remote sensing

Article

Individual Tree Detection in Urban ALS Point Clouds with 3D Convolutional Networks

Stefan Schmohl [1,*], Alejandra Narváez Vallejo [2] and Uwe Soergel [1]

[1] Institute for Photogrammetry, University of Stuttgart, 70174 Stuttgart, Germany; uwe.soergel@ifp.uni-stuttgart.de
[2] Institute of Landscape Planning and Ecology, University of Stuttgart, 70174 Stuttgart, Germany; alejandra.narvaez-vallejo@ilpoe.uni-stuttgart.de
* Correspondence: stefan.schmohl@ifp.uni-stuttgart.de

Abstract: Since trees are a vital part of urban green infrastructure, automatic mapping of individual urban trees is becoming increasingly important for city management and planning. Although deep-learning-based object detection networks are the state-of-the-art in computer vision, their adaptation to individual tree detection in urban areas has scarcely been studied. Some existing works have employed 2D object detection networks for this purpose. However, these have used three-dimensional information only in the form of projected feature maps. In contrast, we exploited the full 3D potential of airborne laser scanning (ALS) point clouds by using a 3D neural network for individual tree detection. Specifically, a sparse convolutional network was used for 3D feature extraction, feeding both semantic segmentation and circular object detection outputs, which were combined for further increased accuracy. We demonstrate the capability of our approach on an urban topographic ALS point cloud with 10,864 hand-labeled ground truth trees. Our method achieved an average precision of 83% regarding the common 0.5 intersection over union criterion. 85% of the stems were found correctly with a precision of 88%, while tree area was covered by the individual tree detections with an F1 accuracy of 92%. Thereby, we outperformed traditional delineation baselines and recent detection networks.

Keywords: deep learning; sparse convolutional network; object detection; airborne laser scanning; vegetation

check for updates

Citation: Schmohl, S.; Narváez Vallejo, A.; Soergel, U. Individual Tree Detection in Urban ALS Point Clouds with 3D Convolutional Networks. *Remote Sens.* **2022**, *14*, 1317. https://doi.org/10.3390/rs14061317

Academic Editors: Dong Chen, Jiju Poovvancheri, Zhengxin Zhang and Liqiang Zhang

Received: 9 February 2022
Accepted: 7 March 2022
Published: 9 March 2022

1. Introduction

Urban trees are an essential part of cities' green infrastructure and draw particular attention in city development and environment preservation. High and medium vegetation, e.g., trees and shrubs, have positive effects on urban quality of living. Ecosystem services provided by urban forests include, but are not limited to, a reduction of air pollution [1], an improvement of locale climate through shading and evapotranspiration [2–5], energy savings [6], and the enrichment of biodiversity [7].

Municipal administrations often use dedicated systems to manage their green spaces. Because trees are of particular concern, they are managed in special tree cadasters, containing information such as position, height, crown and trunk diameter, health, and species of the individual trees. These inventories represent vital information for traffic safety, monitoring, and strategic planning of urban green infrastructure, as well as a wide range of research fields [8–11].

However, two major issues interfere with the completeness of these databases. First of all, the data for tree cadasters are traditionally acquired by field-based surveys, which are time-consuming and costly undertakings. Consequently, many of the descriptive and structural attributes might be missing, and the feasibility of continuous monitoring is limited. Furthermore, private areas such as gardens often lie outside immediate on-site access and thus are not surveyed. Existing municipal tree inventories are therefore usually limited to trees on public grounds or along streets [8,11] even though vegetation on private ground is a considerable part of cities' green infrastructure [12,13]. Thus, an automatic and complete mapping through individual tree detection (ITD) in remote sensing data is of particular scientific and practical interest [14,15].

Large-scale mapping of areal tree coverage is traditionally performed by satellite, airborne imagery, or airborne laser scanning (ALS) in increasingly higher resolutions [16]. In contrast, the detection of individual urban trees is especially challenging due to their diverse spatial arrangements (as singulars, groups, or dense stands), occlusions, as well as varying shapes and sizes. As pointed out by [17], while simple methods such as local maxima filters and basic segmentation of a height model perform generally well, more sophisticated techniques are often overly fine-tuned by hand and specialized to certain data conditions and assumptions in regards to tree structure. In contrast, data-driven deep learning techniques recently became popular for vegetation mapping [18]. Once a basic neural network methodology is found, adaption to new data distributions is straightforward by labeling and re-training on new reference data. As ALS provides not only high spatial resolution height information, but also fully 3D point clouds that capture sub-canopy tree structure, 3D algorithms can result in higher accuracy for ITD. However, thus far, only two-dimensional deep learning networks have been applied to individual tree detection.

In this work, we investigated the use of a deep 3D single-shot detector for individual urban tree identification in ALS point clouds, in contrast to previous works that only used two-dimensional detection networks. It is an extension of our previous proof-of-concept study in [19]. Compared to this previous version, small, but significant improvements to the pipeline enhanced the accuracy by a large margin. We tested our methodology on a new, large-scale dataset and bore the relation to a greater number of baselines. An extensive ablation study analyzed the impact of the individual adjustments on overall performance.

This paper is organized as follows: First, we review related works in Section 2. Then, we describe the architecture of the neural network and its training process in Section 3.1 and the data material used in this work in Section 3.2. After outlining the baselines (Section 3.3) and therewith comparing our results (Section 4), Section 4.4 further analyzes the impact of different components of our methodology in an ablation study before the final discussion in Section 5.

2. Related Work

2.1. Individual Tree Detection in (Urban) Forests

Tree detection and delineation from airborne laser scanning is a widely studied task in forestry [20–23]. Numerous methods have been developed to identify individual trees. Most common approaches leverage a light-detection-and-ranging (LiDAR)-based canopy height model (CHM), which is a normalized digital surface model (nDSM) over exclusively tree-covered areas. In such homogeneous scenes, single tree detection can be achieved by searching for local maxima in the CHM. For delineation, a variety of unsupervised segmentation methods have been used in the literature. Starting from local maxima as seed points, region growing was used to iteratively add adjacent pixels to the initial tree segments [24–26]. A particular popular segmentation method is the watershed transform of the inverted CHM [27–29]. In [30], tree points were clustered through slope climbing.

The capability of LiDAR to penetrate vegetation can provide a more complete view of individual trees in the 3D point cloud compared to the CHM surface, including stems and understory trees. Therefore, 3D clustering methods have been proposed [31–33], including three-dimensional mean-shift segmentation [34–36]. Alternatively, template matching was used to fit three-dimensional geometric models to the vertical height information in the CHM or directly into the point cloud [30,37–40]. Other works solved tree delineation by translating it into graph optimization problems [17,41].

In urban areas, non-tree regions have to be masked out before applying such tree detection methods. This can be achieved with vegetation indices from multispectral imagery such as the normalized difference vegetation index (NDVI) or geometrical features from the point cloud [42–48]. ALS provides additional features beneficial for vegetation recognition. Laser beams emitted by an ALS sensor can partly penetrate higher vegetation such as trees and bushes and so record multiple returning echos, resulting in characteristic representations of vegetation in the resulting point clouds [49–51]. Generally speaking, such unsupervised segmentation and classification methods usually require careful pre- and post-processing to avoid under- and over-segmentation.

2.2. MLS and Other Data Sources

Other kinds of data have been studied for urban tree detection as alternatives to airborne LiDAR. Mobile mapping systems typically capture high-density point clouds coupled with high-resolution imagery. For example, Weinmann et al. [52] separated high-density point clouds from mobile laser scanning (MLS) in tree or non-tree points based on eigenvalue features and a random forest classifier, followed by a 2D mean shift segmentation to separate individual trees. Other works used region growing [53–56], clustering [54,57,58], or graph-based segmentation [59,60]. In [61], starting with an already filtered point cloud, lower tree points were outlined into individual trunks by graph-based segmentation. Then, the rest of the tree points were clustering to each closest trunk using a weighted distance combining spatial spacing and local point densities.

Such methods often comprise complex pipelines with parameters tuned for each specific dataset and rely on the high point density inherent to MLS, including pronounced tree stems as seed points. Therefore, their transferability to topographic ALS data is limited. Furthermore, as a surveying method, MLS can only cover trees seen from street level, and its data impose distinct prerequisites compared to airborne data such as ALS.

In this work, we focused on ALS point clouds, which offer higher coverage, but generally much lower point densities than those from mobile laser scanning. Moreover, a streamlined end-to-end deep learning pipeline is straightforward to fine-tune for new datasets just by labeling a new training set once a general set of hyperparameters is found.

2.3. Deep Learning for Individual Tree Detection

In recent years, deep learning has become the state-of-the-art in computer-vision-related tasks and was also applied to individual tree detection. Xie et al. [62] first identified tree proposals in a normalized height model by optimizing a tree-shape approximator in an urban scenario and then filtered out non-tree objects by a sliding window convolutional neural network (CNN). Similarly, Li et al. [63] applied a sliding window CNN to QuickBird satellite imagery for palm detection. In [64], palm detection was viewed as a binary semantic segmentation problem, where all pixels within a 2 m radius around a tree center were considered as tree points. Individual trees were then identified during inference by local maximum search in the probability map predicted by a U-Net.

Several well-established deep learning algorithms and their variants exist for general 2D object detection [65–68]. Applied to individual tree crown detection, Weinstein et al. [69] used a RetinaNet on multispectral imagery (red, green, and blue (RGB)) of an open-canopy site with semi-supervised learning and extended this in [70] to cross-site transfer learning in regards to a range of forest types. Culman et al. [71] used the same approach for palm detection. On the other hand, the authors of [72] trained a Faster-RCNN on 2D density projections of very high resolution ALS for pine forest inventorying. Plesoianu et al. [73] used a single-shot detection network to find individual trees in orchards, wooded areas, and a city site and experimented with varying combinations of spectral and height-derived input features. On the side of mobile mapping, Branson et al. [74] combined Faster-RCNN detections from both street-level and aerial RGB imagery, while Lumnitz et al. [75] applied Mask-RCNN to street-level images only. In contrast to these works, our network takes as the input an ALS point cloud, using the 3D information directly and obviating the need to find the optimal birds-eye-view rendering or multi-view combination and re-projection.

2.4. Three-Dimensional Deep Learning

The last four years saw a rapid growth in the number of papers concerning deep learning for point cloud classification. Prevalent examples are PointNet++ [76], sparse convolutional networks [77–79], PointCNN [80], and KPConv [81], just to name a few. Such neural networks proved to be accurate and fast tools for semantic segmentation of large ALS point clouds [82–86]. Yet, since deep learning tends to require large amounts of training data, generating semantic point labels to train such a deep neural network is a tedious task. Alternatively, it may be easier to label relatively few ground truth tree objects in individual bounding boxes. Moreover, object-centered detection networks yield individual entities including important attributes such as positions and sizes.

Deep learning architectures for 3D object detection are mostly found in the field of autonomous driving [87,88] and indoor scene understanding [89]. Broadly speaking, such approaches can be divided into three groups: (a) 2D networks working on projections of point clouds [90–92]; (b) networks with 2D backbones and a learned projection from a small 3D sub-network at the beginning of the pipeline [93]; (c) pure 3D networks. In the latter case, some works used dense 3D convolution [94,95], which is generally expensive to compute. Several other papers utilized PointNet-style backbones [96–99] or sparse convolutional backbones [78,100–103], which is a more efficient approach than the use of PointNet [78,101]. In contrast to cars, bicycles, and pedestrians, individual trees may be more challenging to detect in point clouds due to the overlap of bounding boxes, strongly varying sizes, and ambiguous distinction among individuals and towards other vegetation such as bushes.

In this work, detection and semantic segmentation heads were placed on top of a 3D sparse convolutional backbone [104] in the U-Net shape [105], since this type of architecture previously achieved good results in regards to ALS point cloud classification and allows for direct 2D ablation studies.

2.5. Ground Truth Data

In contrast to semantic point cloud segmentation, reference data for individual tree detection are much more scarce, let alone benchmark datasets. Therefore, it is common for research papers to use proprietary datasets. Reference labels are often obtained manually through visual interpretation of remote sensing data, while in situ field surveys are less frequent [18]. An alternative is existing municipal tree inventories, some of which are open data [106]. However, these typically only contain trees along streets and/or on public grounds [8,11]. This is less of a problem if the detection focuses on street-level data acquisition [74]. Other works complemented existing inventories by further surveys [45].

Pre-training is a standard technique in deep learning to compensate for small domain-specific training data. In general object detection, networks are often trained on classification tasks with large datasets before learning the actual detection objective [66,67]. The lack of large appropriate training data for tree detection was addressed in [69,71] by first pre-training on a large number of noisy predictions from an unsupervised detection method and then fine-tuning on a much smaller set of manually labeled, higher quality ground truth trees.

In terms of benchmarks, there had been timely limited comparisons of different methods on shared datasets in the past [20,22,23]. However, only recently, efforts were made for a curated, open, and long-term benchmark for individual tree detection in forests [107].

In this work, we generated the three-dimensional ground truth by manually labeling tree cylinders directly inside the ALS point cloud, with a municipal tree inventory as a starting point, assisted by automatically derived weak labels for pre-training. This ensured complete coverage, evaded temporal mismatch, and was more efficient compared to field surveys.

3. Materials and Methods

3.1. Methodology

For identifying individual trees in ALS point clouds of urban environments, we propose the use of 3D convolutional networks. In this section, the architecture, training regime, and inference process of the neural network, as well as the refinement of object detections by predictions from semantic point cloud segmentation are presented.

3.1.1. Architecture

The architecture employed in this work takes a 3D point cloud as the input and outputs semantic point labels, as well as object-based detections. For clarity, we divided the architecture into three main sub-networks: a sparse 3D backbone for feature extraction, a semantic segmentation head for pointwise predictions, and a detection head to output individual tree objects (Figure 1).

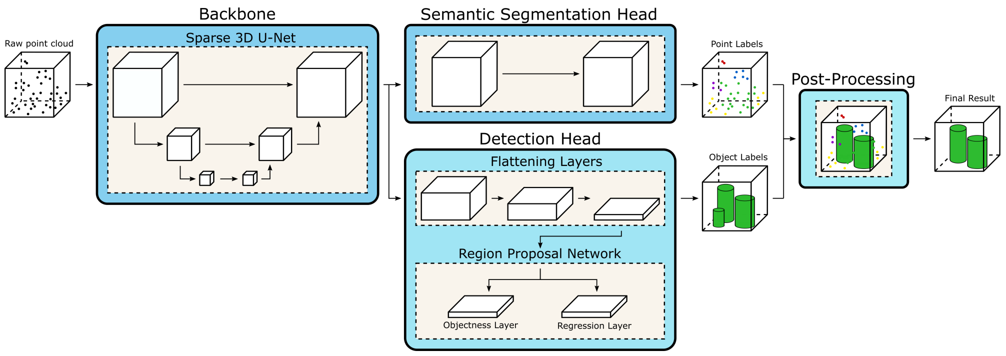

Figure 1. Overall pipeline including the network architecture scheme. Note that we used two independently trained instances of the backbone; see Section 3.1.2 for details.

As the backbone, we used a U-Net-style [105] 3D submanifold sparse convolutional network (SCN) [104] for three-dimensional feature extraction. Submanifold sparse convolution is a technique conceptually similar to standard convolution, but utilizes a general matrix multiply formulation of discrete convolution to leave out certain parts of the input feature maps. This is particularly useful to build convolutional neural networks for sparse 3D point cloud data. SCN architectures proved to be successful for ALS point cloud classification [82,86]. Furthermore, in contrast to more specialized point cloud architectures such as PointNet++ or KPConv, this kind of network allows us to conduct ablations with a direct 2D CNN counterpart (see Sections 4.3 and 4.4.6).

The backbone was eighteen convolutional layers deep in four resolution levels, with ReLU non-linearities and batch-normalization [108]. As the input, the SCN took voxelized point cloud tiles, which comply with a discrete, but sparser form of the point cloud. Per-voxel features (number of returns, return number, intensity, RGB, and height above the ground) were averaged from all respective inlying points. We chose a sample tile size of $64 \times 64 \times 128$ m and a voxel size of 0.5 m per side. Hence, the network's input shape was $128 \times 128 \times 256$ voxels. The larger size in (relative) height ensured that tiles covered the entire vertical data spread at their respective location, therefore obviating vertical tiling. The output of the backbone sub-network was 32 learned features per voxel, which were then piped further to a semantic segmentation head or a detection head.

For the pointwise tree classification, the 32 feature maps were transformed to binary pseudo-probability outputs for each voxel and then mapped back to the point cloud. The semantic segmentation head consisted of two $1 \times 1 \times 1$ convolutional layers with dropout ($p = 0.5$) in between and softmax at the end. Weighted cross-entropy loss and stochastic gradient descent were used to train the segmentation network. Since this paper focused on individual object detection, the reader is referred to [82] for further details on the semantic segmentation of ALS point clouds with SCNs.

In the detection head, inspired by [78,95], three sparse convolutional layers flattened the three-dimensional activations down to two dimensions between the purely 3D backbone and the region proposal network. We achieved this by setting both the stride and filter size in the z direction to eight for the first two layers. For the third layer, the vertical direction was convolved with a stride and size of four, so that only one channel was left in

z. Removing the height dimension and thereby vertical redundancy narrowed the search space and sped up the computations later on. The final flattening layer was also strided by 2 for x and y, which further quartered the number of possible detections to reduce the computational complexity later on. Thereby, the flattened shape was $64 \times 64 \times 1 \times (3+1) \cdot K$, where the last dimension represents the convolutional filter channels and K is the number of default anchor shapes in the region proposal network. We used ReLU activation functions and batch normalization after each convolutional layer and a dropout layer at the end ($p = 0.5$).

To finally output individual trees, we adapted a single shot detection style sub-network [65]. All layers in this region proposal network were standard—i.e., dense—2D layers. Since the point cloud's horizontal projection was mostly dense in our case, there was no computational benefit to be gained from using sparse convolution at this stage of the pipeline. Two separate branches of 1×1 convolutional layers processed the flattened activations in parallel: an objectness layer and a regression layer. The objectness layer predicted a detection score x_i for each of the K anchor shapes at each of the 64×64 anchor points. The regression layer likewise output 3 offsets $\hat{\Delta}_x$, $\hat{\Delta}_y$, $\hat{\Delta}_r$, for each anchor.

In contrast to standard bounding boxes, we used a circular object definition to better fit the typical tree shape. We followed the usual differential encoding principle [109], however adapted to circles instead of rectangular bounding boxes. The network's actual regression output were offsets $\hat{\Delta}$ to the so-called *anchors* or *default boxes/circles*. Compared to regressing parameters directly, this allowed aiding the network by pre-defining default anchor sizes. Furthermore, the exact position of a detection was refined by regressing offsets to the spatial location of the object in the output activation map.

$K = 6$ default anchor shapes were used, defined by radius $r^a = \{2, 3, 5, 8, 10, 12\}$ m. Anchor position x^a and y^a were the location in the output activation map. The actual coordinate and size of a detection in the local sample reference system was then:

$$x = x^a + \hat{\Delta}_x \cdot r^a \tag{1a}$$
$$y = y^a + \hat{\Delta}_y \cdot r^a \tag{1b}$$
$$r = \exp(\hat{\Delta}_r) \cdot r^a \tag{1c}$$

Inversely, when deriving the offset of a ground truth cylinder g to a default anchor for the regression loss during training, we computed:

$$\Delta_x = (x^g - x^a)/r^a \tag{2a}$$
$$\Delta_y = (y^g - y^a)/r^a \tag{2b}$$
$$\Delta_r = log(r^g/r^a) \tag{2c}$$

3.1.2. Training

Training data were augmented by first rotating in 30° increments and tiling samples with an overlap of 33%. Additionally, samples were randomly scaled in the range $[0.9, 1.1]$, translated up to 10 px, and randomly flipped in the x and y direction. On top of that, the ground truth was further scaled by a factor between 0.98 and 1.02 and translated up to 2 px relative to the sample data. Coordinates per tile were reduced, so that $\min(x) = \min(y) = \min(z) = 0$. Stochastic gradient descent was employed to train the detection network with an initial learning rate of 0.1, which decreased by a factor of 0.7 every 3 non-improving epochs. The weight decay was set to 10^{-5} and momentum to 0.9. Training was stopped after 8 epochs of the stagnating of the validation's mean average precision (mAP).

Before calculating the detection loss at each training iteration, one must first determine which of the $64 \cdot 64 \cdot K$ detections are to be considered true positives or false positives, more commonly known in the object detection literature as positives and negatives. For each ground truth tree, all detections with an intersection over union (IoU) higher than 0.6 were identified as matches and thereby positives. To assist training in early stages, for each ground truth, the detection with the highest IoU overlap was also counted as a match. From all other detections, those that did not have any IoU to the ground truth higher than 0.4 were selected as negative candidates. As most of the hyperparameters in this work, these values were empirically determined by an extensive hyperparameter search, but were similar to comparable works [66,78].

We used a combined object detection loss similar to works as [65,66]. The overall loss (per tile sample) is the sum of mean objectness loss L_{obj} and mean regression loss L_{reg}:

$$L = \frac{1}{N_{pos} + N_{neg}} \cdot L_{obj} + \frac{1}{N_{pos}} \cdot L_{reg} \tag{3}$$

with N_{pos} being the number of positive detections and N_{neg} the number of negative detections. For the objectness loss, we opted for the binary cross-entropy. Let x_i be the output of the objectness layer for a detection i (at position x^a, y^a, and for anchor k), then the loss is defined as:

$$L_{obj} = -\, p \sum_{i \in Pos}^{N_{pos}} y_i \cdot \log(\sigma(x_i)) - \sum_{i \in Neg}^{N_{neg}} (1 - y_i) \cdot \log(1 - \sigma(x_i)) \tag{4}$$

where σ is the sigmoid function to scale the objectness value to a score between 0 and 1. The ground truth objectness was $y_i = 1$ for positive and $y_i = 0$ for negative candidates. We balanced the influence of positive and negative detections on the loss by specifying $p = N_{neg}/N_{pos}$, which was set to 0 for $N_{pos} = 0$. For regression, smooth L1 loss [109] was used on all positive detections and averaged over the regression objectives. For $\hat{\mathbf{\Delta}}_\mathbf{i}$ being the predicted regression offset vector for detection i and $\mathbf{\Delta}_\mathbf{i}$ being the offset of the matching ground truth to the detections default anchor $k(i)$, the regression losses were:

$$L_{reg} = \frac{1}{[j]} \sum_{i \in Pos}^{N_{pos}} \sum_{j \in \{x,y,r\}} \beta_j \cdot L1_{smooth}(\hat{\Delta}_{i,j}, \Delta_{i,j}) \tag{5}$$

$$L1_{smooth} = \begin{cases} 0.5 \cdot (\hat{\Delta} - \Delta)^2 & \text{if } |\hat{\Delta} - \Delta| < 1 \\ |\hat{\Delta} - \Delta| - 0.5 & \text{otherwise} \end{cases} \tag{6}$$

Instead of balancing objectness and overall regression loss, we weighted each regression parameter separately. We set $\beta_r = 16$ and the rest to 1. Insights about this choice of parameter are given in Section 4.4.2.

Following [69], the object detection network was first pre-trained on a larger set of weak labels generated by the nDSM-based baseline A (see Section 3.2.2). This network state was then used as initialization for fine-tuning on the core training set with higher-quality labels.

For the simplicity of the pipeline and to circumvent the need to balance the two tasks, the networks for semantic segmentation and object detection were trained separately, i.e., they shared the same backbone architecture, but did not share their weights. Thereby, either part could be replaced without affecting the other one.

3.1.3. Inference

At inference, we predicted detections for each 64×64 m tile of the test set individually, again with 33% overlap. Non-maximum suppression (NMS) was applied with a maximum allowed IoU overlap of 0.3 and a minimum objectness score of 0.1. Then, all detections were transformed into the original global coordinate system and merged together. A final NMS removed overlapping trees at former tile borders. Further post-processing (which was also performed for all baselines) included the deletion of lower-scoring tree detections that were fully inside another detection, even if their IoU was below the aforementioned threshold. Height and z were drawn directly from the maximum nDSM or mean digital terrain model (DTM) value in a half-radius area.

3.1.4. Region Growing

Because our methodology can output semantic pointwise labels, as well as abstract object detections, we combined these two predictions for a better fit of the object approximations to the ground truth. In the post-processing step, the global predicted tree point cloud was intersected with the object detection result by assigning all inlying tree points to each cylinder. Where multiple trees overlapped, points were associated with the closest tree. Then, starting from the tree edges as seed points, a 2D region growing, similar to [26], expanded the tree objects iteratively along the previously unassigned tree points. Empirically, growth was restricted to points that: (a) were beneath the respective tree height; (b) were higher than 0.4-times the tree height and at least 2 m high; (c) were at a maximum 20 cm higher than the parent seed; (d) were not further away from tree center more than 1.5-times the initially predicted radius, as well as (e) not further away than half the tree height. Finally, tree detections with radii smaller than 0.5 m were deleted, which occurred if too few tree points were inside a tree cylinder.

3.2. Data

Since deep learning is mainly a data-driven methodology, extra emphasis should be given to the (ground truth) materials. In this section, first, the study area and raw data materials used in the present work are described. Subsequently, the process of ground truth generation is elaborated.

3.2.1. Study Area

In this work, a section of the city center of Stuttgart, Germany, was chosen as the study area for individual tree detection. In addition to an ALS point cloud, other available data included an aerial RGB orthophoto and a municipal tree cadaster. The selected area was surveyed by ALS in 2016, expanded over 2×2 km, and was divided into three subsets for training, validation, and testing (Figure 2a). These were outlined so that each subset depicts all parts of the diverse inner-city scene, from large commercial buildings to residential areas with gardens on the outer hillsides. Multiple parks were present among all three subsets. Relevant point attributes were intensity, number of returns, and return number, with up to 7 returns. The average first return point density in the study area was 16.6 $^{\text{pts}}/_{\text{m}^2}$. The semantic classes provided by the dataset were *ground*, *building*, *low points*, *bridges*, and *vegetation including other non-ground points*. The true orthophoto with a 20 cm ground sampling distance (GSD) was used to colorize the point cloud for additional spectral features. It was taken in the early leaf-on period in 2017.

A municipal tree cadaster is a database in which only trees on public land are recorded, mainly in parks or along streets. The purpose of such inventories is mainly to ensure compliance with safety regulations [110] and general maintenance [8,11]. The present inventory had 207,248 entries for the entire city area, with the most recent entry dating back to 2014. Dedicated forest areas were excluded. Of these trees, 6150 were located in the core study area. However, besides horizontal coordinates, neither height, canopy width, nor similar size descriptors were noted, which are essential parameters in object-based tree recognition methods such as neural detection networks. Since this tree cadaster was spatially incomplete and lacked the necessary information to train and evaluate our network, it was manually enhanced, as will be described in the following section.

3.2.2. Ground Truth Generation

The basis for our labeling campaign was a modified version of the tool by [111], by which 3D cylinders can be tightly fit into the point cloud around all points comprising each tree. Since the dataset was too large to work efficiently in the labeling tool, we divided the whole dataset into 100 individual jobs, each 200×200 m wide and extending 10 m into their neighbors, thereby creating 20 m overlap regions.

Each job was initialized with the entries from the tree cadaster. Since it only contained the horizontal positions, z was taken from a derived DTM and the height from the nDSM.

Their initial radius was set to 5 m. After editing the raw initial object labels and deleting obsolete ones, further trees were added to the ground truth by the operators to complete the scene. Due to the point cloud resolution, large shrubs were very difficult to differentiate from trees and therefore also included in the ground truth (see Section 4.4.4 for the size distribution of the ground truth).

In a second step, the results were controlled by another operator. Ambiguous cases, such as tree clusters, were checked with the help of additional data sources, in particular Google Street View and the 3D Mesh from Google Maps, taking into account the difference in the time of capture. Tree clusters were separated if visible bottom trunks were clearly apart or if they had distinctive crowns, either by shape or color. Finally, additional in situ controls were performed for the test set.

The root-mean-squared error (RSME) of 2710 merged duplicates from within the overlaps of adjacent job tiles can be used as an indication of the labeling quality, 37 cm for x and y, 44 cm in the z direction, 59 cm for the radius, and 60 cm in height. The mean IoU between matching duplicates was 0.78. In total, we obtained 10,864 final ground truth cylinders, the vast majority of which were deciduous trees. Of these, 3981 fell into the training set; furthermore, 1733 and 5150 trees were in the validation and testing sets, respectively.

In order to also train semantic segmentation networks for binary tree classification, we augmented the pointwise ground truth by relabeling all points of the miscellaneous non-ground class that were inside any of the tree cylinders. To account for protruding branches, radii were enlarged by 10% + 0.5 m, and the height was given an additional margin of 1 m. Points lower than 2.5 m were disregarded to avoid cars, fences, etc. In the last step, the new point labels were furthermore manually corrected. The result is shown in Figure 2.

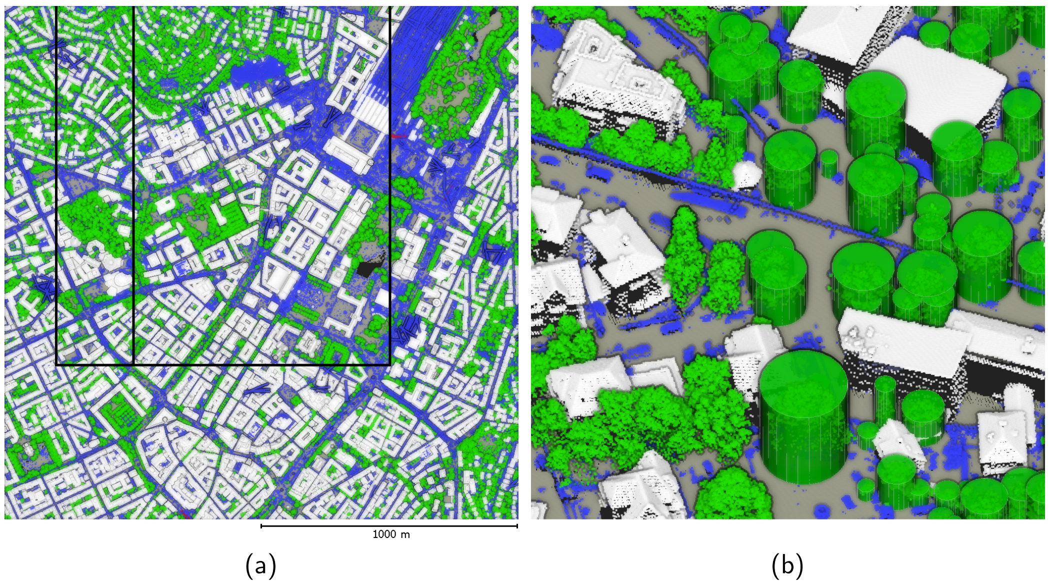

Figure 2. Ground truth of the dataset. (**a**) Overview with the three subsets visually delineated: training split in the center, the slim validation split and the u-shaped test set. (**b**) Detailed view with ground truth cylinders on the right side. Grey: *ground*; white: *building*; green: *tree*; blue: *other*.

Weak labels for pre-training were generated from an 12 km^2 space around the core study area via Baseline A (see Section 3.3). There were 10 of the 1×1 km tiles (30,101 trees) used for training and 2 tiles (7028 trees) for validation.

3.3. Baselines

Since there was, unfortunately, no benchmark for urban single tree detection, we implemented four baselines to compare our method. Hence, we chose two traditional approaches based on nDSM segmentation (plus two variants) and two more contemporary 2D object detection networks.

3.3.1. nDSM Segmentation

Individual tree delineation with unsupervised segmentation of the canopy height model is a standard technique in forestry. In urban areas, this becomes a two-step process, since the CHM first has to be derived from the nDSM by masking out non-tree areas. This binary classification is a major accuracy bottleneck of this general approach. The 2D projections and height models that were input to the baselines were computed with a raster resolution of 50 cm.

The first baseline was based on the watershed segmentation of the CHM. The binary tree mask was derived from the ALS point cloud by thresholding at a minimum of three returns. Only areas higher than 2 m above the ground were considered. After that, morphological operations were used to filter out outliers and to close obvious gaps in tree borders. Variant *A-BM* used the building class from the point cloud's pre-classification to mask out all buildings to simulate the availability of a building cadaster. In a third variant (Baseline *A-TM*), we derived the tree mask via semantic point cloud segmentation with SCN in order to replace the hand-crafted rule set with a learned classification. For this, the predicted class of the highest point per pixel was projected into the horizontal plane. For individual tree delineation, the nDSM was firstly smoothed by Gaussian filtering and then further refined by morphological filters to remove small objects and avoid over-segmentation. Finally, a watershed transform [112] segmented the inverted CHM to the final tree delineations.

The second baseline was based on the tree segmentation algorithm by [26] and included additional multi-spectral data sources. Instead of pixelwise classification, we used object-based image analysis (OBIA) combined with an elaborate rule set separated tree from non-tree areas. The basic inputs were pansharpened Pléiades satellite imagery (RGB + near infrared (NIR), taken on a clear sky day in Mai 2017) together with nDSM and the number of returns from ALS. From this, 12 spectral indices (NDVI, renormalized difference vegetation index (RDVI), normalized difference soil index (NDSI), etc.) and two geometrical nDSM features (slope [113] and standard deviation) were computed. In the first step of the baseline, buildings were excluded through the ALS pre-classification. Secondly, all pixels higher than 1 m above the ground were split into segments by a multiresolution segmentation. A decision tree then split tree from non-tree segments by utilizing all the previously mentioned features. Small bushes were removed by only keeping segments that were at least partially above 2.5 m. Geometrical cleaning was needed to remove small, undesired objects such as street lamps, traffic signals, vehicles, and powerlines. At the same time, special rules were set to avoid pines' exclusion. The single tree delineation followed the process described in [26], using the implementation by [114] (https://github.com/r-lidar/lidR, accessed on 27 February 2022): First, treetops were searched by local maxima filters with a variable window size depending on the nDSM (in the general assumption that the horizontal size of a tree correlates with its height). The nDSM was then smoothed by a Gaussian filter and masked by the tree filter from the OBIA classification to obtain the CHM. The treetops were finally used as seeds for a region growing on the CHM to obtain individual tree segments.

The irregularly shaped tree segments from Baselines A and B were further converted to fit the ground truth's approximation defined by the crown radius. For each segment, we first computed the centroid of all segment pixel coordinates as the horizontal tree position. Then, all border pixels i were selected, and a circle was fit by estimating the tree crown radius:

$$\hat{r} = \bar{r} + \eta \cdot (\max(r_i) - \bar{r}) \tag{7}$$

with $\eta \in [0,1]$ as a parameter to balance between the average and maximum radius to segment the center. We performed a grid search over η for all baselines. For Baselines *A* and *A-BM*, it was found that $\eta = 0.4$ yielded the best results, but for *A-TM*, the optimum was at $\eta = 0.7$. For Baseline *B*, we set $\eta = 0.2$.

3.3.2. Two-Dimensional Object Detection Networks

More state-of-the-art baselines are 2D deep object detection networks. Here, we used *YOLOv2* [67] and *Faster-RCNN* [66] as examples for "off-the-shelf" detection networks. We chose ResNet-50 [115] pre-trained on ImageNet [116] as the backbone in both cases. The first convolutional layer was changed to accept the input with 128 × 128 pix and six input channels: RGB, nDSM, intensity, and number of returns. For YOLOv2, we placed the detection head on top of the seventh residual block (after twenty-two convolutional layers). A later place in the backbone network would increase the depth, but decrease the output resolution of the anchor locations. For example, we found that choosing the 10th or 40th layer as the feature extraction layer lowered the accuracy by about 10 points of the mAP. For Faster-RCNN, we kept the default feature pyramid scheme [117] from the TorchVision implementation. As for our main SCN network, input samples were 64 × 64 m with 33% overlap and 50 cm resolution. Further similarities were in the data augmentation, training, and inference strategy, as well as pre-training on the weak label set from the extended study area.

4. Results

In this section, we compare the accuracy of our method (*SCN-OD*) on the test set to the baselines from the previous section through a set of evaluation metrics both on the object level, as well as the pixel level. To compensate for statistical variations, all results from the neural networks were averaged over ten instances, independently trained with different random seeds. The outputs of the object detection networks came with predicted objectness scores. In application, one can balance between precision and recall by choosing a score threshold that defines which detections are to be considered. Therefore, in addition to reporting threshold-independent average precision, when comparing to the non-deep-learning baselines, we generally chose score thresholds so that the precision equaled the recall in individual tree detection. Such a prediction was then also refined by post-processing through region growing (*SCN-OD-RG*). Exemplary results are shown in Figure 3.

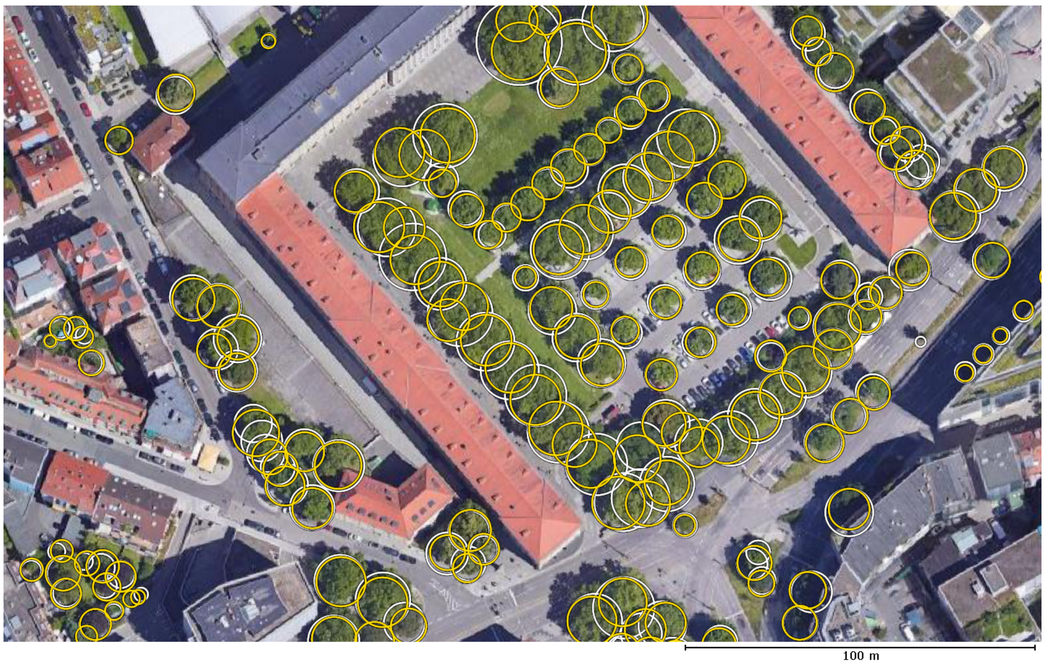

Figure 3. Exemplary detections from *SCN-OD-RG* in the test set. Yellow: predictions; white: ground truth.

4.1. Individual Tree Detection

First, individual tree detection was evaluated in terms of pairwise 2D circular IoU overlap to the ground truth trees, accepting detections as true positives in regard to five different IoU thresholds. We computed continuous average precision (AP) from the interpolated precision–recall curve $P(R)$ according to the Pascal VOC 2012 devkit, where further detections of the same ground truth were counted as false positives.

The results are shown in Tables 1 and 2. Overall, our *SCN-OD* outperformed all baselines in all metrics. For IoU > 0.3, *A-TM* had 0.5 higher precision, but much lower recall than *SCN-OD*, although the precision of *SCN-OD* can still be improved by adjusting the prediction threshold (Figure 4). Within the CHM baselines, *A-TM* performed the best, except for the stricter IoU thresholds, where Baseline *B* showed higher recall. *YOLOv2* had similar precision as the best nDSM baseline, but higher recall, while *Faster-RCNN* was closer to *SCN-OD*. *SCN-OD* achieved an average precision of 82.9% and P/R $= 80.6$ on the widely used IoU > 0.5 acceptance threshold.

After combining the raw object detection results from *SCN-OD* with the pointwise semantic predictions from *SCN-SemSeg* through region growing, *SCN-OD-RG* showed even higher precision, but a the cost of slightly lower recall. The reason was the removal of detections without pointwise tree predictions inside. Only for the high shape prerequisite IoU > 0.7, the recall increased also.

Table 1. Results for individual tree detection, measured by the average precision (AP) for different intersection of union thresholds in %.

	mAP	**AP@.3**	**AP@.4**	**AP@.5**	**AP@.6**	**AP@.7**
YOLOv2	66.0	87.7	82.5	72.9	55.2	31.3
Faster-RCNN	72.3	86.6	84.7	79.3	66.6	43.7
SCN-OD	**76.0**	**91.0**	**88.8**	**82.9**	**70.0**	**47.5**

Table 2. Results for individual tree detection, measured by the precision and recall for different IoU thresholds in %.

	P/R@.3	**P/R@.4**	**P/R@.5**	**P/R@.6**	**P/R@.7**
A	79.6/66.0	73.6/61.0	63.0/52.2	50.5/41.9	34.6/28.7
A-BM	83.7/65.3	77.2/60.2	65.9/51.5	52.6/41.1	35.9/28.0
A-TM	85.8/65.0	79.8/60.4	69.5/52.6	57.4/43.5	42.2/31.9
B	54.8/70.1	50.0/63.9	42.3/54.1	32.2/41.2	21.5/27.5
YOLOv2	82.0	79.1	73.3	61.1	42.7
Faster-RCNN	84.3	83.0	79.2	70.1	53.4
SCN-OD	**85.3**	**84.2**	**80.6**	**72.5**	56.2
SCN-OD-RG	**87.2**/85.1	**85.9**/83.8	**82.3**/80.3	**74.1**/72.3	**58.2**/**56.8**

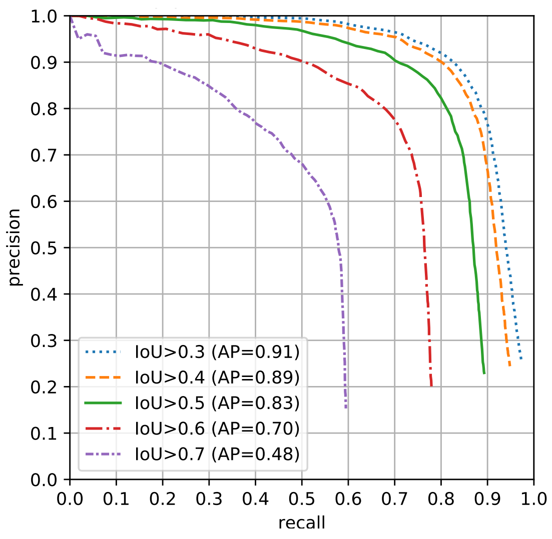

Figure 4. Precision-recall curves of the average SCN-OD network regarding individual tree detection for different intersection of union thresholds.

4.2. Stem Precision and Recall

In addition to measuring the similarity by pairwise surface parity, it is common to count the recall as the relative number of detection centers inside ground truth trees and the precision as the relative number of ground truth tree centers covered by a detection. In each case, we counted a tree position (both GT and prediction) only once as the true positive. Besides the full tree radius [20,69], we also considered 3 m as the maximum distance between the ground truth and detected trees [20,63,64]. Table 3 shows a similar pattern as for IoU matching at a threshold of 0.3. Again, the overall best performance was achieved by *SCN-OD* with further improvement in precision by *SCN-OD-RG*. Only *A-TM* managed higher precision in exchange for severely lower recall.

Table 3. Stem precision and recall in %.

	Dist < Tree Radius		**Dist < 3 m**	
	AP	**P/R**	**AP**	**P/R**
A	-	83.2/68.8	-	76.6/63.5
A-BM	-	87.3/68.0	-	80.3/62.7
A-TM	-	**89.5**/67.9	-	83.4/63.2
B	-	56.4/73.2	-	51.4/65.7
YOLOv2	89.5	82.8	86.3	80.8
Faster-RCNN	87.3	84.7	85.5	83.5
SCN-OD	**92.0**	**85.7**	**89.8**	**84.5**
SCN-OD-RG	-	87.7/85.4	-	**86.4**/84.3

4.3. Tree Canopy Cover

Next, we analyzed how well the different methods captured tree canopy cover. For this, we converted the segmentation results of the nDSM baselines to a pixelwise tree/non-tree classification. Furthermore, we trained the SCN for semantic point cloud segmentation on binary tree classification and projected the prediction of the highest point into the horizontal plane. Likewise, a *2D U-Net* was trained with the same architecture as its 3D SCN counterpart, except with standard dense two-dimensional convolution with the same input channels as for the 2D object detection networks. We also employed a pointwise random forest (*RF*) classifier taking into account a multi-scale feature set as described in [118]. For the object detection networks, predicted tree cylinders were intersected with the point cloud, whereby all points lying in cylinders were labeled as tree points. The classification of the highest point per cell was projected into the pixelwise prediction. These binary classifications were compared with a semantic 2D ground truth that was equally derived from the pointwise ground truth. The results are shown in Table 4.

Table 4. Tree canopy cover accuracy in %.

	Tree			**Non-Tree**			
	Prec.	**Recall**	**F1**	**Prec.**	**Recall**	**F1**	**OA**
A	86.2	84.2	85.2	97.5	97.8	97.7	95.9
A-BM	87.9	82.5	85.0	97.2	98.2	97.7	96.0
A-TM	**96.4**	80.8	87.9	97.0	**99.5**	98.3	96.9
B	80.7	88.7	84.5	98.2	96.6	97.4	95.5
2D U-Net	90.1	**98.7**	94.2	**99.8**	98.3	99.0	98.3
3D RF	95.2	95.5	95.2	99.3	99.3	99.3	98.8
SCN-SemSeg	95.8	**98.7**	**97.2**	**99.8**	99.3	**99.5**	**99.2**
YOLOv2	94.0	89.3	91.6	98.3	99.1	98.7	97.7
Faster-RCNN	94.9	89.5	92.1	98.3	99.2	98.8	97.9
SCN-OD	95.7	87.9	91.6	98.1	99.4	98.7	97.8
SCN-OD-RG	96.3	88.2	92.1	98.1	**99.5**	98.8	97.9

Unsurprisingly, the overall best results were achieved by the semantic segmentation methods, whereby the 3D network was better still than the 2D version and closely followed by random forest. With F1 scores over 97%, we reached the limits of ground truth quality [86]. Similar results can be expected from other point cloud classification networks [84,86,119].

The nDSM-based baselines showed comparatively low tree precision and recall, except for *A-TM* , with the highest tree precision (which used the SCN prediction as a basis for the tree mask, refined by morphological filtering), however to the cost of the lowest tree recall. Inversely, Baseline *B* had the lowest tree precision, but a tree recall comparable to the object detection networks.

The object detection networks reached similar performance in tree precision (even higher than *2D U-Net*). However, this quantitative assessment is misleading, because the quality of the error was less for the detection networks since they tended to include building points protruding into the predicted tree cylinders, while the segmentation networks tended to include shrubs in the tree class. Furthermore, their tree recall suffered from the tree cylinder approximation, which was sometimes too small to include all tree points.

Differences in the non-tree class were smaller compared to the tree class due to the seven-times higher class frequency in the test scene. Still, the classification networks performed best, followed by the detection networks and then the traditional baselines.

4.4. Ablations and Things That Did Not Work

In this section, we removed parts of our method in order to gain insights into their individual effects on the accuracy. Furthermore, we describe techniques we tried to further improve the performance, but turned out to have no or even negative effects, so that future research can focus on other ideas. A summary of all ablations is given in Table 5 (from left to right).

4.4.1. Pre-Training

Using an initial network state pre-trained on the larger set of weak labels from the same domain as the main data and then fine-tuning on higher quality training data generally improved accuracy, as shown in Table 5. In the basic setting, the accuracy improved by +7.9 points of the mAP, but only a small amount from 75.6% to 76.0% mAP for the final result. Good quality training data for fine-tuning are still important however, since the network purely trained on weak labels achieved an mAP of only 50.9% on the test set. For

YOLOv2, pre-training followed by fine-tuning lifted the mAP from 49.9% to 65.9% and for Faster-RCNN from 70.4% to 72.3%.

4.4.2. Weighted Shape Regression and Loss Function

As described in Section 3.1.2, our regression loss had a weighting term β_j to increase the importance of the individual components of the regression loss rather than weighting regression as a whole relative to the objectness loss, which would be the standard approach for this type of problem. However, we found that we needed to weight the shape regression higher than the location, which makes sense given the relatively high resolution of anchor locations and thereby the different value ranges between the regression targets. From this observation, we empirically set the hyperparameter $\beta_r = 16$, although higher values performed similarly. Starting from the basic setting, this improved the mAP from 55.8% to 67.1%. We also experimented with extending the objectness loss to a focal loss [68] with $\gamma = 1$; however, this tended to decrease accuracy. Furthermore, we experienced no gain from using the IoU to the respective ground truth as the objectness target. A grid search over other hyperparameters such as the learning rate was performed early on to achieve good training convergence (see also [19]).

Table 5. Ablation table. Values are 2D circular mAP in %.

Circular shape regression	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
RGB input features	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
3D backbone	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Weighted shape regression			✓	✓	✓	✓	✓	✓	✓	✓	✓
Improved data augmentation				✓	✓	✓	✓	✓	✓	✓	✓
No height regression						✓	✓	✓	✓	✓	✓
Pre-trained		✓			✓			✓	✓		✓
mAP	55.8	63.7	67.1	72.8	75.5	75.6	74.5	**76.0**	75.3	70.6	70.8

4.4.3. Data Augmentation

Furthermore, we found data augmentation to be an important, but relatively sensitive hyperparameter. For semantic segmentation, it had been sufficient to rotate and overlap samples to compensate for small training data. For tree detection however, it was helpful to apply further augmentation as described in Section 3.1.2. In the absence of pre-training, this additional augmentation resulted in a +5.7 mAP from 67.1% to 72.8%. When the augmentation magnitude was doubled for example, the mAP gain was only +0.5 points.

4.4.4. Shape Regression

The regression loss in Equation (5) includes only x, y, and *radius*, but not z or *height*, although three-dimensional ground truth data were available. Initially, we aimed at predicting vertical tree shape as well by adding the following terms to form a cuboid regression loss (compare to Section 3.1.2):

$$z = z^a + \hat{\Delta}_z \cdot h^a \tag{8a}$$

$$h = \exp(\hat{\Delta}_h) \cdot h^a \tag{8b}$$

$$\Delta_z = (z^g - z^a)/h^a \tag{8c}$$

$$\Delta_h = log(h^g/h^a) \tag{8d}$$

with z^a set to zero. Accordingly, additional dimensions were added to the anchors: $z = 0$ and height $= \{5, 8, 14, 22, 25, 30\}$. However, the 3D mAP was worse compared to deriving z and *height* directly from the DTM and nDSM (48.9% vs. 68.5%). Furthermore, as can be seen in Figure 5, instead of learning to predict the correct radius to height distribution, the network learned a linear radius–height relationship, even though it should have been able to learn the height from 3D convolution or a fortiori from the height above the ground input feature.

We tried two approaches to overcome this issue. Firstly, as it is common practice to weight predictions reciprocally to their ground truth class frequency, detections in the regression loss were weighted inversely to their ground truth's shape frequency expressed by a multivariate radius–height histogram. Although this gained slightly better radius coverage when using a very small anchor set, in the final configuration, the accuracy declined about −1.5% throughout. Secondly, although anchors are helpful to cover the full range of radii in the predictions, more than the finally used anchors did not improve the results significantly either, but led to slower training and inference. In the extreme case, we added enough anchors to cover the whole spectrum of three-dimensional tree shapes in 2 m increments. Thereby, the 3D mAP increased to 53.9%, but so did the computing time by a factor o six, and the 2D mAP dropped by −4.3 points. In conclusion, we removed the height and z from the loss in order to focus on the radius, which effectively improved the 2D mAP by +2.6 or +0.5 if pre-trained.

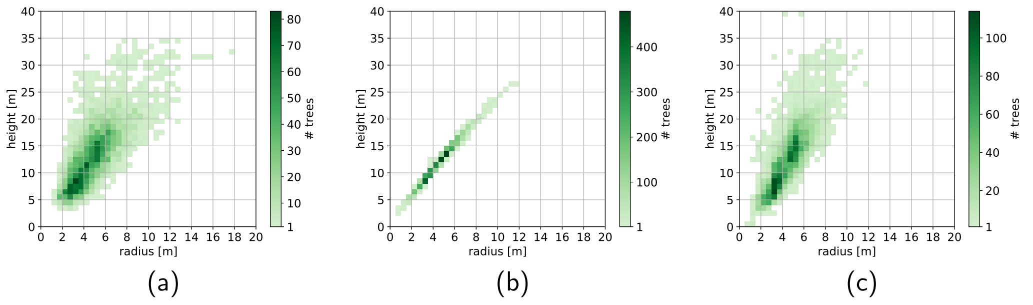

Figure 5. Three-dimensional tree size distribution in the test set. (**a**) Ground truth. (**b**) Height and z predicted by the network. (**c**) Verticality derived from the nDSM and DTM.

Typically, object detection networks work with rectangular bounding boxes. However, trees shapes are more closely approximated by circles/cylinders than rectangles/cuboids; thus, we chose to use radius as the regressed shape parameter and evaluate the circular mAP and P(R). In order to analyze its effect on accuracy, we trained an ablation on the rectangular ground truth and shape regression. After converting the output back to the radius, the (circular) mAP was −1.1 points lower (74.5%) than training on the circular ground truth. If the mAP was evaluated on rectangles, the accuracy was generally about 0.5–1 percentage points higher.

4.4.5. RGB Input Features

The point clouds were colorized with a multispectral RGB orthophoto in pre-processing. Experience had shown that the addition of optional multi-spectral point attributes was helpful, but not necessary for point cloud classification with 3D deep learning (for example, in [82]). Here, the gain from RGB on the best-performing configuration was +0.7 mAP.

4.4.6. Three-Dimensional vs. Two-Dimensional Backbone

Finally, we investigated the influence of fully 3D processing throughout the backbone of the neural network instead of projecting the ALS point cloud beforehand into 2D inputs. Just as in Section 4.3, a 2D copy of the detection network, with dense instead of sparse convolution, was trained on the tree detection task. The input channels were the projected point attributes (again from the highest point per cell): RGB, nDSM, intensity, and number of returns. Networks with 2D backbones were outperformed by their 3D counterparts by 5.1 percentage points regarding the mAP and 5.9 points for AP@0.5.

5. Discussion

Overall, our method achieved good results and outperformed the baselines in almost all metrics in terms of individual tree detection, stem detection, and tree canopy cover. Where baselines showed better precision, it was due to building upon predictions from our network and to the cost of recall. Combining object detection with semantic segmentation improved precision and slightly enhanced radius estimates. However, one problem of the region growing was cases of winding, outlying branches, which tended to cause some trees to grow too large to the detriment of their adjacent neighbors. In general, such region growing should only be seen as an optional post-processing step to refine the results if an appropriate rule set can be identified. If the goal is to estimate tree canopy cover, 3D semantic segmentation alone is the easier and faster alternative. The inference time on the test set was about 2 min for the object detector and 40 s for the pointwise classification on a single NVIDIA Titan RTX. In the ablations, we showed the positive effects of finely tuned loss weighting and data augmentation, as well as the benefits of a 3D backbone and circular tree approximation compared to the usual rectangular one. Pre-training on a larger set of weak labels showed higher gains the less refined the methodology. It should be noted that the ground truth quality was not perfect, despite the efforts taken in the process and the advantage of labeling in the original 3D point cloud compared to 2D. Labeling was limited in the resolution of the point cloud and penetration of the LiDAR beams. Individual trunks, which were the decisive criterion for differentiation if the more prevalent crowns were not distinctive enough, were not always represented by the data or were very difficult to identify. Furthermore, trees were not differentiated from larger shrubs, while very small trees may have been missed. Relying on more precise in situ measurements in this scale would have been too costly, partially impossible due to a lack of accessibility and error prone due to the time distance to the LiDAR measurement. Therefore, it is important to understand that the neural networks trained on this ground truth would not necessarily learn to correctly find individual trees or tree points, but rather to emulate the training data.

Comparison to other works was difficult due to their different datasets and evaluation procedures. In the following, we try to set our result in relation to the more recent and most comparable works. The authors of [59] counted detections as positive if within a 60% average tree distance to the nearest ground truth. Their 3D graph cut achieved a detection rate of 77.2% for MLS and 83.6% for ALS. Xie et al. [62] reported precisions of 90–95% and a recall of 80–85%; however, it is unclear how this was measured. Furthermore, this was one of the computationally most expensive approaches, requiring about 100 h for 8.75 km^2. Among the deep-learning-based methods, the sliding window approach in [63] detected palms within a 3 m maximum distance with a precision and recall of 94.8–98.7% and 96.6–98.9%, respectively. Freudenberg et al. [64] achieved in the same metric $p = 94.4\%$ and $R = 93.5\%$ with their semantic segmentation approach for individual palm detection. The RetinaNet in [71] identified palms with $p = 78.1\%$ and $R = 67.7\%$ for IoU > 0.4. These results were partly much higher than ours; however, palms have very distinctive patterns that are more easily distinguishable than the groups of deciduous trees in our dataset. In a forest dataset with an ultra-high point density, Windrim and Bryson [72] reached a precision of 100% and a recall between 67% and 93% (IoU > 0.5) with a Faster-RCNN on point density maps. Weinstein et al. [69] reported a recall of 69% and a precision of 61% for IoU > 0.5 on a open-canopy site. The recall of field-collected stems was 81%. The single-shot detector in [73] had a recall of 77% in an orchard, 72% in a wooded area, and 76% in a city dataset (IoU > 0.5). Unfortunately, no precision was given by the authors. Using street-view imagery, Lumnitz et al. [75] achieved AP@.5 values ranging from 62.0% to 68.2% with a MaskRCNN. Combined with airborne imagery, Branson at al. [74] achieved AP = 71% whilst counting true detections as positives if within a 4 m distance to a ground truth tree.

6. Conclusions

In this work, we showed the utility of a deep 3D single-shot detection network for urban tree detection in ALS point clouds. A large-scale case study demonstrated good accuracy in individual tree detection and tree canopy cover estimation while outperforming the baselines. Measured at 50% IoU overlap, the average precision regarding individual tree detection was 82.9% for the 3D network and 77.0% for its 2D version, compared to the 79.3% of Faster-RCNN. Likewise, traditional CHM segmentation methods were outperformed by 12.8 pp and 26.1 pp in the precision and recall, respectively. Insights were given about the influence of essential parts of our proposed methodology using an extensive ablation study, which we hope will help improve tree detection methods in future works.

In the future, we expect 3D neural networks for object detection in topographic ALS point clouds to benefit even more from the use of higher point densities compared to 2D detection networks in general, but especially in individual tree detection due to the better resolution of tree stems in the data [23]. Furthermore, we anticipate larger, high-quality training datasets to improve the generalization for urban tree detection including outlying tree sizes and other edge cases.

Author Contributions: Conceptualization, S.S.; data curation, S.S.; investigation, S.S.; methodology, S.S.; software, S.S. and A.N.V.; supervision, U.S.; validation, S.S.; writing—original draft, S.S. and A.N.V.; writing—review and editing, S.S., A.N.V. and U.S. All authors contributed to the text. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to show their gratitude to the State Office for Geoinformation and Land Development (Landesamt für Geoinformation und Landentwicklung (LGL)), Baden-Württemberg, for providing the ALS point clouds covering the city of Stuttgart. Furthermore, we would like to thank both the Stuttgart City Surveying Office (Stadtmessungsamt) for providing the orthophoto and Stuttgart's Office for Parks, Cemeteries and Forest (Garten-, Friedhofs- und Forstamt) for providing the municipal tree cadaster. AIRBUS Defence and Space Pléiades-1 satellite images were accessed as an ESA Third Party Mission within the ESA TPM project research id 49634.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nowak, D.J.; Crane, D.E.; Stevens, J.C. Air pollution removal by urban trees and shrubs in the United States. *Urban For. Urban Green.* **2006**, *4*, 115–123. [CrossRef]
2. Weng, Q.; Lu, D.; Schubring, J. Estimation of land surface temperature–vegetation abundance relationship for urban heat island studies. *Remote Sens. Environ.* **2004**, *89*, 467–483. [CrossRef]
3. Chen, X.L.; Zhao, H.M.; Li, P.X.; Yin, Z.Y. Remote sensing image-based analysis of the relationship between urban heat island and land use/cover changes. *Remote Sens. Environ.* **2006**, *104*, 133–146. [CrossRef]
4. Kurn, D.M.; Bretz, S.E.; Huang, B.; Akbari, H. *The Potential for Reducing Urban Air Temperatures and Energy Consumption through Vegetative Cooling*; Technical report; Lawrence Berkeley Laboratory: Berkeley, CA, USA, 1994. [CrossRef]
5. Huang, Y.J.; Akbari, H.; Taha, H. The wind-shielding and shading effects of trees on residential heating and cooling requirements. In Proceedings of the ASHRAE Winter Conference, Atlanta, GA, USA, 11–14 February 1990; pp. 11–14.
6. McPherson, E.G.; Simpson, J.R. Potential energy savings in buildings by an urban tree planting programme in California. *Urban For. Urban Green.* **2003**, *2*, 73–86. [CrossRef]
7. Pesola, L.; Cheng, X.; Sanesi, G.; Colangelo, G.; Elia, M.; Lafortezza, R. Linking above-ground biomass and biodiversity to stand development in urban forest areas: A case study in Northern Italy. *Landsc. Urban Plan.* **2017**, *157*, 90–97. [CrossRef]
8. Nielsen, A.B.; Östberg, J.; Delshammar, T. Review of urban tree inventory methods used to collect data at single-tree level. *Arboric. Urban For.* **2014**, *40*, 96–111. [CrossRef]
9. Bardekjian, A.; Kenney, A.R.M. Trends in Canada's Urban Forests. In Trees Canada—Arbres Canada and Canadian Urban Forest Network—Réseau Canadien de la Floret Urbaine, 2016. Available online: https://treecanada.ca/wp-content/uploads/2017/10/Article-1-EN-CUFN-Trends-in-Canada%E2%80%99s-Urban-Forests.pdf (accessed on 9 February 2021).
10. Hauer, R.J.; Peterson, W.D. *Municipal Tree Care and Management in the United States: A 2014 Urban & Community Forestry Census of Tree Activities*; Special Publication 16-1; College of Natural Resources, University of Wisconsin—Stevens Point: Madison, WI, USA, 2016.
11. Östberg, J.; Wiström, B.; Randrup, T.B. The state and use of municipal tree inventories in Swedish municipalities—Results from a national survey. *Urban Ecosyst.* **2018**, *21*, 467–477. [CrossRef]

12. McPherson, E.G. Structure and sustainability of Sacramento's urban forest. *J. Arboric.* **1998**, *24*, 174–190. [CrossRef]
13. Cameron, R.W.; Blanuša, T.; Taylor, J.E.; Salisbury, A.; Halstead, A.J.; Henricot, B.; Thompson, K. The domestic garden—Its contribution to urban green infrastructure. *Urban For. Urban Green.* **2012**, *11*, 129–137. [CrossRef]
14. Kelly, M. Urban trees and the green infrastructure agenda. In *Trees, People and the Built Environment, Proceedings of the Urban Trees Research Conference, 13–14 April 2011*; Johnston, M., Percival, G., Eds.; Forestry Commission: Macon, GA, USA, 2012; pp. 166–180.
15. Li, X.; Chen, W.Y.; Sanesi, G.; Lafortezza, R. Remote Sensing in Urban Forestry: Recent Applications and Future Directions. *Remote Sens.* **2019**, *11*, 1144. [CrossRef]
16. Casalegno, S.; Anderson, K.; Hancock, S.; Gaston, K.J. Improving models of urban greenspace: From vegetation surface cover to volumetric survey, using waveform laser scanning. *Methods Ecol. Evol.* **2017**, *8*, 1443–1452. [CrossRef]
17. Strîmbu, V.F.; Strîmbu, B.M. A graph-based segmentation algorithm for tree crown extraction using airborne LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 30–43. [CrossRef]
18. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [CrossRef]
19. Schmohl, S.; Kölle, M.; Frolow, R.; Soergel, U. Towards Urban Tree Recognition in Airborne Point Clouds with Deep 3D Single-Shot Detectors. In *Pattern Recognition. ICPR International Workshops and Challenges*; Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 521–535.
20. Kaartinen, H.; Hyyppä, J.; Yu, X.; Vastaranta, M.; Hyyppä, H.; Kukko, A.; Holopainen, M.; Heipke, C.; Hirschmugl, M.; Morsdorf, F.; et al. An International Comparison of Individual Tree Detection and Extraction Using Airborne Laser Scanning. *Remote Sens.* **2012**, *4*, 950–974. [CrossRef]
21. Jakubowski, M.K.; Li, W.; Guo, Q.; Kelly, M. Delineating Individual Trees from Lidar Data: A Comparison of Vector- and Raster-based Segmentation Approaches. *Remote Sens.* **2013**, *5*, 4163–4186. [CrossRef]
22. Eysn, L.; Hollaus, M.; Lindberg, E.; Berger, F.; Monnet, J.M.; Dalponte, M.; Kobal, M.; Pellegrini, M.; Lingua, E.; Mongus, D.; et al. A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space. *Forests* **2015**, *6*, 1721–1747. [CrossRef]
23. Wang, Y.; Hyyppä, J.; Liang, X.; Kaartinen, H.; Yu, X.; Lindberg, E.; Holmgren, J.; Qin, Y.; Mallet, C.; Ferraz, A.; 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. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5011–5027. [CrossRef]
24. Hyyppä, J.; Kelle, O.; Lehikoinen, M.; Inkinen, M. A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 969–975. [CrossRef]
25. Hirschmugl, M.; Ofner, M.; Raggam, J.; Schardt, M. Single tree detection in very high resolution remote sensing data. *Remote Sens. Environ.* **2007**, *110*, 533–544. [CrossRef]
26. Dalponte, M.; Coomes, D.A. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. *Methods Ecol. Evol.* **2016**, *7*, 1236–1245. [CrossRef]
27. Pyysalo, U.; Hyyppä, H. Reconstructing Tree Crowns from Laser Scanner Data for Feature Extraction. *Int. Arch. Photogramm. Remote Sens.* **2002**, *34*, 218–221.
28. Koch, B.; Heyder, U.; Weinacker, H. Detection of Individual Tree Crowns in Airborne Lidar Data. *Photogramm. Eng. Remote Sens.* **2006**, *72*, 357–363. [CrossRef]
29. Zhao, K.; Popescu, S. Hierarchical Watershed Segmentation of Canopy Height Model for Multi-Scale Forest Inventory. *ISPRS Workshop Laser Scanning* **2007**, *3*, 436–441.
30. Persson, A.; Holmgren, J.; Soderman, U. Detecting and Measuring Individual Trees Using an Airborne Laser Scanner. *Photogramm. Eng. Remote Sens.* **2002**, *68*, 925–932.
31. Reitberger, J.; Heurich, M.; Krzystek, P.; Stilla, U. Single tree detection in forest areas with high-density LIDAR data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2007**, *36*, 139–144.
32. Li, W.; Guo, Q.; Jakubowski, M.K.; Kelly, M. A New Method for Segmenting Individual Trees from the Lidar Point Cloud. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 75–84. [CrossRef]
33. Lu, X.; Guo, Q.; Li, W.; Flanagan, J. A bottom-up approach to segment individual deciduous trees using leaf-off lidar point cloud data. *ISPRS J. Photogramm. Remote Sens.* **2014**, *94*, 1–12. [CrossRef]
34. Ferraz, A.; Bretar, F.; Jacquemoud, S.; Gonçalves, G.; Pereira, L.; Tomé, M.; Soares, P. 3-D mapping of a multi-layered Mediterranean forest using ALS data. *Remote Sens. Environ.* **2012**, *121*, 210–223. [CrossRef]
35. Dai, W.; Yang, B.; Dong, Z.; Shaker, A. A new method for 3D individual tree extraction using multispectral airborne LiDAR point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 400–411. [CrossRef]
36. Xiao, W.; Zaforemska, A.; Smigaj, M.; Wang, Y.; Gaulton, R. Mean Shift Segmentation Assessment for Individual Forest Tree Delineation from Airborne Lidar Data. *Remote Sens.* **2019**, *11*, 1263. [CrossRef]
37. Pollock, R.J. Model-based approach to automatically locating tree crowns in high spatial resolution images. In *Image and Signal Processing for Remote Sensing*; Desachy, J., Ed.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 1994; Volume 2315, pp. 526–537. [CrossRef]
38. Tittmann, P.; Shafii, S.; Hartsough, B.R.; Hamann, B. Tree Detection and Delineation from LiDAR point clouds using RANSAC. In Proceedings of SilviLaser, 11th International Conference on LiDAR Applications for Assessing Forest Ecosystems, Hobart, Australia, 16–20 October 2011.

39. Lindberg, E.; Eysn, L.; Hollaus, M.; Holmgren, J.; Pfeifer, N. Delineation of Tree Crowns and Tree Species Classification From Full-Waveform Airborne Laser Scanning Data Using 3-D Ellipsoidal Clustering. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 3174–3181. [CrossRef]
40. Hadas, E.; Kölle, M.; Karpina, M.; Borkowski, A. Identification of Peach Tree Trunks from Laser Scanning Data obtained with small Unmanned Aerial System. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *2*, 735–740. [CrossRef]
41. Reitberger, J.; Schnörr, C.; Krzystek, P.; Stilla, U. 3D segmentation of single trees exploiting full waveform LIDAR data. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 561–574. [CrossRef]
42. Wolf, B.M.; Heipke, C. Automatic extraction and delineation of single trees from remote sensing data. *Mach. Vis. Appl.* **2007**, *18*, 317–330. [CrossRef]
43. Iovan, C.; Boldo, D.; Cord, M. Detection, Characterization, and Modeling Vegetation in Urban Areas From High-Resolution Aerial Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2008**, *1*, 206–213. [CrossRef]
44. Yang, L.; Wu, X.; Praun, E.; Ma, X. Tree Detection from Aerial Imagery. In GIS '09, Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009; ACM: New York, NY, USA, 2009; pp. 131–137. [CrossRef]
45. Zhang, C.; Zhou, Y.; Qiu, F. Individual Tree Segmentation from LiDAR Point Clouds for Urban Forest Inventory. *Remote Sens.* **2015**, *7*, 7892–7913. [CrossRef]
46. Bulatov, D.; Wayand, I.; Schilling, H. Automatic Tree-Crown Detection in Challenging Scenarios. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 575–582. [CrossRef]
47. Liew, S.C.; Huang, X.; Lin, E.S.; Shi, C.; Yee, A.T.K.; Tandon, A. Integration of Tree Database Derived from Satellite Imagery and LiDAR Point Cloud Data. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII-4/W10*, 105–111. [CrossRef]
48. Man, Q.; Dong, P.; Yang, X.; Wu, Q.; Han, R. Automatic Extraction of Grasses and Individual Trees in Urban Areas Based on Airborne Hyperspectral and LiDAR Data. *Remote Sens.* **2020**, *12*, 2725. [CrossRef]
49. Persson, A. *Extraction of Individual Trees Using Laser Radar Data*; Technical Report; Swedish Defence Research Agency: Stockholm, Sweden, 2001.
50. Reitberger, J. 3D Segmentierung von Einzelbäumen und Baumartenklasifikation aus Daten Flugzeuggetragener Full Waveform Laserscanner. Ph.D. Thesis, Fakultät für Bauingenieur-und Vermessungswesen der Technischen Universität München, Munich, Germany, 2010.
51. Höfle, B.; Hollaus, M.; Hagenauer, J. Urban vegetation detection using radiometrically calibrated small-footprint full-waveform airborne LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2012**, *67*, 134–147. [CrossRef]
52. Weinmann, M.; Weinmann, M.; Mallet, C.; Brédif, M. A Classification-Segmentation Framework for the Detection of Individual Trees in Dense MMS Point Cloud Data Acquired in Urban Areas. *Remote Sens.* **2017**, *9*, 277. [CrossRef]
53. Wu, B.; Yu, B.; Yue, W.; Shu, S.; Tan, W.; Hu, C.; Huang, Y.; Wu, J.; Liu, H. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sens.* **2013**, *5*, 584–611. [CrossRef]
54. Gorte, B.; Oude Elberink, S.; Sirmacek, B.; Wang, J. IQPC 2015 Track: Tree Separation and Classification in Mobile Mapping LiDAR Data. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *40*, 607–612. [CrossRef]
55. Lindenbergh, R.C.; Berthold, D.; Sirmacek, B.; Herrero-Huerta, M.; Wang, J.; Ebersbach, D. Automated large scale parameter extraction of road-side trees sampled by a laser mobile mapping system. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *40*, 589–594. [CrossRef]
56. Li, L.; Li, D.; Zhu, H.; Li, Y. A dual growing method for the automatic extraction of individual trees from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2016**, *120*, 37–52. [CrossRef]
57. Monnier, F.; Vallet, B.; Soheilian, B. Trees Detection from Laser Point Clouds Acquired in Dense Urban Areas by a Mobile Mapping System. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *I-3*, 245–250. [CrossRef]
58. Xu, S.; Xu, S.; Ye, N.; Zhu, F. Automatic extraction of street trees' nonphotosynthetic components from MLS data. *Int. J. Appl. Earth Obs. Geoinf.* **2018**, *69*, 64–77. [CrossRef]
59. Wu, J.; Yao, W.; Polewski, P. Mapping Individual Tree Species and Vitality along Urban Road Corridors with LiDAR and Imaging Sensors: Point Density versus View Perspective. *Remote Sens.* **2018**, *10*, 1403. [CrossRef]
60. Xu, Y.; Sun, Z.; Hoegner, L.; Stilla, U.; Yao, W. Instance Segmentation of Trees in Urban Areas from MLS Point Clouds Using Supervoxel Contexts and Graph-Based Optimization. In Proceedings of the 2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS), Beijing, China, 19–20 August 2018; pp. 1–5. [CrossRef]
61. Hirt, P.R.; Xu, Y.; Hoegner, L.; Stilla, U. Change Detection of Urban Trees in MLS Point Clouds Using Occupancy Grids. *PFG–J. Photogramm. Remote Sens. Geoinf. Sci.* **2021**, *89*, 301–318. [CrossRef]
62. Xie, Y.; Bao, H.; Shekhar, S.; Knight, J. A Timber Framework for Mining Urban Tree Inventories Using Remote Sensing Datasets. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 1344–1349. [CrossRef]
63. Li, W.; Fu, H.; Yu, L.; Cracknell, A. Deep Learning Based Oil Palm Tree Detection and Counting for High-Resolution Remote Sensing Images. *Remote Sens.* **2017**, *9*, 22. [CrossRef]
64. Freudenberg, M.; Nölke, N.; Agostini, A.; Urban, K.; Wörgötter, F.; Kleinn, C. Large Scale Palm Tree Detection in High Resolution Satellite Images Using U-Net. *Remote Sens.* **2019**, *11*, 312. [CrossRef]

65. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector (v5). *arXiv* **2016**, arXiv:1512.02325v5
66. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
67. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
68. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Available online: https://openaccess.thecvf.com/content_ICCV_2017/papers/Lin_Focal_Loss_for_ICCV_2017_paper.pdf (accessed on 9 February 2022).
69. Weinstein, B.G.; Marconi, S.; Bohlman, S.; Zare, A.; White, E. Individual Tree-Crown Detection in RGB Imagery Using Semi-Supervised Deep Learning Neural Networks. *Remote Sens.* **2019**, *11*, 1309. [CrossRef]
70. Weinstein, B.G.; Marconi, S.; Bohlman, S.A.; Zare, A.; White, E.P. Cross-site learning in deep learning RGB tree crown detection. *Ecol. Inform.* **2020**, *56*, 101061. [CrossRef]
71. Culman, M.; Delalieux, S.; Van Tricht, K. Individual Palm Tree Detection Using Deep Learning on RGB Imagery to Support Tree Inventory. *Remote Sens.* **2020**, *12*, 3476. [CrossRef]
72. Windrim, L.; Bryson, M. Detection, Segmentation, and Model Fitting of Individual Tree Stems from Airborne Laser Scanning of Forests Using Deep Learning. *Remote Sens.* **2020**, *12*, 1469. [CrossRef]
73. Pleșoianu, A.I.; Stupariu, M.S.; Șandric, I.; Pătru-Stupariu, I.; Drăguț, L. Individual Tree-Crown Detection and Species Classification in Very High-Resolution Remote Sensing Imagery Using a Deep Learning Ensemble Model. *Remote Sens.* **2020**, *12*, 2426. [CrossRef]
74. Branson, S.; Wegner, J.D.; Hall, D.; Lang, N.; Schindler, K.; Perona, P. From Google Maps to a fine-grained catalog of street trees. *ISPRS J. Photogramm. Remote Sens.* **2018**, *135*, 13–30. [CrossRef]
75. Lumnitz, S.; Devisscher, T.; Mayaud, J.R.; Radic, V.; Coops, N.C.; Griess, V.C. Mapping trees along urban street networks with deep learning and street-level imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *175*, 144–157. [CrossRef]
76. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Iguassu Falls, Brazil, 2017; pp. 5105–5114.
77. Graham, B.; van der Maaten, L. Submanifold Sparse Convolutional Networks. *arXiv* **2017**, arXiv:1706.01307.
78. Yan, Y.; Mao, Y.; Li, B. SECOND: Sparsely Embedded Convolutional Detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
79. Choy, C.; Park, J.; Koltun, V. Fully Convolutional Geometric Features. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 2019; pp. 8957–8965. [CrossRef]
80. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution On X-Transformed Points. In *Advances in Neural Information Processing Systems 31*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Iguassu Falls, Brazil, 2018; pp. 820–830.
81. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 6410–6419. [CrossRef]
82. Schmohl, S.; Soergel, U. Submanifold Sparse Convolutional Networks for Semantic Segmentation of Large-Scale ALS Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4*, 77–84. [CrossRef]
83. Winiwarter, L.; Mandlburger, G.; Schmohl, S.; Pfeifer, N. Classification of ALS Point Clouds Using End-to-End Deep Learning. *PFG—J. Photogramm. Remote Sens. Geoinf. Sci.* **2019**, *87*, 75–90. [CrossRef]
84. Varney, N.; Asari, V.K.; Graehling, Q. Available online: https://openaccess.thecvf.com/content_CVPRW_2020/papers/w11/Varney_DALES_A_Large-Scale_Aerial_LiDAR_Data_Set_for_Semantic_Segmentation_CVPRW_2020_paper.pdf (accessed on 9 February 2022).
85. Lin, Y.; Vosselman, G.; Cao, Y.; Yang, M.Y. Active and incremental learning for semantic ALS point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 73–92. [CrossRef]
86. Li, N.; Kaehler, O.; Pfeifer, N. A Comparison of Deep Learning Methods for Airborne LiDAR Point Clouds Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 6467–6486. [CrossRef]
87. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [CrossRef]
88. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. Available online: https://openaccess.thecvf.com/content_CVPR_2020/papers/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.pdf (accessed on 9 February 2022).
89. Song, S.; Lichtenberg, S.P.; Xiao, J. SUN RGB-D: A RGB-D scene understanding benchmark suite. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 567–576. [CrossRef]
90. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D Object Detection Network for Autonomous Driving. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6526–6534. [CrossRef]
91. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3D Proposal Generation and Object Detection from View Aggregation. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–8. [CrossRef]

92. Simon, M.; Milz, S.; Amende, K.; Gross, H.M. Complex-YOLO: An Euler-Region-Proposal for Real-Time 3D Object Detection on Point Clouds. In *Proceedings of the Computer Vision—ECCV 2018 Workshops*; Leal-Taixé, L., Roth, S., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11129, pp. 197–209. [CrossRef]
93. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection From Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12689–12697. [CrossRef]
94. Li, B. 3D fully convolutional network for vehicle detection in point cloud. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1513–1518. [CrossRef]
95. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499. [CrossRef]
96. Hu, P.; Held, D.; Ramanan, D. Learning to Optimally Segment Point Clouds. *IEEE Robot. Autom. Lett.* **2020**, *5*, 875–882. [CrossRef]
97. Qi, C.R.; Litany, O.; He, K.; Guibas, L. Deep Hough Voting for 3D Object Detection in Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 9276–9285. [CrossRef]
98. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 770–779. [CrossRef]
99. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 1951–1960. [CrossRef]
100. Ren, M.; Pokrovsky, A.; Yang, B.; Urtasun, R. SBNet: Sparse Blocks Network for Fast Inference. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8711–8720. [CrossRef]
101. Shi, S.; Wang, Z.; Wang, X.; Li, H. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *arXiv* **2019**, arXiv:1907.03670.
102. Zhu, B.; Jiang, Z.; Zhou, X.; Li, Z.; Yu, G. Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. *arXiv* **2019**, arXiv:1908.09492.
103. Wang, D.Z.; Posner, I. Voting for Voting in Online Point Cloud Object Detection. In Proceedings of the Robotics: Science and Systems, Rome, Italy, 13–17 July 2015. [CrossRef]
104. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
105. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*; LNCS; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9351, pp. 234–241. [CrossRef]
106. Bennett, S. openTrees.org [WWW Document]. 2021. Available online: http://opentrees.org (accessed on 13 August 2021).
107. Weinstein, B.G.; Graves, S.J.; Marconi, S.; Singh, A.; Zare, A.; Stewart, D.; Bohlman, S.A.; White, E.P. A benchmark dataset for canopy crown detection and delineation in co-registered airborne RGB, LiDAR and hyperspectral imagery from the National Ecological Observation Network. *PLOS Comput. Biol.* **2021**, *17*, e1009180. [CrossRef]
108. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 2015; Volume 37, pp. 448–456.
109. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
110. Forschungsgesellschaft Landschaftsentwicklung Landschaftsbau e. V. (FLL). *Baumkontrollrichtlinie—Richtlinien für Baumkontrollen zur Überprüfung der Verkehrssicherheit*, 3rd ed.; Forschungsgesellschaft Landschaftsentwicklung Landschaftsbau e. V. (FLL): Bonn, Germany, 2020.
111. Walter, V.; Kölle, M.; Yin, Y. Evaluation and Optimisation of Crowd-based Collection of Trees from 3D Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *5*, 49–56. [CrossRef]
112. Meyer, F. Topographic distance and watershed lines. *Signal Process.* **1994**, *38*, 113–125. [CrossRef]
113. Zevenbergen, L.W.; Thorne, C.R. Quantitative analysis of land surface topography. *Earth Surf. Process. Landforms* **1987**, *12*, 47–56. [CrossRef]
114. Roussel, J.R.; Auty, D.; Coops, N.C.; Tompalski, P.; Goodbody, T.R.; Meador, A.S.; Bourdon, J.F.; de Boissieu, F.; Achim, A. lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. *Remote Sens. Environ.* **2020**, *251*, 112061. [CrossRef]
115. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016; pp. 770–778. [CrossRef]
116. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
117. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]

118. Kölle, M.; Laupheimer, D.; Schmohl, S.; Haala, N.; Rottensteiner, F.; Wegner, J.D.; Ledoux, H. The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo. *ISPRS Open J. Photogramm. Remote Sens.* **2021**, *1*, 100001. [CrossRef]
119. Lian, Y.; Feng, T.; Zhou, J.; Jia, M.; Li, A.; Wu, Z.; Jiao, L.; Brown, M.; Hager, G.; Yokoya, N.; et al. Large-Scale Semantic 3-D Reconstruction: Outcome of the 2019 IEEE GRSS Data Fusion Contest—Part B. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 1158–1170. [CrossRef]