Tree object detection using airborne images and LiDAR point clouds

Alexandre Bry

2024-07-17

Abstract

This is the abstract. It can be on multiple lines and contain **Markdown**.

Table of contents

Introduction

The goal of the internship was to study the possibility of combining LiDAR point clouds and aerial images in a deep learning model to identify individual trees. The two types of data are indeed complementary, as point clouds capture geometric shapes, while images capture colors. However, combining them into a format that allows a model to handle them simultaneously is not a straightforward task because they inherently have a very different spatial repartition and encoding.

In this work, I focused on one specific deep learning model, and tried to improve it by using more information from the LiDAR point cloud. To do this, I had to create my own tree annotations dataset, with which I also tried to study the ability of this new model to detect trees that are covered by other trees.

Chapter 1

State-of-the-art

1.1 Computer vision tasks related to trees

Before talking about models and datasets, let's define properly the task that this project focused on, in the midst of all the various computer vision tasks, and specifically those related to tree detection.

The first main differentiation between tree recognition tasks comes from the acquisition of the data. There are some very different tasks and methods using either ground data or aerial/satellite data. This is especially true when focusing on urban trees, since a lot of street view data is available [2].

This leads to the second variation, which is related to the kind of environment that we are interested in. There are mainly three types of environments, which among other things, influence the organization of the trees in space: urban areas, tree plantations and forests. This is important, because the tasks and the difficulty depends on the type of environment. Tree plantations are much easier to work with than completely wild forests, while urban areas contain various levels of difficulty ranging from alignment trees to private and disorganized gardens and parks. For this project, we mainly focused on urban areas, but everything should still be applicable to tree plantations and forests.

Then, the four fundamental computer vision tasks have their application when dealing with trees [27]:

- Classification, although this is quite rare for airborne tree applications since there are multiple trees on each image most of the time
- Detection, which consists in detecting objects and placing boxes around them
- Semantic segmentation, which consists in associating a label to every pixel of an image

- Instance segmentation, which consists in adding a layer of complexity to semantic segmentation by also differentiating between the different instances of each class

These generic tasks can be extended by trying to get more information about the trees. The most common information are the species and the height, but some models also try to predict the health of the trees, or their carbon stock.

In this work, the task that is tackled is the detection of trees, with a special classification between several labels related to the discrepancies between the different kinds of data. The kind of model that is used would also have allowed to focus on some more advanced tasks, by replacing detection with instance segmentation and asking the model to also predict the species. But due to the difficulties regarding the dataset, a simpler task with a simpler dataset was used, without compromising the ability to experiment with different possible improvements of the model. The difficulties and the experiments are developed below.

1.2 Datasets

1.2.1 Requirements

Before presenting the different promising datasets and the reasons why they were not fully usable for the project, let's enumerate the different conditions and requirements for the tree instance segmentation task:

- Multiple types of data:
  - Aerial RGB images
  - LiDAR point clouds (preferably aerial)
  - (Optional) Aerial infrared images
- Tree crown annotations or bounding boxes
- High-enough resolution:
  - For images, about 25 cm
  - For point clouds, about 10 cm

Here are the explanations for these requirements. As for the types of data, RGB images and point clouds are required to experiment on the ability of the model to combine the two very different kinds of information they hold. Having infrared data as well could be beneficial, but it was not necessary. Regarding tree annotations, it was necessary to have a way to spatially identify them individually, using crown contours or simply bounding boxes. Since the model outputs bounding boxes, any kind of other format could easily be transformed to bounding boxes. Finally, the resolution had to be high enough to identify individual trees and be able to really use the data. For the point clouds especially, the whole idea was to see if and how the topology of the trees could be learnt, using at least the trunks and even the biggest branches if possible. Therefore, even if they are not really comparable, this is the reason why the required resolution is more precise for the point clouds.

Unfortunately, none of the datasets that I found matched all these criteria. Furthermore, I didn't find any overlapping datasets that I could merge to create a dataset with all the required types of data. In the next parts, I will go through the different kinds of datasets that exist, the reasons why they did not really fit for the project and the ideas I got when searching for a way to use them.

1.2.2 Existing tree datasets

As explained above, there were quite a lot of requirements to fulfill to have a complete dataset usable for the task. This means that almost all the available datasets were missing something, as they were mainly focusing on using one kind of data and trying to make the most out of it, instead of trying to use all the types of data together.

The most comprehensive list of tree annotations datasets was published in OpenForest [22]. FoMo-Bench [5] also lists several interesting datasets, even though most of them can also be found in OpenForest. Without enumerating all of them, there were multiple kinds of datasets that all have their own flaws regarding the requirements I was looking for.

Firstly, there are the forest inventories. TALLO [18] is probably the most interesting one in this category, because it contains a lot of spatial information about almost 500K trees, with their locations, their crown radii and their heights. Therefore, everything needed to localize trees is in the dataset. However, I didn't manage to find RGB images or LiDAR point clouds of the areas where the trees are located, making it impossible to use these annotations to train tree detection.

Secondly, there are the RGB datasets. ReforesTree [26] and MillionTrees [31] are two of them and the quality of their images are high. The only drawback of these datasets is obviously that they don't provide any kind of point cloud, which make them unsuitable for the task.

Thirdly, there are the LiDAR datasets, such as [19] and [24]. Similarly to RGB datasets, they lack one of the data source for the task I worked on. But unlike them, they have the advantage that the missing data could be much easier to acquire from another source, since RGB aerial or satellite images are much more common than LiDAR point clouds. However, this solution was abandoned for two main reasons. First it is quite challenging to find the exact locations where the point clouds were acquired. Then, even when the location is known, it is often in the middle of a forest where the quality of satellite imagery is very low.

Finally, I also found two datasets that had RGB and LiDAR components. The first one is MDAS [15]. This benchmark dataset encompasses RGB images, hyperspectral images and Digital Surface Models (DSM). There were however two major flaws. The obvious one was that this dataset was created with land semantic segmentation tasks in mind, so there was no tree annotations. The less obvious one was that a DSM is not a point cloud, even though it is some kind of 3D information and was often created using a LiDAR point cloud. As a consequence, I would have been very limited in my ability to use the point cloud.

The only real dataset with RGB and LiDAR came from NEON [32]. This dataset contains exactly all the data I was looking for, with RGB images, hyperspectral images and LiDAR point clouds. With 30975 tree annotations, it is also a quite large dataset, spanning across multiple various forests. The reason why I decided not to use it despite all this is that at the beginning of the project, I thought that the quality of the images and the point clouds was too low. Looking back on this decision, I think that I probably could have worked with this dataset and gotten great results. This would have saved me the time spent annotating the trees for my own dataset, which I will talk more about later. My decision was also influenced by the quality of the images and the point clouds available in the Netherlands, which I will talk about in the next section.

1.2.3 Public data

After rejecting all the available datasets I had found, the only solution I had left was to create my own dataset. I won't dive too much in this process that I will explain in Section 3. I just want to mention all the publicly available datasets that I used or could have used to create this custom dataset.

For practical reasons, the two countries where I mostly searched for available data are France and the Netherlands. I was looking for three different data types independently:

- RGB (and eventually infrared) images
- LiDAR point clouds
- Tree annotations

These three types of data are available in similar ways in both countries, although the Netherlands have a small edge over France. RGB images are really easy to find in France with the BD ORTHO [16] and in the Netherlands with the Luchtfotos [3], but the resolution is better in the Netherlands (8 cm vs 20 cm). Hyperspectral images are also available in both countries, although for those the resolution is only 25 cm in the Netherlands.

As for LiDAR point clouds, the Netherlands have a small edge over France, because they have already completed their forth version covering the whole country with AHN4 [1], and are working on the fifth version. In France, data acquisition for the first LiDAR point cloud covering the whole country started a few years ago [17]. It is not yet finished, even though data is already available for half of the country. The other advantage of the data from Netherlands regarding LiDAR point clouds is that all flights are performed during winter, which allows light beams to penetrate more deeply in trees and reach trunks and branches. This is not the case in France.

The part that is missing in both countries is related to tree annotations. Many municipalities have datasets containing information about all the public trees they handle. This is for example the case for Amsterdam [12] and Bordeaux [4]. However, these datasets cannot really be used as ground truth for a custom dataset for several reasons. First, many of them do not contain coordinates indicating the position of each tree in the city. Then, even those that contain coordinates are most of the time missing any kind of information allowing to deduce a bounding box for the tree crowns. Finally, even if they did contain everything, they only focus on public trees, and are missing every single tree located in a private area. Since public and private areas are obviously imbricated in all cities, it means that any area we try to train the model on would be missing all the private trees, making the training process impossible because we cannot have only a partial annotation of images.

The other tree annotation source that we could have used is Boomregister [8]. This work covers the whole of the Netherlands, including public and private trees. However, the precision of the masks is far from perfect, and many trees are missing or incorrectly segmented, especially when they are less than 9 m heigh or have a crown diameter smaller than 4 m. Therefore, even it is a very impressive piece of work, we thought that it could not be used as training data for a deep learning models due to its biases and imperfections.

1.2.4 Dataset augmentation techniques

When a dataset is too small to train a model, there are several ways of artificially enlarging it.

The most common way to do it is to randomly apply deterministic or random transformations to the data, during the training process, to be able to generate several unique and different realistic data instances from one real data instance. There are a lot of different transformations that can be applied to images, divided into two categories: pixel-level and spatial-level [6]. Pixel-level transformations modify the value of individual pixels, by applying different filters, such as random noise, color shifts and more complex effects like fog and sun flare. Spatial-level transformations modify the spatial arrangement of the image, without changing the pixel values. In other words, these transformations move the pixels in the image. The transformations range from simple rotations and croppings to complex spatial distortions. In the end, all these transformations are simply producing one artificial image out of one real image.

Another way to enlarge a dataset is to instead generate completely new input data sharing the same properties as the initial dataset. This can be done using Generative Adversarial Networks (GAN). These models usually have two parts, a generator and a discriminator, which are trained in parallel. The generator learns to produce realistic artificial data, while the discriminator learns to identify real data and artificial data produced by the generator. If the training is successful, we can then use the generator and random seeds to generate random but realistic artificial data similar to the dataset. This method has for example been successfully used to generate artificial tree height maps [28].

1.3 Algorithms and models

In this section, the different algorithms and methods are grouped according to the type of data they use as input.

1.3.1 Images only

Then, there are methods that perform tree detection using only visible or hyperspectral images or both. Several different algorithms have been developed to analytically delineate tree crowns from RGB images, by using the particular shape of the trees and its effect on images [13]. Without diving into the details, here are a few of them. The watershed algorithm identifies trees to inverted watersheds in the grey-scale image and tree crowns frontiers are found by incrementally flooding the watersheds [29]. The local maxima filtering uses the intensity of the pixels in the grey-scale image to identify the brightest points locally and use them as treetops [35]. Reversely, the valley-following algorithm uses the darkest pixels which are considered as the junctions between the trees since shaded areas are the lower part of the tree crowns [14]. Another interesting algorithm is template matching. This algorithm simulates the appearance of simple tree templates with the light effects, and tries to identify similar patterns in the grey-scale image [23]. Combinations of these techniques and others have also been proposed.

But with the recent developments of deep learning in image analysis, deep learning models are increasingly used to detect trees using RGB images. In some cases, deep learning is used to extract features that can then be the input of one of the algorithms described above. One example is the use of two neural networks to predict masks, outlines and distance transforms which can then be the input of a watershed algorithm [11]. In other cases, a deep learning model is responsible of directly detecting tree masks or bounding boxes, often using CNNs, given the images [33].

1.3.2 LiDAR only

Some of the methods to identify individual trees use LiDAR data only. There are a lot of different ways to use and analyze point clouds, but the one that is mostly used for trees is based on height maps, or Canopy Height Models (CHM).

A CHM is a raster computed as the subtraction of the Digital Terrain Model (DTM) to the Digital Surface Model (DSM). What it means is that a CHM contains the height above ground of the highest point in the area corresponding to each pixel. This CHM can for example be used as the input raster for the watershed algorithm, as it contains the height values that can be used to determine local maxima [20]. A lot of different analytical methods and variations of the simple CHM were proposed to perform individual tree detection, but in the end, most of them still the concept of local maxima [10, 30]. A CHM can also be used as the input of any kind of convolutional neural network (CNN) because it is shaped exactly like any image. This allows to use a lot of different techniques usually applied to object detection in images.

Then, even though I finally used an approach similar to the CHM, I want to mention other kinds of deep learning techniques that exist and could potentially leverage all the information contained in a point cloud. These techniques can be divided in two categories: projection-based and point-based methods [9]. The main difference between the two is that projection-based techniques are based on grids while point-based methods take unstructured point clouds as input. Among projection-based methods, the most basic method is 2D CNN, which is how CHM can be processed. Then, multiview representation tries to tackle the 3D aspect by projecting the point cloud in multiple directions before merging them together. To really deal with 3D data, volumetric grid representation consists in using 3D occupancy grids, which are processed using 3D CNNs. Among point-based methods, there are methods based on PointNet, which are able to extract features and perform the classical computer vision tasks by taking point clouds as input. Finally, Convolutional Point Networks use a continuous generalization of convolutions to apply convolution kernels to arbitrarily distributed point clouds.

1.3.3 LiDAR and images

Let's now talk about the models of interest for this work, which are machine learning pipelines using both LiDAR point cloud data and RGB images.

The first pipeline [25] uses a watershed algorithm to extract crown boundaries, before extracting individual tree features from the LiDAR point cloud, hyperspectral and RGB images. These features are then used by a random forest classifier to identify which species the tree belongs to. This pipeline therefore makes the most out of all data to identify species, but sticks to an improved variant of the watershed for individual tree segmentation, which only uses a CHM raster.

Other works focused on using only one model that is able to take both the CHM and the RGB data as input and combine them to make the most out of all the available data. Among other models, there are for example ACE R-CNN [21], an evolution of Mask region-based convolution neural network (Mask R-CNN) and AMF GD YOLOv8 [36], an evolution of YOLOv8. These two models have proven to give much better results when using both the images and the LiDAR data as a CHM thant when using only one of them.

Chapter 2

Objectives and motivations

2.1 Multiple layers of CHM

Benchmark of 8 methods using LiDAR [10]: one of the methods uses multiple CHM layers, computed iteratively by removing everything in the 0.5 m below the previous CHM

2.2 Hidden trees

[30] shows that in forests, you can have up to more than 50% of the trees which crowns are completely or partially covered by other trees.

Chapter 3

Dataset creation

The highest resolution of the CHM which keeps a high enough quality depends entirely on the density of the point cloud. Also, depending on the season when the point cloud is acquired, using a CHM might imply throwing away the majority of the information contained in the point cloud.

3.1 Definition and content

3.2 Challenges and solutions

- Shift between RGB images, CIR images and LiDAR point clouds due to images not being perfectly orthonormal
- Variations of the trees over time, with new small trees being planted and old trees being cut off
- Not so easy to define what we consider as a tree and identify them. Problems with bushes for example. This problem is also mentioned in another paper [34].

3.3 Augmentation methods

Chapter 4

Deep learning model

4.1 Architecture

4.2 Training loop

Chapter 5

Results

5.1 Evaluation method

sortedAP: [7]

5.2 Training parameters

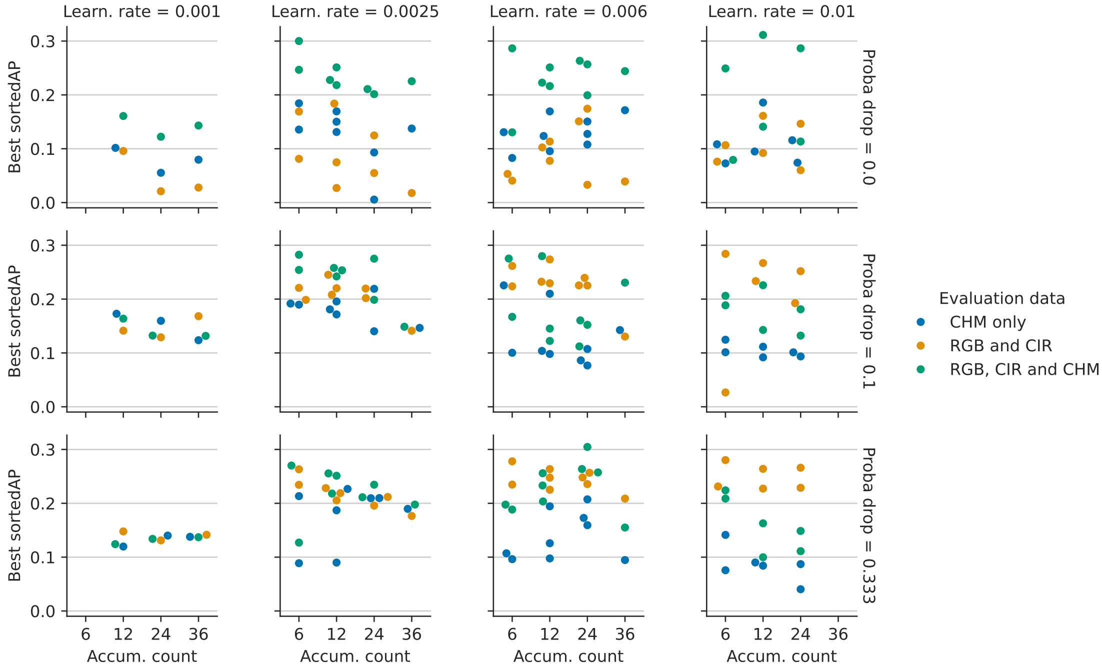

Figure 5.1: Results with different training parameters

Source: Article Notebook

5.3 Data used

5.4 CHM layers

5.5 Hard trees

Chapter 6

Discussion

6.1 Dataset

- DeepForest: A Python package for RGB deep learning tree crown delineation [33]: uses only RGB data to detect trees, but uses LiDAR to create millions of annotations of moderate quality to pre-train the model, before using around 10,000 hand-annotations to finalize and specialize the training on a certain area.

6.2 Combination of data types

Conclusion

Blablabla

Source: Article Notebook

Bibliography

[1] Actueel Hoogtebestand Nederland. *AHN4 - Actual Height Model of the Netherlands*. Accessed: 2024-07-09. 2020. URL: https://www.ahn.nl/.

[2] Tito Arevalo-Ramirez et al. "Challenges for computer vision as a tool for screening urban trees through street-view images". In: *Urban Forestry & Urban Greening* 95 (2024), p. 128316. ISSN: 1618-8667. DOI: 10.1016/j.ufug.2024.128316. URL: https://www.sciencedirect.com/science/article/pii/S1618866724001146.

[3] Beeldmateriaal Nederland. *Luchtfoto's (Aerial Photographs)*. Accessed: 2024-07-09. 2024. URL: https://www.beeldmateriaal.nl/luchtfotos.

[4] Bordeaux Métropole. *Patrimoine arboré de Bordeaux Métropole (Tree Heritage of Bordeaux Metropole)*. Accessed: 2024-07-09. 2024. URL: https://opendata.bordeaux-metropole.fr/explore/dataset/ec_arbre_p/information/?disjunctive.insee.

[5] Nikolaos Ioannis Bountos, Arthur Ouaknine, and David Rolnick. "FoMo-Bench: a multi-modal, multi-scale and multi-task Forest Monitoring Benchmark for remote sensing foundation models". In: *arXiv preprint arXiv:2312.10114* (2023). URL: https://arxiv.org/abs/2312.10114.

[6] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[7] Long Chen et al. "SortedAP: Rethinking Evaluation Metrics for Instance Segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2023, pp. 3923–3929. URL: https://openaccess.thecvf.com/content/ICCV2023W/BIC/html/Chen_SortedAP_Rethinking_Evaluation_Metrics_for_Instance_Segmentation_ICCVW_2023_paper.html.

[8] Coöperatief Boomregister U.A. *Boom Register (Tree Register)*. Accessed: 2024-07-11. 2014. URL: https://boomregister.nl/.

[9] Ahmed Diab, Rasha Kashef, and Ahmed Shaker. "Deep Learning for LiDAR Point Cloud Classification in Remote Sensing". In: *Sensors (Basel)* 22.20 (Oct. 2022), p. 7868. DOI: 10.3390/s22207868.

[10] Lothar Eysn et al. "A Benchmark of Lidar-Based Single Tree Detection Methods Using Heterogeneous Forest Data from the Alpine Space". In:

Forests 6.5 (2015), pp. 1721–1747. ISSN: 1999-4907. DOI: 10.3390/f6051721. URL: https://www.mdpi.com/1999-4907/6/5/1721.

[11] Maximilian Freudenberg, Paul Magdon, and Nils Nölke. "Individual tree crown delineation in high-resolution remote sensing images based on U-Net". In: *Neural Computing and Applications* 34.24 (2022), pp. 22197–22207. ISSN: 1433-3058. DOI: 10.1007/s00521-022-07640-4.

[12] Gemeente Amsterdam. *Bomenbestand Amsterdam (Amsterdam Tree Dataset)*. Accessed: 2024-07-09. 2024. URL: https://maps.amsterdam.nl/open_geodata/?k=505.

[13] Marilia Ferreira Gomes and Philippe Maillard. "Detection of Tree Crowns in Very High Spatial Resolution Images". In: *Environmental Applications of Remote Sensing*. Ed. by Maged Marghany. Rijeka: IntechOpen, 2016. Chap. 2. DOI: 10.5772/62122.

[14] François A Gougeon et al. "Automatic individual tree crown delineation using a valley-following algorithm and rule-based system". In: *Proc. International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry, Victoria, British Columbia, Canada*. Citeseer. 1998, pp. 11–23. URL: https://d1ied5g1xfgpx8.cloudfront.net/pdfs/4583.pdf.

[15] J. Hu et al. "MDAS: a new multimodal benchmark dataset for remote sensing". In: *Earth System Science Data* 15.1 (2023), pp. 113–131. DOI: 10.5194/essd-15-113-2023. URL: https://essd.copernicus.org/articles/15/113/2023/.

[16] Institut national de l'information géographique et forestière (IGN). *BD ORTHO*. Accessed: 2024-07-09. 2021. URL: https://geoservices.ign.fr/bdortho.

[17] Institut national de l'information géographique et forestière (IGN). *LiDAR HD*. Accessed: 2024-07-09. 2020. URL: https://geoservices.ign.fr/lidarhd.

[18] Tommaso Jucker et al. "Tallo: A global tree allometry and crown architecture database". In: *Global Change Biology* 28.17 (2022), pp. 5254–5268. DOI: 10.1111/gcb.16302. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/gcb.16302. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/gcb.16302.

[19] Ekaterina Kalinicheva et al. *Multi-Layer Modeling of Dense Vegetation from Aerial LiDAR Scans*. 2022. arXiv: 2204.11620 `[cs.CV]`. URL: https://arxiv.org/abs/2204.11620.

[20] Doo-Ahn Kwak et al. "Detection of individual trees and estimation of tree height using LiDAR data". In: *Journal of Forest Research* 12.6 (2007), pp. 425–434. ISSN: 1610-7403. DOI: 10.1007/s10310-007-0041-9.

[21] Yingbo Li et al. "ACE R-CNN: An Attention Complementary and Edge Detection-Based Instance Segmentation Algorithm for Individual Tree Species Identification Using UAV RGB Images and LiDAR Data". In: *Remote Sensing* 14.13 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14133035. URL: https://www.mdpi.com/2072-4292/14/13/3035.

[22] Arthur Ouaknine et al. *OpenForest: A data catalogue for machine learning in forest monitoring*. 2023. arXiv: 2311.00277 `[cs.CV]`.

[23] Richard James Pollock. “The automatic recognition of individual trees in aerial images of forests based on a synthetic tree crown image model”. AAINN14815. PhD thesis. 1996. ISBN: 0612148157. URL: https://dx.doi.org/10.14288/1.0051597.

[24] Stefano Puliti et al. *FOR-instance: a UAV laser scanning benchmark dataset for semantic and instance segmentation of individual trees*. 2023. arXiv: 2309.01279 [cs.CV]. URL: https://arxiv.org/abs/2309.01279.

[25] Haiming Qin et al. “Individual tree segmentation and tree species classification in subtropical broadleaf forests using UAV-based LiDAR, hyperspectral, and ultrahigh-resolution RGB data”. In: *Remote Sensing of Environment* 280 (2022), p. 113143. ISSN: 0034-4257. DOI: 10.1016/j.rse.2022.113143. URL: https://www.sciencedirect.com/science/article/pii/S0034425722002577.

[26] Gyri Reiersen et al. *ReforesTree: A Dataset for Estimating Tropical Forest Carbon Stock with Deep Learning and Aerial Imagery*. 2022. arXiv: 2201.11192 [cs.CV]. URL: https://arxiv.org/abs/2201.11192.

[27] Anastasiia Safonova et al. “Olive Tree Biovolume from UAV Multi-Resolution Image Segmentation with Mask R-CNN”. In: *Sensors* 21.5 (2021), p. 1617. ISSN: 1424-8220. DOI: 10.3390/s21051617. URL: https://www.mdpi.com/1424-8220/21/5/1617.

[28] Chenxin Sun et al. “Individual Tree Crown Segmentation and Crown Width Extraction From a Heightmap Derived From Aerial Laser Scanning Data Using a Deep Learning Framework”. In: *Frontiers in Plant Science* 13 (2022). ISSN: 1664-462X. DOI: 10.3389/fpls.2022.914974. URL: https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.914974.

[29] L. Vincent and P. Soille. “Watersheds in digital spaces: an efficient algorithm based on immersion simulations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.6 (1991), pp. 583–598. DOI: 10.1109/34.87344.

[30] Yunsheng Wang et al. “International Benchmarking of the Individual Tree Detection Methods for Modeling 3-D Canopy Structure for Silviculture and Forest Ecology Using Airborne Laser Scanning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 54.9 (2016), pp. 5011–5027. DOI: 10.1109/TGRS.2016.2543225.

[31] Ben Weinstein. *MillionTrees*. 2023. URL: https://milliontrees.idtrees.org/ (visited on 07/08/2024).

[32] Ben Weinstein, Sergio Marconi, and Ethan White. *Data for the NeonTreeEvaluation Benchmark (0.2.2)*. 2022. DOI: 10.5281/zenodo.5914554.

[33] Ben G. Weinstein et al. “DeepForest: A Python package for RGB deep learning tree crown delineation”. In: *Methods in Ecology and Evolution* 11.12 (2020), pp. 1743–1751. DOI: 10.1111/2041-210X.13472. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.13472. URL: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13472.

[34] Ben G. Weinstein et al. "Individual Tree-Crown Detection in RGB Imagery Using Semi-Supervised Deep Learning Neural Networks". In: *Remote Sensing* 11.11 (2019). ISSN: 2072-4292. DOI: 10.3390/rs11111309. URL: https://www.mdpi.com/2072-4292/11/11/1309.

[35] Mike Wulder, K.Olaf Niemann, and David G. Goodenough. "Local Maximum Filtering for the Extraction of Tree Locations and Basal Area from High Spatial Resolution Imagery". In: *Remote Sensing of Environment* 73.1 (2000), pp. 103–114. ISSN: 0034-4257. DOI: 10.1016/S0034-4257(00)00101-2. URL: https://www.sciencedirect.com/science/article/pii/S0034425700001012.

[36] Hao Zhong et al. "Individual Tree Species Identification for Complex Coniferous and Broad-Leaved Mixed Forests Based on Deep Learning Combined with UAV LiDAR Data and RGB Images". In: *Forests* 15.2 (2024). ISSN: 1999-4907. DOI: 10.3390/f15020293. URL: https://www.mdpi.com/1999-4907/15/2/293.