

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

Noah Petri Alting

March 21, 2025

student number      4828968

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



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Urban analysis in general . . . . .	2
1.2	vegetation in CFD . . . . .	2
1.3	Get tree species from lidar . . . . .	2
<b>2</b>	<b>Related work</b>	<b>3</b>
2.1	Vegetation in Urban CFD Modelling . . . . .	3
2.2	Filtering Vegetation from Aerial Lidar . . . . .	4
2.2.1	Generation of Multispectral Features . . . . .	4
2.3	Tree Instance Segmentation . . . . .	5
2.3.1	Pretrained Model Using Aerial Imagery . . . . .	5
2.3.2	Cuboid-Based Tree Delineation . . . . .	5
2.3.3	Tree Delineation from Lidar and Hyper-Spectral Imagery . . . . .	5
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	3D Geometry Reconstruction for Urban Trees . . . . .	8
2.5.1	Tree Level of Details . . . . .	8
2.5.2	take species into account . . . . .	8
2.5.3	How trees are currently reconstructed . . . . .	8
<b>3</b>	<b>Research Goal and Questions</b>	<b>9</b>
<b>4</b>	<b>Methodology</b>	<b>10</b>
4.1	Preprocessing . . . . .	10
4.2	Tree Instance Segmentation . . . . .	12
4.3	Tree feature extraction . . . . .	12
4.4	Species classification . . . . .	15
4.5	3D geometry for CFD simulation . . . . .	15
<b>5</b>	<b>Time planning</b>	<b>16</b>
5.1	Supervision . . . . .	16
<b>6</b>	<b>Tools and datasets used</b>	<b>16</b>
<b>7</b>	<b>Appendix</b>	<b>17</b>

# **1 Introduction**

## **1.1 Urban analysis in general**

- [global overview]

Kulicki et al. (2024) terrestrial lidar, deep learning, species classification

Chehreh et al. (2023) overview of latest trends in tree species classification, covering multi-spectral images and lidar.

- [why CFD]

## **1.2 vegetation in CFD**

- [zoomed in on trees in CFD]

## **1.3 Get tree species from lidar**

- [it will help the reconstruction part if not doing alpha wrap but more specific (maybe not needed for CFD)]

- [assign porosities based on species]

## 2 Related work

### 2.1 Vegetation in Urban CFD Modelling

Vegetation is increasingly recognised as a relevant factor in urban digital twin workflows, particularly in the context of airflow and microclimate modelling. In their recent work, ? present a pipeline for procedural 3D visualisation using geospatial data, aimed at bridging GIS-based datasets with simulation-ready urban models.

While the authors focus primarily on the general feasibility of such workflows, they emphasise that current solutions lack a comprehensive end-to-end integration of vegetation in CFD-ready representations. Their case study, based on a coniferous forest, illustrates the aerodynamic relevance of vegetation but lacks urban-specific complexity such as varied species morphology or crown architecture. Nevertheless, the paper references earlier CFD studies by García-Sánchez and others, highlighting the importance of semantic differentiation — including vegetation — for reliable urban airflow simulations. This supports the idea that vegetation is acknowledged, yet still underserved, in current digital twin and urban CFD efforts.

## 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).

### 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.

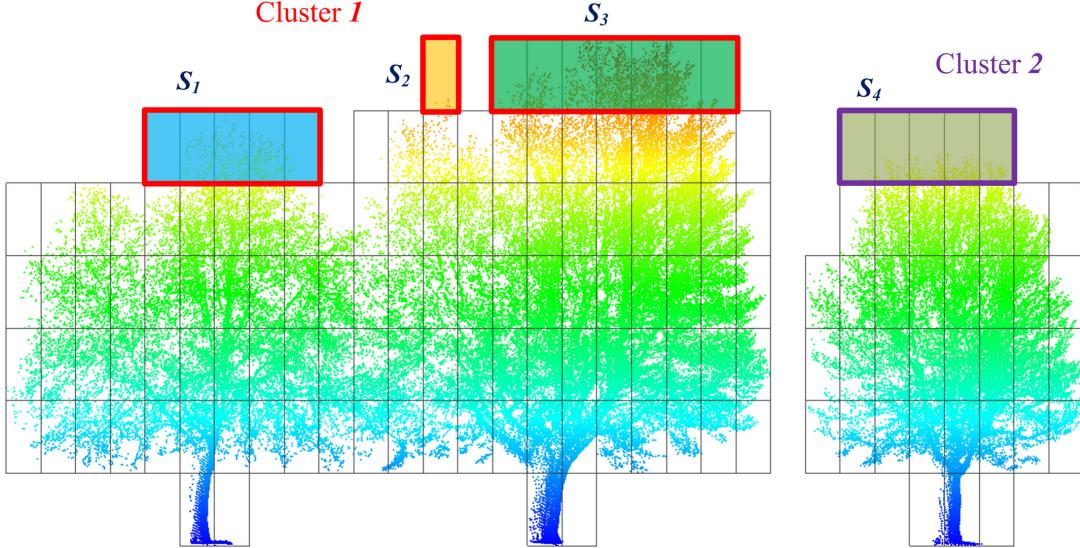


Figure 1: Tree instance segmentation process from Wang et al. (2018), showing the use of cuboid-based clustering for separating individual trees in a lidar point cloud. Seed regions ( $S_1, S_2, S_3, S_4$ ) are identified at the canopy level and used to propagate labels downward, forming distinct tree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

## 2.4 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 3.

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.

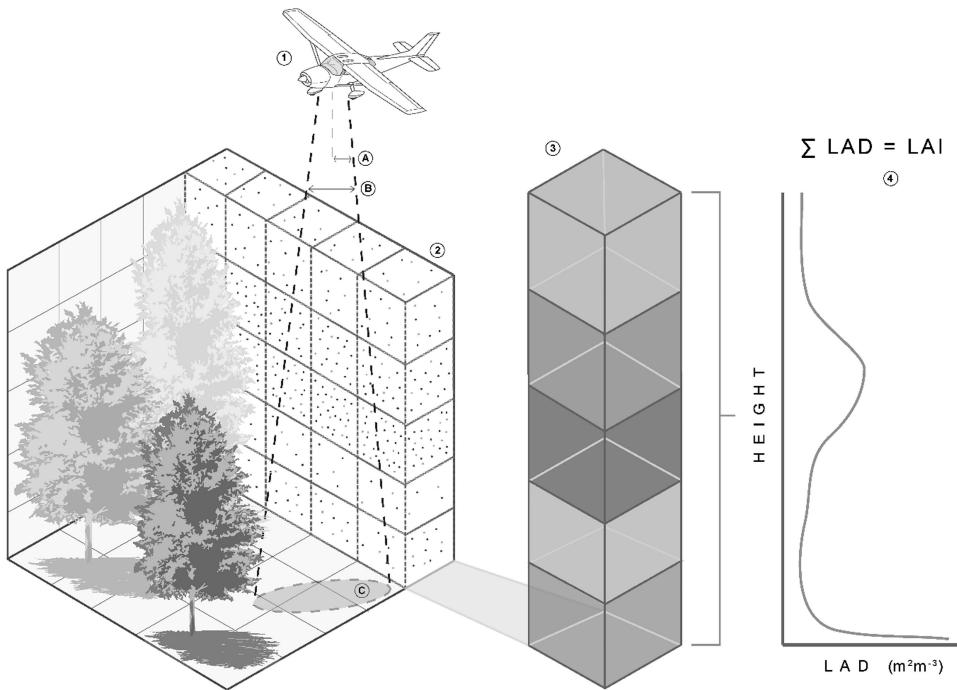


Figure 2: Voxelisation method to retrieve LAD and LAI from aerial lidar as described in Kamoske et al. (2019).

## 2.5 3D Geometry Reconstruction for Urban Trees

### 2.5.1 Tree Level of Details

Lessie M Ortega-Córdova (2018) defines lod's for trees.

### 2.5.2 take species into account

Geert Jan de Groot (2020) describes a method to reconstruct trees from airborne lidar into multiple LoDs. Furthermore, he uses public tree species data from the municipality of Rotterdam.

### 2.5.3 How trees are currently reconstructed

Hermann (2024) describes a method to automate extracting tree datasets using LAI, NDVI and mean crown height.

Geert Jan de Groot (2020) proposes to use the alpha shape of trees, which is also done automatically from

- [alpha wrapping]
- [quality of CFD]
- [assigning porosities to forest model]
- [mention Buccolieri et al. (2018) (urban CFD vs tree review)]

### 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.
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 aerodynamic simplification for CFD-ready trees.

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 aerodynamic 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.

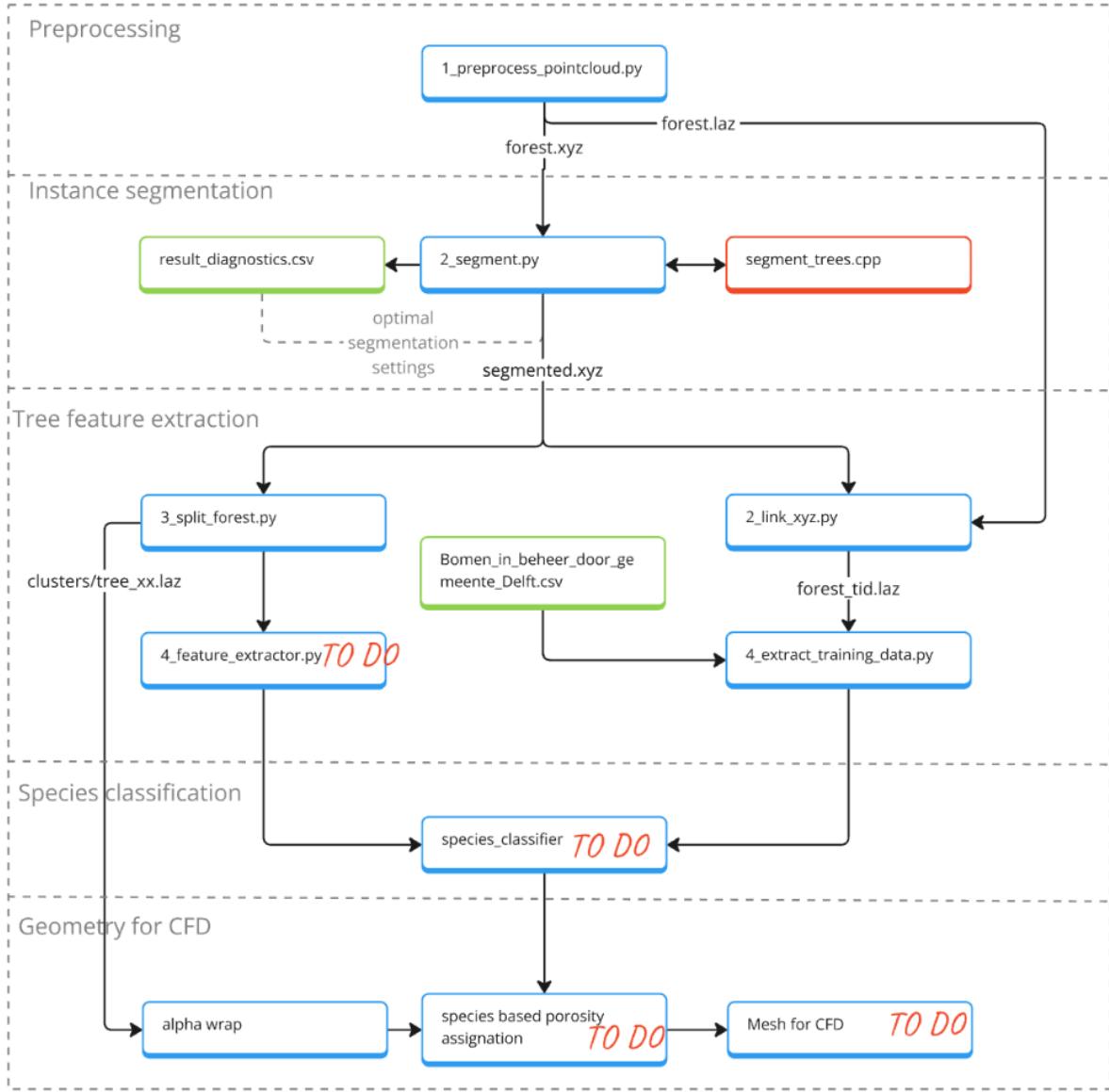


Figure 3: Pipeline of the full methodology I propose. The blue cells are done in python and the orange and green cells represents c++ code and csv files, respectively.

## 4 Methodology

### 4.1 Preprocessing

The point cloud from AHN4 contains all points and needs to be filtered. First, the tile obtained from Geotiles is cropped to a region of interest. This step is important because a Geotiles tile is too large to process. In my proof-of-concept case, I select a 100m square of Wilhelmina Park in Delft, as it is near my home, allowing me to intuitively assess whether the results are realistic. If the research area were larger, I would explore streaming processes to parallelize the computation and later stitch the results back together. For the proof-of-concept case, the area is intentionally kept small to test each step of the pipeline before scaling up.

The resulting tile contains all points within this geographical bounding box. The biggest and simplest filter to apply is filtering on the number of returns. Since we are interested in vegeta-

tion points, we want to filter out buildings and the ground. The AHN point cloud contains the attributes *return number* and *last return*. Each point that is the last return is considered ground or solid building material and is filtered out.

I deliberately avoid filtering the point cloud based on AHN's classification to ensure the proposed pipeline remains applicable to regions beyond the Netherlands. Since non-AHN datasets may lack a classification attribute, omitting this filter enhances the robustness of my approach.

Subsequently, there may still be outliers in the point cloud. Therefore I apply the statistical outlier removal algorithm provided by the *Open 3D python library*. This algorithm uses two parameters: *nb\_neighbors* and *std\_ratio*. For now, I have used the default values of 20 and 2.0, respectively. Additionally, downsampling the point cloud could help reduce computational load. The AHN4 point cloud has an average density of 10-14 points per square meter, which corresponds to a point spacing of approximately 8.4 to 10 cm <sup>?</sup>. For tree level analysis, this density may exceed the accuracy needed. However, at this stage the testing area is rather small so downscaling the point cloud would become relevant when scaling up the research to a larger area.

After the first filtering steps above, the NDVI value is calculated. Any points that are below a NDVI threshold are considered non-vegetation or dead/unhealthy vegetation. In the testing phase this NDVI value has been lower than expected and seem differing based on sunlit vs shade differences. Due to the unexpected behaviour of the NDVI value, I have not decided on NDVI threshold yet and plan to look further into why this happens before I apply a NDVI filter.

Additionally, the normalised green and the MTVI2 value seem to classify trees well according to Chen et al. (2023) and Haboudane et al. (2004). These values are also calculated in this step of the process, but not yet used as a point cloud filter until I research potential thresholds.

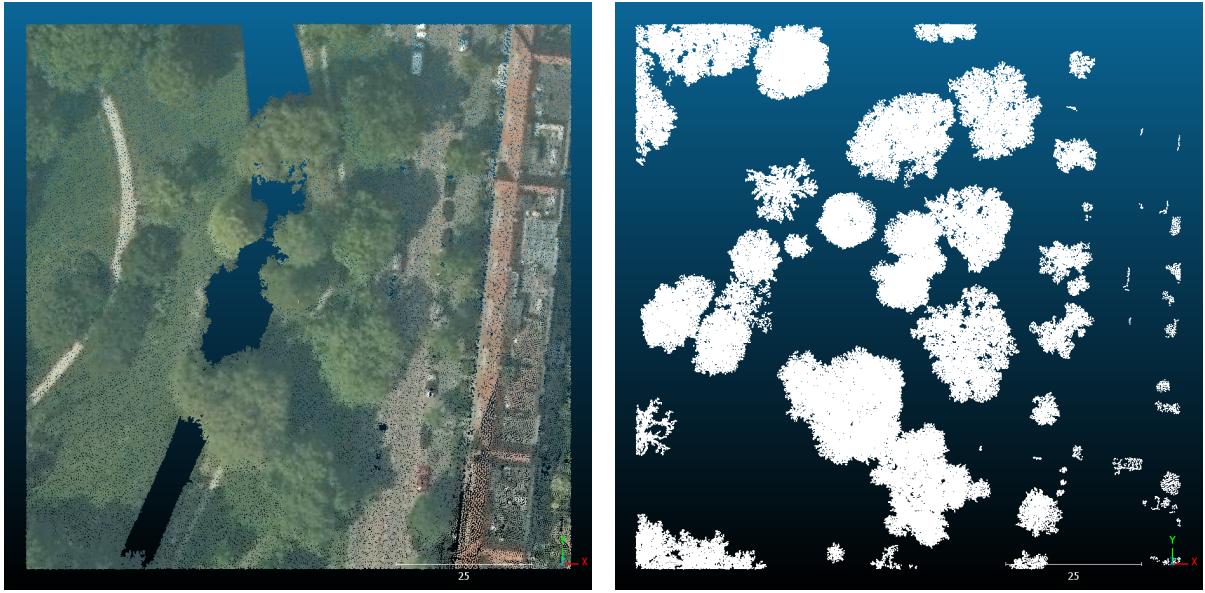


Figure 4: Top-view of the test area, before and after filtering on return number. It filters non-vegetation by excluding points that are the last return of a lidar pulse and pulses that have only 1 return. The point size in these figures is increased for visibility purposes.

## 4.2 Tree Instance Segmentation

Wang et al. (2018) provides an open-source, scalable algorithm to segment tree instances. It voxelises the point cloud and clusters within voxels per horizontal layer. There are three parameters that dictate the individualization results; searching radius, vertical resolution and minimal points per cluster.

To tune the algorithm so it works best for the selected area, I ran the segmentation algorithm for different parameter combinations. Afterwards the resulting files containing the segmentation are compared on total number of points and total number of tree clusters. Based on that a set of parameters can be selected. E.g. the amount of points should be as large as possible, ensuring the trees are as complete as they initially were. Additionally, the number of trees found is an indicator of performance. It should match, or be close to the number of trees in the test dataset. Since the test set is close to my home, I was able to visit the Wilhelmina Park to count the trees to see how many the segmentation should roughly output. In this case around 25, with some cases where the trunk split very low to the ground. The latter gave me reason to allow those 'twin-trees' to be passed as two individual trees, even though the originate from the same stem.

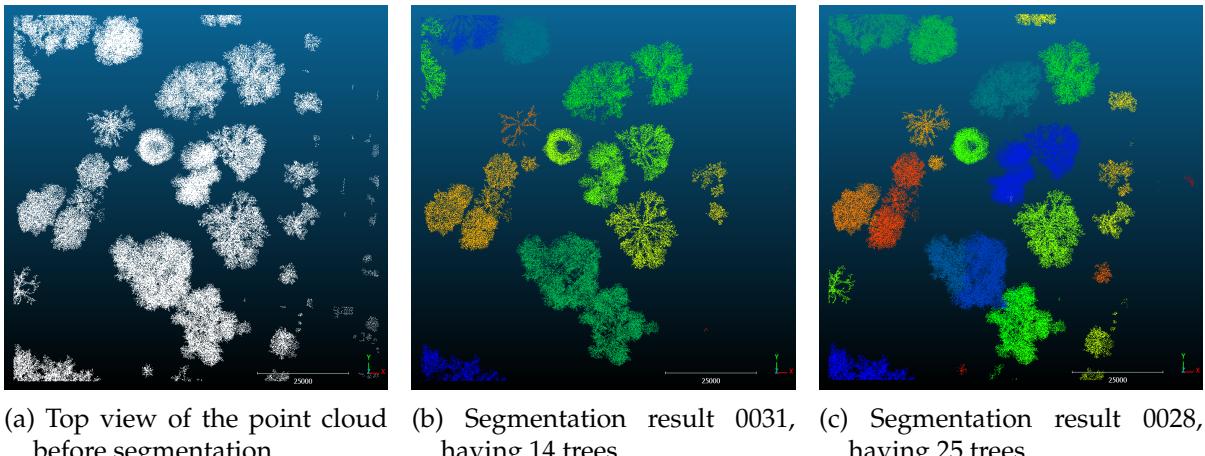


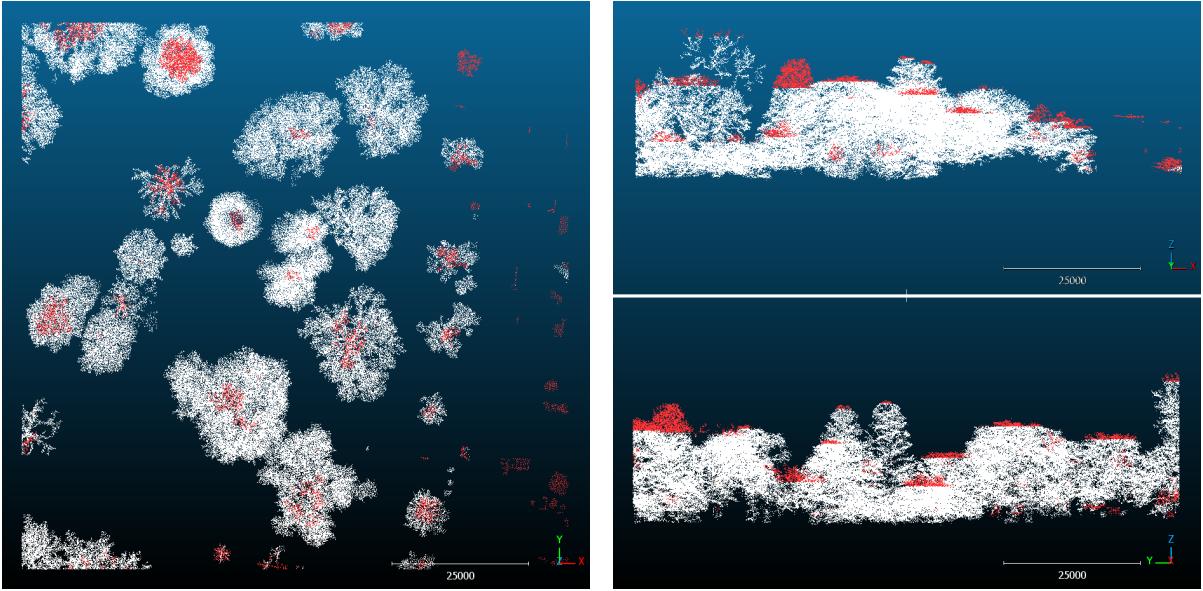
Figure 5: Example of the segmentation results. In figure 5b a top view of segmentation case 0031 is provided, having the parameters  $radius = 5000$  and  $v\_res = 50$  that yielded 14 tree instances. Figure 5c shows case number 0028, which segmented 25 trees using  $radius = 5000$  and  $v\_res = 200$ .

The 100m square area at the edge of the Wilhelmina Park is expected to have over 20 separate tree instances. The exact 'ground-truth' number is difficult to establish since I will allow tree clusters of similar trees to be a single tree after segmentation. At this moment the algorithm seems to work but I will need to tune the parameters using the computing power of the gilfoyle server.

Furthermore, the algorithm does not seem to classify the top layer of a tree instance. For P2 this issue is not solved yet, but put aside for further investigation.

## 4.3 Tree feature extraction

The instance segmentation code uses XYZ files as input and output. That means that only the coordinates of points are passed after segmentation. Since the coordinates of the points remain unaltered the found tree-id (tid) is reconnected to the point cloud after preprocessing



(a) Top-view of the unsegmented points.

(b) Front and Left view of the unsegmented points.

Figure 6: an example of the top part of trees not being included after segmentation. The reason for this exclusion is still unclear.

by adding it as a new attribute in LAS format.

The municipality of Delft provided a csv-file containing information about the public trees the municipality maintains. For each tree the file contains, among other attributes, a 2D coordinate in EPSG : 28992 Amersfoort New and a 'boomsortiment'; species.

To connect the dataset, I take for each tree instance the 2D projection of its convex hull. Then using the geopandas library, a simple spatial join appends a list of all species within the 2D hull to each tree instance.

The intermediate step that is given in Figure 7 can be used to assess the segmentation step. There are four cases distinguishable for 2D projected convex hull,  $H_2$  and public trees points,  $t_i$ :

1. **There is no tree inside the convex hull**,  $|t_i \cap H_2| = 0$

The tree cannot be validated using the municipality maintenance dataset and therefore the tree is excluded as training data. A private tree in someone's backyard would be a good example of this case.

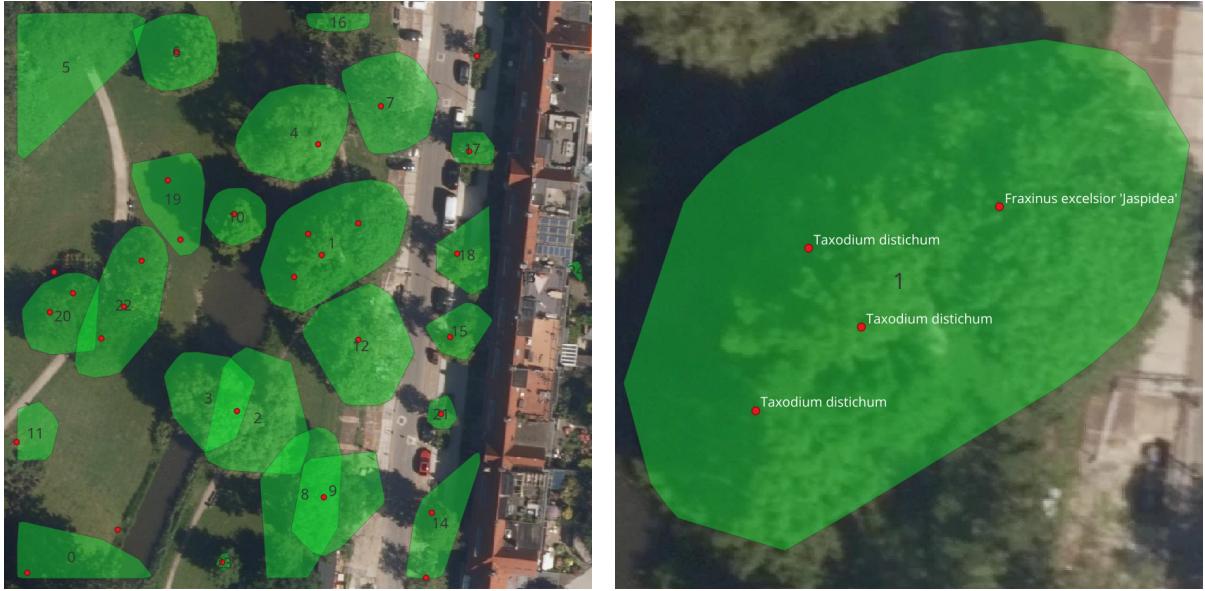
2. **There is exactly one tree inside the convex hull**,  $|t_i \cap H_2| = 1$

The tree species found can be assigned to the tree instance as a ground truth species. The species classifier will train that the point cloud belonging to the hull is of said species

3. **There are more than one tree inside the convex hull**  $|t_i \cap H_2| > 1$

The tree instance segmentation algorithm clustered some trees together as one tree. For CFD modelling purposes this is fine as long as the trees are the same species so that the cluster can be assigned an accurate porosity. Therefore the tree cluster is only included in the training data if the species of all trees in the convex hull are homogenous. I still have questions about how to handle this case, since I foresee that the classifier could take the shape of tree clusters into account, e.g. low and wide bush is always species  $x$ .

4. **A public tree is shared by  $n$  convex hulls**  $|t_i \cap H_2| = \frac{1}{n}$



(a) Top-view of the test area

(b) Zoomed top-view of the tree 1

Figure 7: Convex hulls of the trees after instance segmentation, including the points provided by the municipality of Delft. The tree hulls are shown in green, along with their *tid*. The public trees are red points. In the zoomed view the species is added as a label.

The segmentation algorithm split a tree with one base into two twin crowns. This is fine for the CFD, since both crowns originated from one trunk base, and are then in fact assigned the same species. Ideally, since it is one tree, it is best to merge a twin case back together as one tree. At this point, no triplet (or  $n > 2$ ) is detected, but theoretically this is possible.

There are some cases worth mentioning.

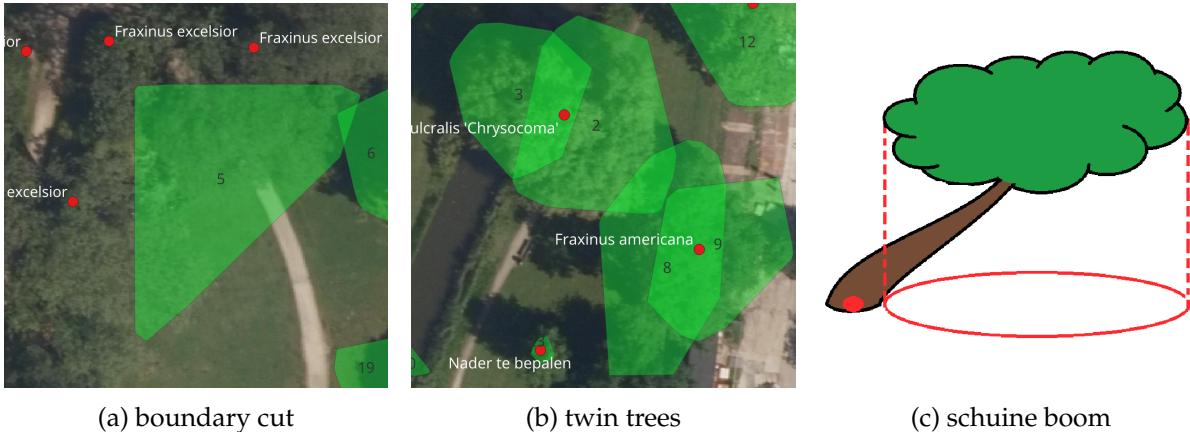


Figure 8: Special cases to address during automatic training data extraction.

Figure 8a shows a case where the point cloud is cut in a way where points belonging to a tree crown *is* included, but the tree base *is not*. Processing a larger area will reduce the number of trees close to the boundary.

Figure 8b shows two cases where the segmentation algorithm splits a tree that has two crowns coming from one trunk base. These cases should be identified and merged back together since

training on these as separate trees could influence the machine learning species classification. Merely assigning the two crowns the same species will not be sufficient because this will, in a later step, create two trunks. This case has not been fixed yet, but will be given priority since the tree trunks are solids and have a large impact on the flow in the CFD model.

Figure 8c showcases an example of a case that did not occur yet, but might in future analysis. A tree that grew under an angle because of a history of strong winds might have a trunk based outside the 2D projection of the crown's convex hull. At this point, such trees are excluded from the training data. If a species exists that tends to grow in this style, it is good to research this further. For now I will leave it at that.

#### 4.4 Species classification

Once the trees are segmented and saved as separate files, I will classify their species to obtain a species-based porosity for the CFD simulation.

- [use classifier to train on tree point clouds]
- [shape and geometrical attributes (from Chi et al. (2025))]
- [individual point features]
- [tree wide features (per layer/percentile/distance to centre)]

#### 4.5 3D geometry for CFD simulation

- [create geometry with alpha wrap]
- [potentially taking tree shape into account Geert Jan de Groot (2020)]
- [maybe not important for CFD though, ask clara]
- [porosity based on species]  
(no source yet)
- [meshing step between forest obj and snappyhexmesh? ask clara]

## 5 Time planning

- [make a gantt chart..]

### 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
AHN4	.laz	free to download
Boombeheer Delft	.csv	free upon request

---

Table 2: Datasets that I plan to use.

### Noteworthy 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

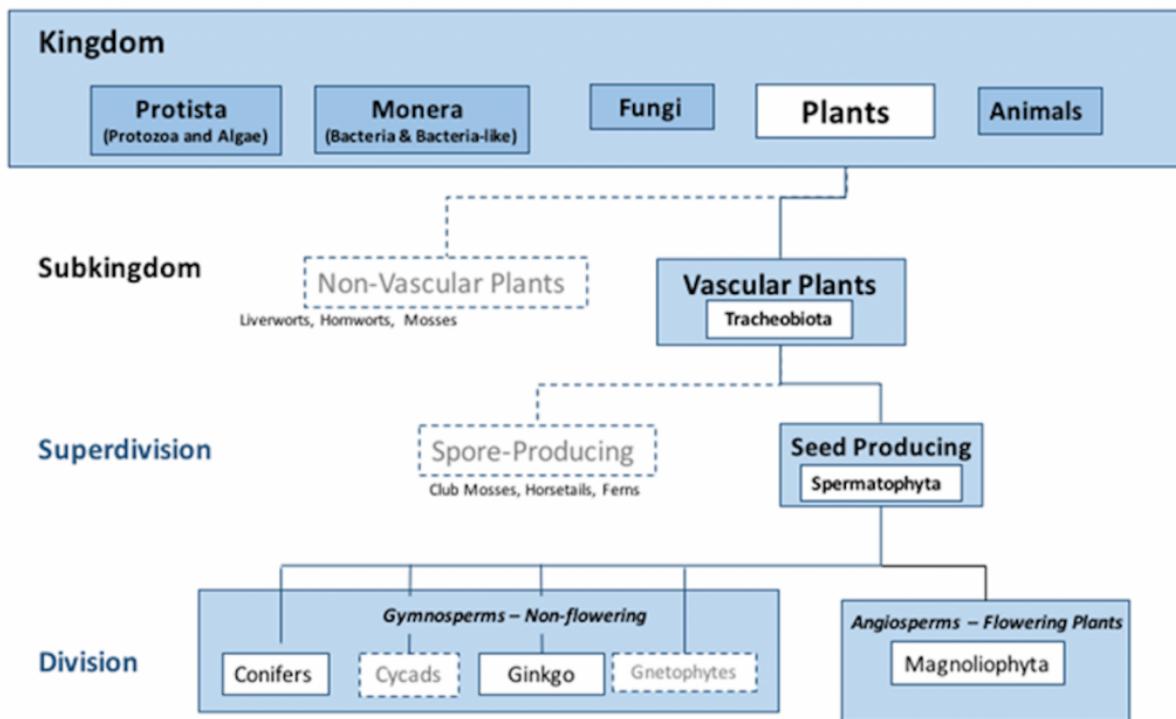


Figure 9: Tree of Life from the kingdom level, taken from Extension (2021)

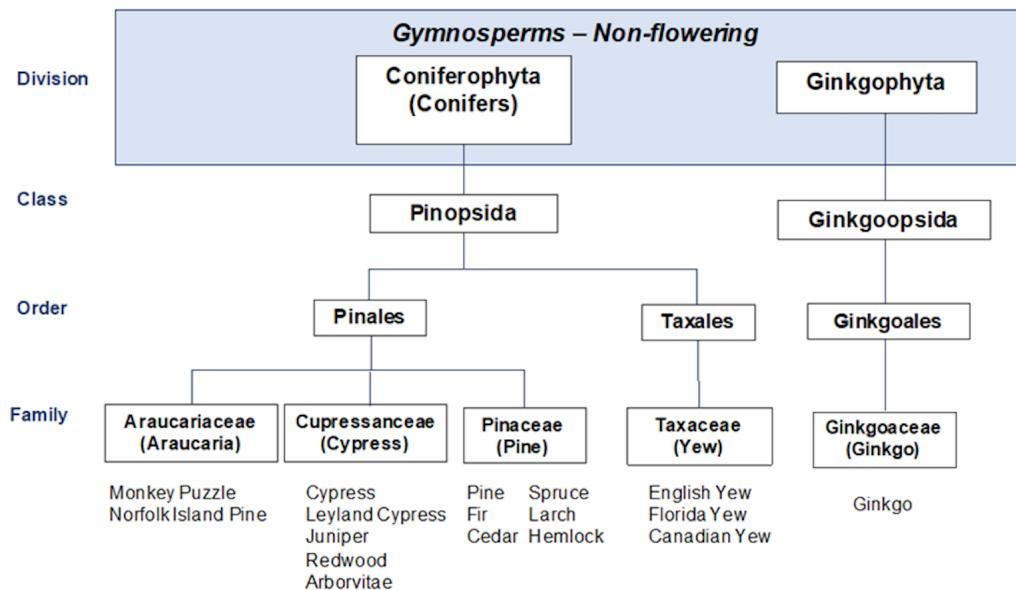


Figure 10: Tree of Life from the division level, taken from Extension (2021)

## 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.

- Chehreh, B., Moutinho, A., and Viegas, C. (2023). Latest Trends on Tree Classification and Segmentation Using UAV Data—A Review of Agroforestry Applications. *Remote Sensing*, 15(9):2263. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.
- 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.
- 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/pdf/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.
- Geert Jan de Groot (2020). *Automatic construction of 3D tree models in multiple levels of detail from airborne LiDAR data*. PhD thesis, TU Delft.
- 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.
- 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.

- Kulicki, M., Cabo, C., Trzciński, T., Bedkowski, J., and Stereńczak, K. (2024). Artificial Intelligence and Terrestrial Point Clouds for Forest Monitoring. *Current Forestry Reports*, 11(1):5.
- 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.
- 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.
- 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.
- 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.
- 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.

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 3: Tree classification features and descriptions.