Depending on the final label distribution, class imbalance may become a concern. If so, balancing techniques (e.g., undersampling or SMOTE) and performance metrics such as the F1-score will be introduced. A basic classification objective will be to distinguish between coniferous and deciduous trees—any improvements beyond this binary split would be considered a success at this stage.

Evaluation will be conducted using a confusion matrix and standard accuracy metrics, including overall accuracy (OA) and per-class recall. If dataset size permits, cross-validation (e.g., k-fold) will be used to ensure generalisability. The Clark–Evans index will be investigated as an additional input feature, given its reported benefit to Random Forest performance in Slavík et al. (2023).

## 4.6 3D Geometry Reconstruction for CFD

### Stage Overview

This final stage of the pipeline converts segmented and labelled tree instances into 3D geometries suitable for CFD simulation. As shown in Figure 14, the process includes estimating trunk dimensions, generating canopy geometry, and computing porosity either per voxel or per tree. Outputs are exported in a format compatible with CFD solvers and are enriched with metadata to support parameter assignment in simulation environments.

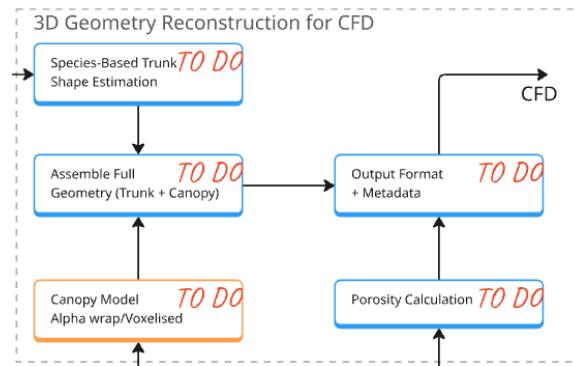


Figure 14: Isolated section of the pipeline, dedicated to geometry reconstruction for CFD.

### Species-Based Trunk Shape Estimation

Trunk reconstruction is carried out using species-specific traits. Trunks are considered solid and impermeable volumes, in line with assumptions in CFD studies such as Hong et al. (2017). Since the lower stem of trees is often occluded in airborne lidar datasets like AHN4, direct observation of the diameter at breast height (DBH) is not possible. Instead, DBH is inferred from observable crown metrics such as tree height, LAD, and crown width, possibly using regression-based models.

In particular, the method proposed by Fu et al. (2020), which estimates DBH from crown attributes using non-linear mixed-effect models, is a promising starting point. While their dataset is denser than AHN4, adapting their model will be explored. Incorporating species as a categorical predictor could further improve estimation accuracy.

## Canopy Geometry Modelling

Tree crown geometry is reconstructed using an *alpha wrapping* algorithm, as implemented in the CGAL library Documentation (2025). This method generates a watertight mesh around the foliage points. Its behaviour is controlled by two parameters: `alpha`, which sets the maximum radius of an empty circumsphere to include a Delaunay facet, and `offset`, which tightens the final isosurface mesh. These parameters control the level of detail and smoothness of the output. Examples of the effect of these parameters are included in Figure 15.

As an alternative, a voxelised representation of the canopy may be used. Keerthinathan et al. (2025) showed that voxelised vegetation can be sufficiently accurate for CFD purposes, especially at larger scales. Given that the LAD is already computed via voxelisation, reusing the voxel grid for crown representation could simplify the pipeline and reduce computational overhead. However, it may compromise visual realism. As shown by García-Sánchez et al. (2021), geometric detail influences flow and turbulence predictions, so this trade-off must be tested for its effect on simulation output.

## Porosity Calculation and Output Format

Porosity is derived from the voxel-level LAD grid created earlier during feature extraction. It remains undecided whether porosity will be assigned per voxel or per tree instance, as this depends on how vegetation is treated in the CFD framework. If trees are modelled individually, fine-grained voxel porosity may be applied. If grouped by species or spatial unit, aggregate porosity might be preferable.

This design decision affects not only the granularity of porosity but also the structure of the exported CFD geometry. If each tree is output individually, associated metadata—such as DBH, LAD, and species—will need to be bundled with each geometry file. Alternatively, a separate metadata linkage file could be used to map instance IDs to simulation parameters, which may help automate parameter assignment in OpenFOAM or similar solvers.

Trunks, being solid, must be exported separately and marked as impermeable in the simulation mesh. The final output format is still under consideration but will prioritise reproducibility, scalability, and compatibility with existing CFD workflows. Coordination with the supervisors, is essential for finalising these details.

While this stage remains under development, its design will be guided by practical feedback from CFD stakeholders and tested for scalability as the pipeline moves toward city-scale deployment. The feasibility of automated parameter assignment and optimal granularity will be evaluated as implementation progresses.

## 5 Time planning

Task	Weeks Needed	Approximate Deadline
Preprocessing		
Tree Instance Segmentation		
Parameter Tuning		
Save tree instances as single tree point cloud files		
Voxelisation		
Tree features extraction		
Training data generator		
ML species classifier		
LAD calculation		
alpha wrapper		
Species based trunk shape estimator		
Canopy geometry having porosity		
Full tree geometry		
Output format for CFD		month before p4?

Table 2: Pipeline Tasks with Time Estimates

### 5.1 Supervision

Weekly meetings with the supervisors will be held on Tuesdays. Dr. Hugo Ledoux and Dr. Clara García-Sánchez will alternate their attendance. If deemed necessary, a meeting with all three of us will be scheduled.

## 6 Tools and datasets used

name	format	openness	URL
geotiles	.laz	CC BY 4.0	<a href="https://geotiles.citg.tudelft.nl">https://geotiles.citg.tudelft.nl</a>
Boombeheer Delft	.csv	Freely to download	<a href="https://www.delft.nl/bomen-delft">https://www.delft.nl/bomen-delft</a>

Table 3: Datasets used.

### Python packages

- Python 3.11
- Numpy 1.26.4
- Pandas 2.2.3
- Geopandas 1.0.1
- Laspy 2.5.4
- Matplotlib 3.9.3
- Scipy 1.15.2

- Scikit-learn 1.6.1

### C++ packages

- lasinfo

## 7 Appendix

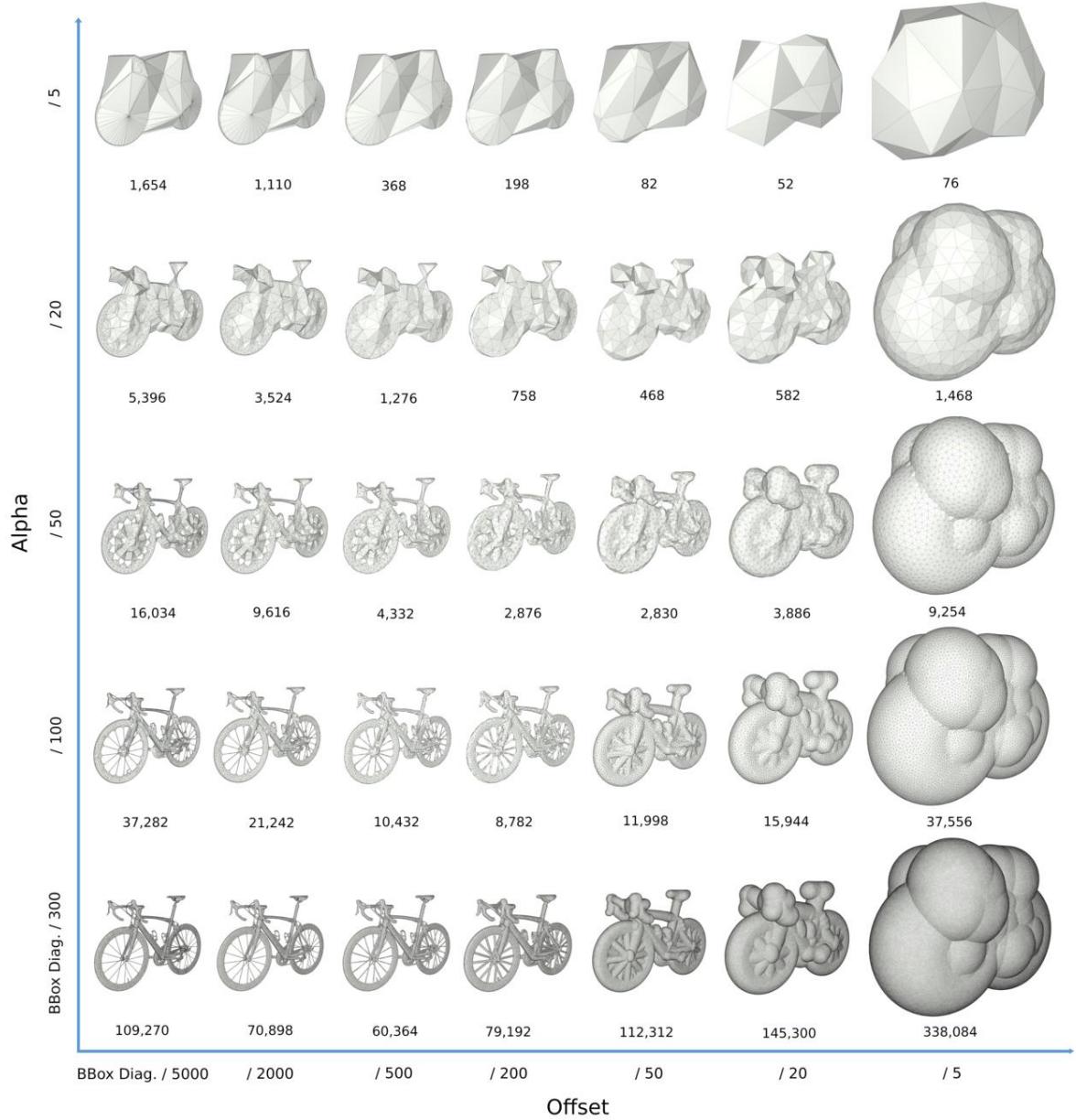


Figure 15: Different alpha and offset values on a bike model taken from the CGAL Documentation (2025). The x-axis represents the offset value equal to  $1/5000$ ,  $1/2000$ ,  $1/500$ ,  $1/200$ ,  $1/50$ ,  $1/20$  and  $1/5$  of the longest diagonal of the input bounding box, from left to right. The y-axis represents the alpha value equal to  $1/300$ ,  $1/100$ ,  $1/50$ ,  $1/20$  and  $1/5$  of the longest diagonal of the input bounding box, from bottom to top. The numbers below each level of detail represents their number of triangles. Depending on the alpha value, an offset too small or too large will produce output mesh with higher complexity. For each alpha, the models with lower complexity can be used as a scale-space representations for collision detection, from near to far distances.

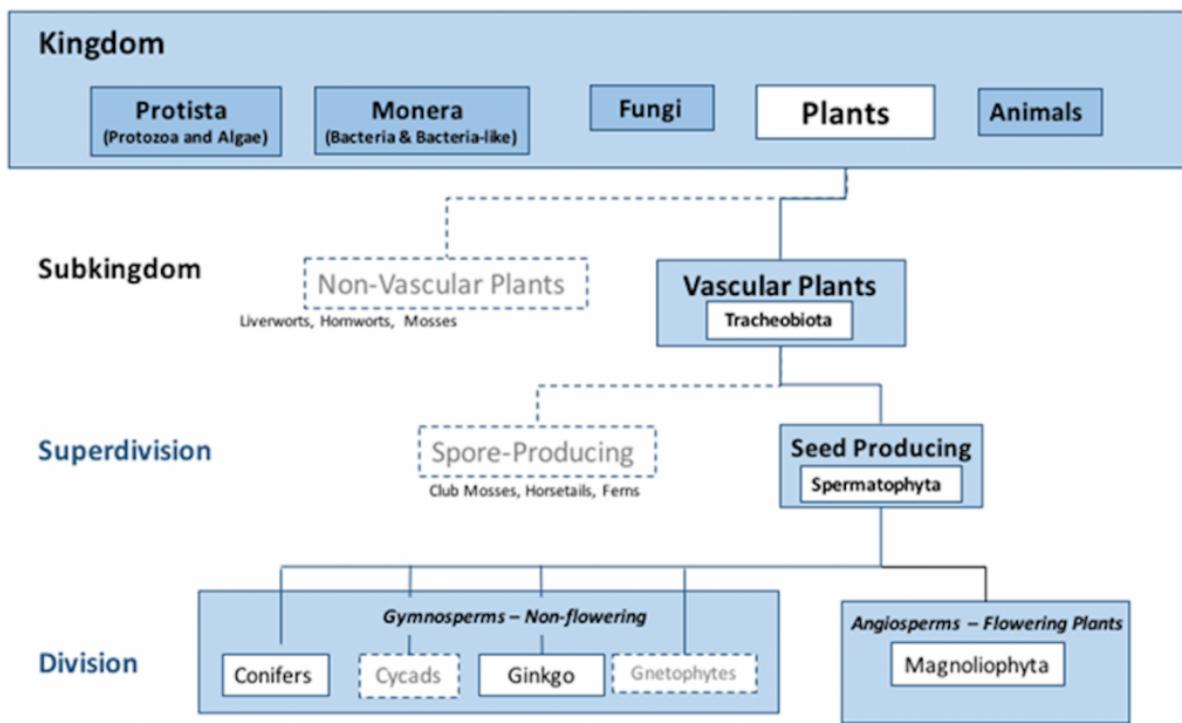


Figure 16: Tree of Life from the kingdom level, taken from Extension (2021)

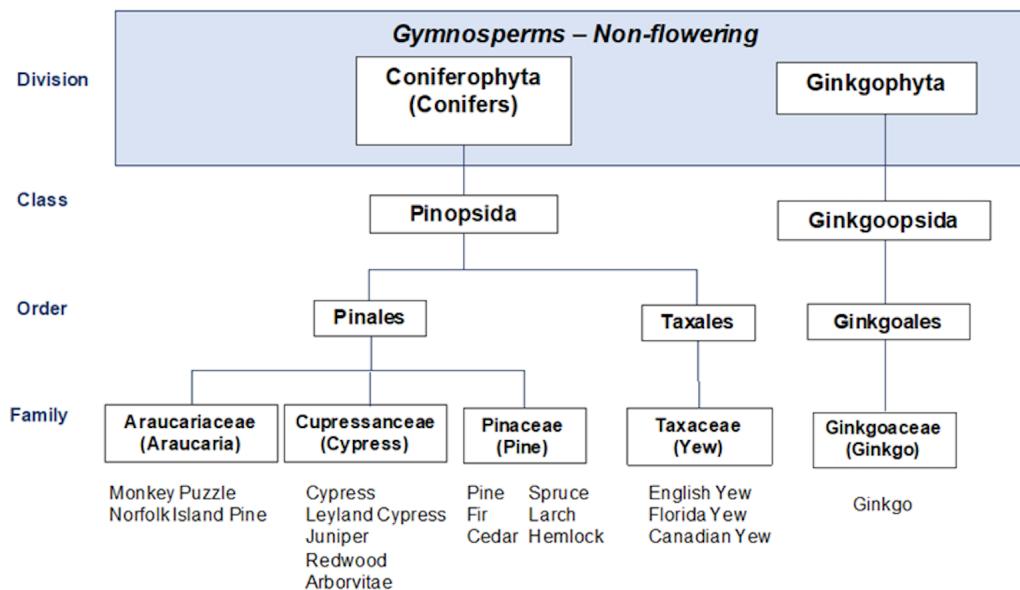


Figure 17: Tree of Life from the division level, taken from Extension (2021)

Feature	Description
<b>Height</b>	
Hmax	Maximum height
Hmed	Median height
Hbase	Crown base height
Hmean	Mean height
Hstd	Standard deviation of height
Hcv	Coefficient of variation of height
Hkur	Kurtosis of height
Hske	Skewness of height
Hp25	25th percentile height
Hp90	90th percentile height
Hfirst_mean	Mean height of first-or-single returns
<b>Intensity</b>	
Imax	Maximum intensity
IaHmed	Mean intensity above median height
IbHmed	Mean intensity below median height
IabHmed	Ratio of IaHmed to IbHmed
Imean	Mean intensity
Istd	Standard deviation of intensity
Icv	Coefficient of variation of intensity
Ikur	Kurtosis of intensity
Iske	Skewness of intensity
Ip25	25th percentile intensity
Ip90	90th percentile intensity
Ifirst_mean	Mean intensity of first-or-single returns
<b>Crown Size and Shape</b>	
CWHmed	Crown width at median height
CWHp75	Crown width at 75th percentile height
CWHp90	Crown width at 90th percentile height
CL_Hmax	Ratio of crown length to maximum height
Hmed_CW	Ratio of crown height to width: median height
Hp75_CW	Ratio of crown height to width: 75th percentile height
Hp90_CW	Ratio of crown height to width: 90th percentile height
CWHP90_Hmean	Ratio of width at 90th percentile height to mean height
CWns_ew	Ratio of N-S width to E-W width
CRR	Canopy relief ratio
<b>Crown Porosity and Density</b>	
Hmean_med	$(\text{Hmean} - \text{Hmed}) / \text{Hmax}$
NHmean	Count of returns in 0.5 m vertical slice at mean height divided by width at that height
NHmed	Count of returns in 0.5 m vertical slice at median height divided by width at that height
NHp90	Count of returns in 0.5 m vertical slice at 90th percentile height divided by width at that height
Nfirst	Percentage of first-or-single returns
Nlast	Percentage of last returns
N	Total number of points

Table 4: Tree classification features and descriptions as described by Chi et al. (2025).

## References

- Buccolieri, R., Santiago, J.-L., Rivas, E., and Sanchez, B. (2018). Review on urban tree modelling in CFD simulations: Aerodynamic, deposition and thermal effects. *Urban Forestry & Urban Greening*, 31:212–220.
- Chen, J., Chen, Y., and Liu, Z. (2021). Classification of Typical Tree Species in Laser Point Cloud Based on Deep Learning. *Remote Sensing*, 13(23):4750. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute.
- Chen, X., Shen, X., and Cao, L. (2023). Tree Species Classification in Subtropical Natural Forests Using High-Resolution UAV RGB and SuperView-1 Multispectral Imageries Based on Deep Learning Network Approaches: A Case Study within the Baima Snow Mountain National Nature Reserve, China. *Remote Sensing*, 15(10):2697.
- Chi, D., Yan, J., Yu, K., Morsdorf, F., and Somers, B. (2025). Planting contexts affect urban tree species classification using airborne hyperspectral and LiDAR imagery. *Landscape and Urban Planning*, 257:105316.
- Documentation, C. (2025). CGAL 6.0.1 - 3D Alpha Wrapping: User Manual.
- Extension, V. C. (2021). Chapter 5: Tree Taxonomy, Identification, and Measurement. Book Title: Tree Steward Manual Publisher: Virginia Cooperative Extension in association with Virginia Tech Publishing.
- Fang, H., Baret, F., Plummer, S., and Schaepman-Strub, G. (2019). An Overview of Global Leaf Area Index (LAI): Methods, Products, Validation, and Applications. *Reviews of Geophysics*, 57(3):739–799. eprint: <https://onlinelibrary.wiley.com/doi/10.1029/2018RG000608>.
- Fraser, R. H., Van der Sluijs, J., and Hall, R. J. (2017). Calibrating Satellite-Based Indices of Burn Severity from UAV-Derived Metrics of a Burned Boreal Forest in NWT, Canada. *Remote Sensing*, 9(3):279. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- Fu, L., Duan, G., Ye, Q., Meng, X., Luo, P., Sharma, R. P., Sun, H., Wang, G., and Liu, Q. (2020). Prediction of Individual Tree Diameter Using a Nonlinear Mixed-Effects Modeling Approach and Airborne LiDAR Data. *Remote Sensing*, 12(7):1066. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.
- García-Sánchez, C., Vitalis, S., Paden, I., and Stoter, J. (2021). THE IMPACT OF LEVEL OF DETAIL IN 3D CITY MODELS FOR CFD-BASED WIND FLOW SIMULATIONS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-4-W4-2021:67–72. Conference Name: ISPRS TC IV<br>16th 3D GeoInfo Conference 2021 – 11&ndash;14 October 2021, New York City, USA Publisher: Copernicus GmbH.
- Geert Jan de Groot (2020). *Automatic construction of 3D tree models in multiple levels of detail from airborne LiDAR data*. PhD thesis, TU Delft.
- Gröger, G. and Plümer, L. (2012). CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33.
- Haboudane, D., Miller, J. R., Pattey, E., Zarco-Tejada, P. J., and Strachan, I. B. (2004). Hyperspectral vegetation indices and novel algorithms for predicting green LAI of crop canopies: Modeling and validation in the context of precision agriculture. *Remote Sensing of Environment*, 90(3):337–352.

- Hell, M., Brandmeier, M., Briechle, S., and Krzystek, P. (2022). Classification of Tree Species and Standing Dead Trees with Lidar Point Clouds Using Two Deep Neural Networks: PointCNN and 3DmFV-Net. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 90(2):103–121.
- Hermann, T. (2024). *Leaf it to AI: Mapping Urban Tree Morphology and Leaf Area Index with Multimodal Deep-Learning*. PhD thesis, Ecole Polytechnique Federale de Lausanne.
- Hong, B., Lin, B., and Qin, H. (2017). Numerical Investigation on the Effect of Avenue Trees on PM2.5 Dispersion in Urban Street Canyons. *Atmosphere*, 8(7):129. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.
- ICPP (2023). Climate Change 2023: Synthesis Report. Technical report, IPCC. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the IPCC.
- Kamoske, A. G., Dahlin, K. M., Stark, S. C., and Serbin, S. P. (2019). Leaf area density from airborne LiDAR: Comparing sensors and resolutions in a temperate broadleaf forest ecosystem. *Forest Ecology and Management*, 433:364–375.
- Kaufman, Y. J. and Tanré, D. (1996). Strategy for direct and indirect methods for correcting the aerosol effect on remote sensing: From AVHRR to EOS-MODIS. *Remote Sensing of Environment*, 55(1):65–79.
- Keerthinathan, P., Winsen, M., Krishnakumar, T., Ariyanayagam, A., Hamilton, G., and Gonzalez, F. (2025). Modelling LiDAR-Based Vegetation Geometry for Computational Fluid Dynamics Heat Transfer Models. *Remote Sensing*, 17(3):552. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- Lee, J., Coomes, D., Schonlieb, C.-B., Cai, X., Lellmann, J., Dalponte, M., Malhi, Y., Butt, N., and Morecroft, M. (2017). A graph cut approach to 3D tree delineation, using integrated airborne LiDAR and hyperspectral imagery. arXiv:1701.06715 [cs].
- Lessie M Ortega-Córdova (2018). *Urban Vegetation Modeling 3D Levels of Detail*. PhD thesis, TU Delft.
- Mao, Z., Lu, Z., Wu, Y., and Deng, L. (2023). DBH Estimation for Individual Tree: Two-Dimensional Images or Three-Dimensional Point Clouds? *Remote Sensing*, 15(16):4116.
- Park, S., Kim, E.-S., Yun, S.-H., and Lee, D.-K. (2024). Efficiency of urban greening systems with maximized latent heat effect in urban heat island and climate change mitigation. *Environmental Challenges*, 15:100956.
- Parker, G. G. (2020). Tamm review: Leaf Area Index (LAI) is both a determinant and a consequence of important processes in vegetation canopies. *Forest Ecology and Management*, 477:118496.
- Pickett, S. T. A., Cadenasso, M. L., Grove, J. M., Boone, C. G., Groffman, P. M., Irwin, E., Kaushal, S. S., Marshall, V., McGrath, B. P., Nilon, C. H., Pouyat, R. V., Szlavecz, K., Troy, A., and Warren, P. (2011). Urban ecological systems: Scientific foundations and a decade of progress. *Journal of Environmental Management*, 92(3):331–362.
- Ricci, A., Kalkman, I., Blocken, B., Burlando, M., Freda, A., and Repetto, M. P. (2017). Local-scale forcing effects on wind flows in an urban environment: Impact of geometrical simplifications. *Journal of Wind Engineering and Industrial Aerodynamics*, 170:238–255.

Rodriguez, A., Lecigne, B., Wood, S., Carmeliet, J., Kubilay, A., and Derome, D. (2024). Optimal representation of tree foliage for local urban climate modeling. *Sustainable Cities and Society*, 115:105857.

Slavík, M., Kuželka, K., Modlinger, R., and Surový, P. (2023). Spatial Analysis of Dense LiDAR Point Clouds for Tree Species Group Classification Using Individual Tree Metrics. *Forests*, 14(8):1581. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

Somanath, S., Naserentin, V., Eleftheriou, O., Sjölie, D., Wästberg, B. S., and Logg, A. (2024). Towards Urban Digital Twins: A Workflow for Procedural Visualization Using Geospatial Data. *Remote Sensing*, 16(11):1939. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

Wang, J., Lindenbergh, R., and Menenti, M. (2018). Scalable individual tree delineation in 3D point clouds. *The Photogrammetric Record*, 33(163):315–340.

Weinstein, B. G., Marconi, S., Aubry-Kientz, M., Vincent, G., Senyondo, H., and White, E. P. (2020). DeepForest: A `<span style="font-variant:small-caps;">Python</span>` package for RGB deep learning tree crown delineation. *Methods in Ecology and Evolution*, 11(12):1743–1751.

WorldBank, T. (2023). Urban population (% of total population) - Netherlands (1960–2023).

Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., and Liu, H. (2013). A Voxel-Based Method for Automated Identification and Morphological Parameters Estimation of Individual Street Trees from Mobile Laser Scanning Data. *Remote Sensing*, 5(2):584–611. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.