ISPRS Open Journal of Photogrammetry and Remote Sensing 14 (2024) 100078

Contents lists available at ScienceDirect

ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing

Domain adaptation of deep neural networks for tree part segmentation using synthetic forest trees

Mitch Bryson [a,*], Ahalya Ravendran [b], Celine Mercier [c], Tancred Frickey [c], Sadeepa Jayathunga [c], Grant Pearse [d], Robin J.L. Hartley [c]

[a] Australian Centre For Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, NSW 2006, Australia
[b] CSIRO, Marsfield, Australia
[c] Scion, 49 Sala Street, Private Bag 3020, Rotorua 3046, New Zealand
[d] College of Science and Engineering, Flinders University, Adelaide, Australia

ARTICLE INFO

Keywords:
Deep learning
Synthetic data
Domain adaptation
LiDAR
Laser scanning
Forests

ABSTRACT

Supervised deep learning algorithms have recently achieved state-of-the-art performance in the classification, segmentation and analysis of 3D LiDAR point cloud data in a wide-range of applications and environments. One of the main downsides of deep learning-based approaches is the need for extensive training datasets, *i.e.* LiDAR point clouds that have been annotated for target tasks by human experts. One strategy for addressing this issue is the use of simulated/synthetic data (with automatically generated annotations) for training models which can then be deployed on real target data/environments. This paper explores using synthetic data of realistic forest trees to train deep learning models for tree part segmentation from real forest LiDAR data. We develop a new pipeline for generating high-fidelity simulated LiDAR scans of synthetic forest trees and combine this with an unsupervised domain adaptation strategy to adapt models trained on synthetic data to LiDAR data captured in real forest environments.

Models were trained for semantic segmentation of tree parts using a PointNet++ architecture and evaluated across a range of medium to high-resolution laser scanning datasets collected across both ground-based and aerial platforms in multiple forest environments. Results of our work indicated that models trained on our synthetic data pipeline were competitive with models trained on real data, in particular when real data came from non-target sites, and our unsupervised domain adaptation method further improved performance. Our approach has implications for reducing the burden required in manual human expert annotation of large LiDAR datasets required to achieve high-performance from deep learning methods for forest analysis. The use of synthetically-trained models shown here provides a potential way to reduce the barriers to the use of deep learning in large-scale forest analysis, with implications to applications ranging from forest inventories to scaling-up in-situ forest phenotyping.

1. Introduction

In forests, LiDAR data from both airborne and/or ground-based sensing is used in a range of applications including forest inventory (Næsset et al., 2004; Persson et al., 2002; Ayrey and Hayes, 2018; Vandendaele et al., 2021; Lin et al., 2022; Xiang et al., 2024), in-situ forest phenotyping (Lombardi et al., 2022; Hartley et al., 2022) and assessing potential fuel loads and fire risk (Peterson et al., 2015). 3D LiDAR point clouds can provide detailed structural information within forest canopies and at the base/stems of trees that are beneficial for determining forest composition and measuring tree structural characteristics that are otherwise difficult to measure by aerial or satellite imagery. LiDAR is routinely used for forest inventory and a range of different automated data processing techniques have been developed over the last two decades for extracting tree information from point clouds (Gobakken and Naesset, 2004; Maltamo et al., 2007; Raumonen et al., 2013; Olofsson et al., 2014; Lamprecht et al., 2015; Bryson, 2017; Windrim and Bryson, 2020; Wang, 2020; Hao et al., 2022; Xiang et al., 2024).

Recently, processing techniques based on supervised deep learning have achieved state-of-the-art performance in the classification, segmentation and analysis of 3D LiDAR point cloud data in a wide-range of applications and environments (Qi et al., 2017; Li et al., 2020; Lu et al., 2022) including forest point clouds (Xi et al., 2018;

* Corresponding author.
E-mail address: mitch.bryson@sydney.edu.au (M. Bryson).

https://doi.org/10.1016/j.ophoto.2024.100078
Received 20 June 2024; Received in revised form 11 September 2024; Accepted 4 November 2024
Available online 16 November 2024
2667-3932/

Windrim and Bryson, 2020; Krisanski et al., 2021; Liu et al., 2022; Wang and Bryson, 2023; Xiang et al., 2024; Wielgosz et al., 2024). Supervised learning-based algorithms have the potential to improve upon the flexibility and robustness of traditional processing techniques by adapting and optimising to the data on which they are trained. Algorithms based on neural networks can learn from and exploit patterns in data which are difficult to encode by hand-engineered processing methods. Deep learning methods have recently been demonstrated for a wide-range of processing tasks from LiDAR including tree species identification (Hamraz et al., 2018; Chen et al., 2021; Liu et al., 2022), individual tree detection/delineation (Windrim and Bryson, 2020; Li et al., 2023; Wielgosz et al., 2024; Xiang et al., 2024), tree part segmentation (Xi et al., 2018; Wu et al., 2020; Krisanski et al., 2021; Windrim and Bryson, 2020; Kim et al., 2023), branch segmentation/skeletonisation (Digumarti et al., 2018; Dobbs et al., 2023) and tree structural parameter regression (Wang and Bryson, 2023).

A major downside of deep learning approaches is their reliance on extensive training datasets, specifically LiDAR point clouds that are annotated or labelled by human experts for target tasks. Deep learning models are typically trained on hundreds to thousands of labelled examples of task-specific input/output pairs that ideally are well matched to the environments and characteristics of the target datasets on which they are deployed. There have recently appeared a number of large-scale open source labelled datasets that can be used for training deep learning models for forest point clouds (*e.g.* Puliti et al. (2023) for delineating individual trees); however, these datasets are task specific, requiring extra labelling for new tasks (*e.g.* tree part segmentation) or different class semantics. 3D point cloud data is particularly difficult and time-consuming to annotate by human experts due to its spatially complex nature and the potential for mislabelling data is high. Labelling/annotation tools using virtual reality (Ramirez et al., 2020; Chinthammit et al., 2023) can assist in the efficiency of labelling 3D point cloud data; however, the bottleneck to widespread use of deep learning in forest LiDAR point clouds is still the cost of time and manual effort to produce training datasets.

One strategy for addressing this issue is the use of simulated/synthetic data (with automatically generated annotations) for training models which can then be deployed on real target data/environments. If simulations of environments and target objects (*i.e.* trees) can be generated with sufficient fidelity to real data collected in real environments, then an effectively unlimited amount of training data can be produced for training models, where output annotation is automatically generated by the simulation itself (*e.g.* labelling individual trees, tree parts and other structural characteristics of interest). An additional advantage of this approach is the ability to generate datasets with unambiguously correct labels, ensuring high-quality training data. Synthetic data has been successfully combined with supervised deep learning in a number of applications such as autonomous driving (Ros et al., 2016; Saleh et al., 2019; Xiao et al., 2022), plant segmentation (Ward et al., 2018), vision-based face analysis (Wood et al., 2021), space debris classification (Allworth et al., 2021) and characterisation of urban LiDAR data (Griffiths and Boehm, 2019; Uggla and Horemuz, 2021).

Relatively little work has focused on the use of synthetic data for deep learning in forest or tree point cloud applications. In Westling et al. (2021), the authors present a tree simulation process and handheld LiDAR scanning simulation in an orchard environment with the intent of use for deep learning applications; however, results are not shown of the simulation being used to train models. In Digumarti et al. (2018) the authors present algorithms for tree branch mapping/classification where models are both trained and validated on synthetic point clouds of trees; however, the ability for these models to work on real LiDAR scans of trees is not assessed. In Bryson et al. (2023), the authors present a simple model for simulating point cloud data of individual trees and assess its performance on training deep learning models for semantic segmentation which are then evaluated on real LiDAR point clouds of various forest trees. This work is the closest to the contributions presented in our paper; however, the simulation model for both tree generation and LiDAR scanning processes were simple and not necessarily linked to real physical processes, and optimal ways to combined both synthetic and real data during training were not explored.

In our paper, we further explore and evaluate the use of synthetic data of realistic forest trees for training deep learning models for tree part segmentation from forest LiDAR. We develop a new pipeline for generating high-fidelity simulated LiDAR scans of synthetic forest trees and combine this with an unsupervised domain adaptation strategy to adapt models trained on synthetic data to LiDAR data captured in real forest environments. Our synthetic data pipeline combines a biologically-accurate tree model with a high-fidelity LiDAR scanning simulation to produce synthetic training data which emulates the key properties of real forest LiDAR data. Deep learning architectures based on PointNet++ (Qi et al., 2017) are implemented to perform semantic segmentation of individual tree point clouds (*i.e.* per point classification) into two classes (central axis stem strikes vs. branches/foliage). Models based on these architectures are trained on both synthetic datasets and real datasets (with real data-trained models acting as a benchmark by which to compare the performance of synthetic data-trained models). We also develop an unsupervised domain adaptation method (based on a teacher–student knowledge distillation architecture (Caine et al., 2021; Wang and Bryson, 2023)) which uses unlabelled examples from the real target environment to further refine and improve the performance of synthetic data trained models in real target environments.

The main contributions of our work are:

1. Development of a high-fidelity simulation model for LiDAR scans of forest trees which can be used for training deep learning models for tree analysis.
2. Development of an unsupervised domain adaptation strategy that can refine synthetically-trained models using unlabelled real LiDAR target data, and
3. Evaluation of methods for synthetic data training on several real LiDAR datasets, both aerial and ground-based, and an evaluation of which simulation properties best improve performance on real data.

The results of our work have implications for reducing the burden required in manual human expert annotation of large LiDAR datasets needed to achieve high-performance from deep learning methods for forest analysis. The use of synthetically-trained models shown here provides a potential way to reduce the barriers to the use of deep learning in large-scale forest analysis, with implications to applications ranging from forest inventories to scaling-up in-situ forest phenotyping.

2. Related work

2.1. Use of forest LiDAR and point cloud data

LiDAR point clouds are routinely used for a variety of applications in forests, and a wide variety of processing techniques are used for extracting information from point clouds. Original approaches to analysing forest LiDAR consisted of area-based approaches (*e.g.* Gobakken and Naesset (2004), Maltamo et al. (2007)) in which key metrics (*e.g.* tree density, size, volume, basal area) are determined on an area-by-area basis. With the availability of higher resolution scanning, processing techniques have focused towards delineating and detecting individual trees and evaluating properties on a tree-by-tree level (Vandendaele et al., 2021; Neuville et al., 2021; Persson et al., 2022). Individual tree-level analysis algorithms can use over-segmentation and merging (Hao et al., 2022) or graph-based point cloud processing algorithms (Wang, 2020; Moorthy et al., 2020; Brede et al., 2022) to segment individual trees and their key parts, including the main tree stem and branches.

Key parameters such as stem diameter are then often determined via circle and cylinder fitting algorithms (Raumonen et al., 2013; Olofsson et al., 2014; Brede et al., 2022) using segmented points. Although these hand-engineered processing methods can often perform effectively when tailored and tuned to a specific environment, there is often significant manual effort required to tune these parameters by hand when transferring these methods to new datasets with differing structure, species or point cloud resolution.

2.2. Deep learning on forest point cloud data

Supervised deep learning methods have more recently achieved state-of-the art results in forest point cloud analysis (Xiang et al., 2024; Wielgosz et al., 2024), in which flexible algorithms, typically based on neural networks, are trained on point cloud data for specific analysis tasks. Deep learning methods have been demonstrated for tree species identification (Hamraz et al., 2018; Chen et al., 2021; Liu et al., 2022), individual tree detection/delineation (Windrim and Bryson, 2020; Li et al., 2023; Wielgosz et al., 2024; Xiang et al., 2024; Straker et al., 2023), tree part segmentation (Xi et al., 2018; Wu et al., 2020; Krisanski et al., 2021; Windrim and Bryson, 2020; Kim et al., 2023) and tree structural parameter regression (Wang and Bryson, 2023). Most recent architectures take point clouds as a direct input and process points using multiple set abstraction layers (Qi et al., 2017), sparse 3D point convolutions (Xiang et al., 2024) or self-attention networks (Zhao et al., 2021), with supervised input/output pairs used to train and optimise model parameters. In most studies, models are trained using annotated data which is provided within the specific study site by study authors. Standard benchmark datasets (such as the FOR-Instance (Puliti et al., 2023) and Treelearn (Henrich et al., 2024)) have very recently become available, and a limited number of studies have explored how these can be used to train models which are then deployed into new environments (*e.g.* Xiang et al. (2024), Wielgosz et al. (2024), Henrich et al. (2024)). The authors in Lines et al. (2022) discuss the needs and challenges of establishing large, standardised datasets which are relevant to forest environments which can be used to train and evaluate methods for LiDAR analysis.

2.3. Synthetic data for supervised machine learning

To alleviate the cost and effort required to collect and annotate large training datasets, several studies have explored the potential of using synthetic data for training supervised deep learning tasks. Synthetic data training has been used extensively in perception tasks for autonomous driving (*e.g.* Ros et al. (2016), Saleh et al. (2019), Xiao et al. (2022)) in which vision and LiDAR data is generated in large simulated urban cityscapes. The authors in Wood et al. (2021) use simulated models of human faces to generate synthetic image data which can be used to train models for face detection and feature localisation from images. In Ward et al. (2018), the authors used synthetic images of plants to train semantic segmentation algorithms for leaves. Studies in Griffiths and Boehm (2019), Uggla and Horemuz (2021) both generate synthetic LiDAR point cloud data in urban environments for training models for various perception tasks. In Bryson et al. (2023) the authors used a simple randomised tree model to train deep neural networks for tree part segmentation, where results indicated that models trained on synthetic data could out-compete models trained on limited amounts of real data.

2.4. Domain adaptation of point cloud learning tasks

Although synthetic data can prove useful for supervised deep learning, it often contains characteristics that make it distinct from real data examples. These differences are often referred to as a ‘‘domain gap’’ between the simulated and real world, and a variety of techniques are used to train with combinations of synthetic and real data that can account for this gap. This has been addressed via the use of transfer learning (Ros et al., 2016; Allworth et al., 2021), for example by training models on synthetic data and fine-tuning on limited sets of real-world examples, and more recently using techniques in domain adaptation (Saleh et al., 2019; Xiao et al., 2022). Domain adaptation methods can also be used to adapt model trained on one real dataset to another; for example the authors in Amirkolaee et al. (2024) demonstrate aerial image-based algorithms for tree counting by adapting models trained on one forest type to another using a small number of target training examples.

Common approaches to domain adaptation include learning features which are invariant to the differences between real vs. synthetic data (Tzeng et al., 2017; Tsai et al., 2019; Xiao et al., 2022) or learning how to translate synthetic data into the real-data domain (or vice versa) for example using adversarial models or contrastive learning techniques (Imbusch et al., 2022). In Saleh et al. (2019), the authors demonstrate the use of a Cycle GAN architecture for adapting synthetic training example to the real domain before using these to train a model for vehicle detection from LiDAR point clouds. Likewise the authors in Xiao et al. (2022) develop a point cloud simulator for urban driving scenes and demonstrate several domain adaptation techniques for translating synthetic point clouds to the appearance and distribution of a limited set of real point cloud examples. In the point cloud domain, the use of synthetic data combined with domain adaptation is particularly promising due to the reduced domain gap as compared to the visual domain where more complex effects such as lighting and texture may be more difficult to replicate in synthetic data.

3. Methodology

3.1. Real LiDAR datasets

LiDAR point clouds were acquired over a range of different forest environments and using different scanning techniques (airborne and ground-based) (see examples in Fig. 1). Real LiDAR datasets were used for evaluating point cloud segmentation performance, and also for training segmentation models which could be compared to models trained on synthetically-generated data. Data from three different sites/datasets were used in the study:

- **Tumut**: Mature Radiata pine trees from a forest outside Tumut, NSW, Australia captured using high-resolution airborne scanning via a helicopter-mounted Riegl VUX-1 LiDAR flying at a height of 60-90 m above the forest canopy with a resulting resolution of approximately 300–700 points per square-meter.
- **DogPark**: Recreational forest (various pine species) in Rotorua, New Zealand, captured using ground-based mobile laser scanning via a backpack-mounted Emesent Hovermap sensor, resulting in a resolution of approximately 8000 points per square-meter.
- **Kinleith**: Mature Radiata pine plantation forest plots in Kinleith Forest, New Zealand captured using a backpack-mounted Emesent Hovermap sensor. The resulting point clouds had an average point density of approximately 25000 points per square-meter.

At each site, existing processing pipelines for individual tree delineation (Windrim and Bryson, 2020; Bryson, 2017; Hartley et al., 2022) were used to extract sub point clouds that corresponded to each individual tree in the scanning area, with ground and other forest undergrowth points removed. To provide consistent resolutions at each site, each individual tree point cloud was down-sampled to 32k points per tree before further processing.

Datasets comprised of 90 trees at Tumut, 50 trees at DogPark and 105 trees at Kinleith. Points on each tree were manually labelled into one of two classes: (1) *stem* (points corresponding to the central axis stem and any large branches/leaders coming from the main stem), and (2) *foliage* (points corresponding to canopy, foliage and small branches). Each dataset was randomly split into a training split, comprising of

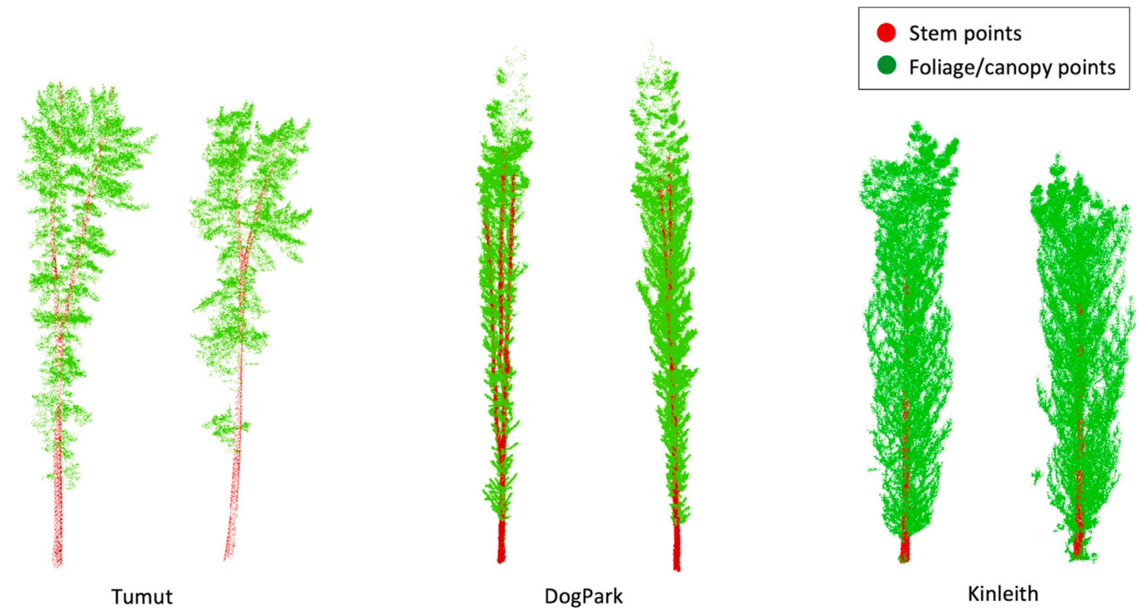

Fig. 1. Example real LiDAR point clouds of individual trees from the three different test sites. LiDAR points have been manually labelled into two classes (stem vs. foliage/branches).

50% of the trees and a testing split comprising of the remaining 50% of the trees in each dataset (used for evaluations and results presented in Section 4 below). Fig. 1 shows example labelled trees taken from various real LiDAR datasets considered.

3.2. Synthetic data generation

A simulation pipeline was developed to generate synthetic examples of forest trees and associated labelled point clouds (Davidson et al., 2023). The simulation pipeline was broken into two stages:

1. **Tree mesh generation**: the growth and structural form of trees was simulated to create a per-tree mesh model representing the geometry of the main tree stem, branches and foliage.
2. **LiDAR scanning simulation**: Individual tree meshes within a simulated forest stand were used to produce a synthetic point cloud representing a LiDAR scan of the stand. Automatic per-point labels are then applied to the final point clouds using part identification in the underlying tree meshes.

The overall simulation process is illustrated in Fig. 2 and further details of each stage are provided in the sub-sections below.

3.2.1. Synthetic tree mesh simulation

Synthetic trees were generated using 'The Grove 11' (www.thegrove3d.com) in Blender 3.2. This software allowed us to grow multiple synthetic trees in parallel within a grove, with the ability to simulate trees interacting to each other's presence in the local environment. The growth of each tree is highly customisable, with over 75 parameters governing growth, shape, response to light, and aging of branches.

Interactions between neighbouring trees and their effect on growth could be simulated in two ways. Firstly, trees could be placed together in a single scene/grove during the growing process: during the simulation of each tree's growth, each tip of each branch sends out rays, oriented to sample a hemispheric area around the tip. These rays either hit an object the tree should take into account, or move into empty space. The fewer rays that are occluded by interacting objects, the greater the 'energy' assigned to a given branch tip, and the greater the growth of the associated branch. Secondly, trees could be grown one at a time with the 'surround' option enabled in the software, which simulates a 'cylinder of shading' at a certain distance from the tree which approximates the real shading the tree would receive if it were grown within a grove. We used 70% shading at 4 m distance, and the cylinder was always resized to be as high as the tree itself. Both approaches reproduce some of the key features observable in real-world tree groves.

The source code of 'TheGrove11' was additionally modified to enable automatic segmentation of each synthetic tree into separate meshes for the 'main stem', 'branches' and 'leaves/twigs/needles' (Fig. 3). These separate meshes were subsequently used to provide ground-truth labels in the simulated LiDAR datasets described below. Mesh files for the combined geometries of simulated trees were exported as a Blender file (https://www.blender.org/) which could be passed to the LiDAR scanning simulation process.

Table 1
Parameter values and settings used in Helios++ synthetic point cloud scanning process.

Parameter/Setting	Value
Trajectory length	35 m
Altitude	25 m
Parallel track width	2 m
Platform	copter_linearpath
Platform speed	7 m/s
Scanner	riegl_vux-1uav
Scanner pulse frequency	550 000 Hz
Scanner frequency	200 Hz
Scanner trajectory time interval	0.01

3.2.2. LiDAR scanning/point cloud generation process

To generate realistic point clouds from the Blender scenes, we developed a pipeline using Helios++ (Winiwarter et al., 2022), a software specifically developed for simulated laser scanning surveys. Helios++ offers a range of scanners and platforms (*e.g.* tripod, drone-based, aircraft) and the positioning and trajectories of these platforms, as well as the scanner parameters, are all dynamically set in the pipeline. For our study, we generated trajectories using the platform 'copter_linearpath' with a speed of 7 m/s, and the laser scanner simulated was the Riegl VUX-1UAV, with a pulse frequency 550 kHz and scan frequency of 200 Hz. The tree mesh Blender files generated by our process in Section 3.2.1 were converted into scenes in XML format using the Blender2Helios Blender add-on (Neumann et al., 2023). These scenes contained references to the original tree parts which could be used to relate the points from the simulated scans to the individual tree and tree

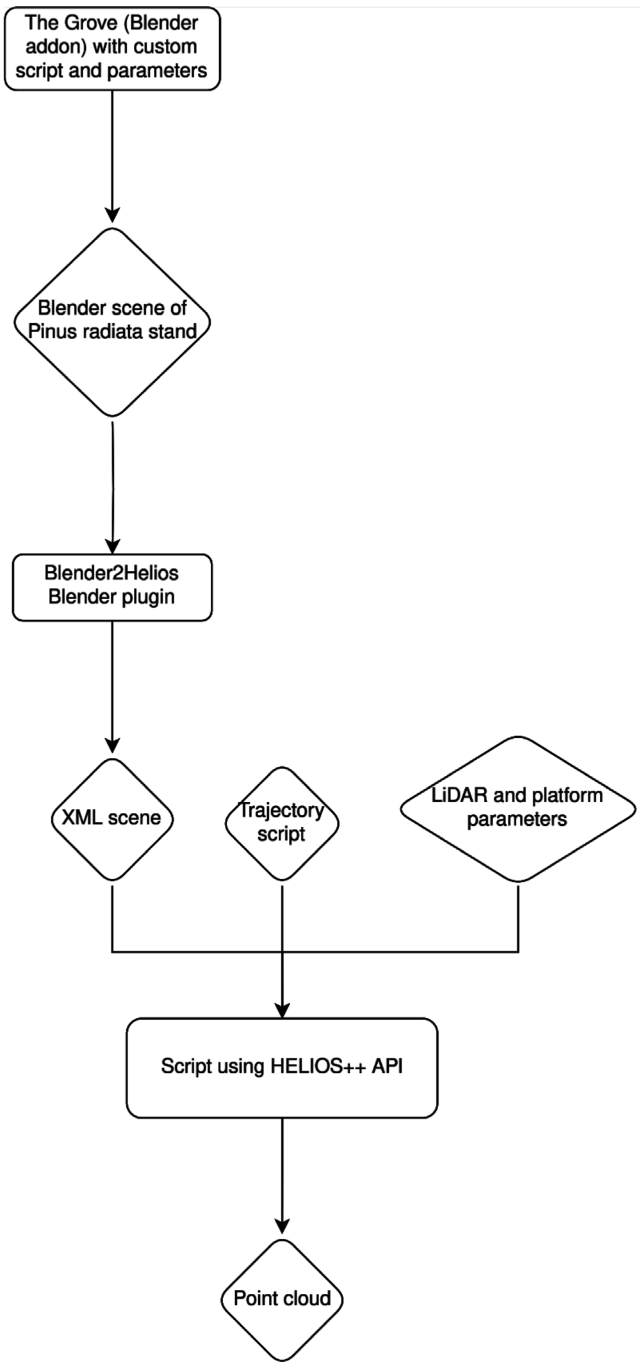

Fig. 2. Synthetic tree data generation process: tree mesh data was simulated using the The Grove 11, and model were imported into Helios++ to generate simulated LiDAR point clouds. Trajectories of a simulated scanning platform (low-flying drone) were used to generate realistic point clouds, which could later be used for training deep learning models for point cloud semantic segmentation.

parts that the point hit corresponded to. Due to the simulated scanning resolutions used (corresponding to low-altitude drone-based LiDAR) we observed that the 'leaves/twigs/needles' mesh components from 'TheGrove11' did not contribute significantly the generated point clouds, and hence were removed prior to scanning. Points corresponding to the *foliage* class hence came hits on the 'branches' mesh component, and points corresponding to the *stem* class can from the 'main stem' mesh component. Table 1 details the settings used in our study and Fig. 4 shows an example stand of virtual trees scanned with the grid-pattern trajectory of the simulated scanning platform shown.

3.2.3. Synthetically-generated datasets

In order to analyse which aspects of the synthetic data generation process were beneficial for learning, we created several synthetically-generated point cloud datasets produced using different parameters in the tree simulation process described above. Three different sets of simulated trees were considered:

- **Standard**: a dataset consisting of 400 synthetic trees, mimicking the characteristic of pine trees all from a similar age and unpruned, where a single set of grow parameters were used for all trees in TheGrove11.
- **PruneVariation**: a dataset of 100 synthetic trees which contained random variations in pruning applied to the branches of the trees. Pruning of branches was automatically performed by setting the 'auto prune enable' option in TheGrove11 to 'True', which applies pruning during last round/flush of growing.
- **AgeVariation**: a dataset of 100 synthetic trees which came from two different age classes (mature and juvenile trees), all unpruned. Two different set of growth parameters were used for each age class in TheGrove11.

Fig. 5 illustrates example labelled point clouds for each of these different sets. For each set of synthetic trees, we generated two different point cloud datasets, one corresponding to the use of the Helios++ scanning technique described in Section 3.2.2 above (Helios) and a second dataset where points were produced by simple uniform random sampling of the underlying tree meshes, without added noise (random sample).

Similarly to the real datasets, synthetic point cloud files containing a grove/stand of multiple trees were split up into per-tree sub point clouds, and each per-tree point cloud down-sampled to 32k points per tree before further processing and training to provide consistent resolutions across different datasets.

3.3. Deep learning architecture for tree part segmentation

A supervised deep learning process was implemented to develop models that could perform per-point classification (semantic segmentation) of individual tree point clouds. The PointNet++ (Qi et al., 2017) architecture was used to perform semantic segmentation, with models trained using either synthetic or real point cloud data from individual trees. Each trained model processed 4096 input points corresponding to a sample of points taken from a single tree, and provided 4096 binary class labels (stem vs. foliage/branches) corresponding to each point. Models were implemented in Keras/Tensorflow (http://keras.io) in Python using custom PointNet++ layers implemented in the PointNet++ tensorflow 2.0 layers package (https://github.com/dgriffiths3/pointnet2-tensorflow2).

The architecture used four set abstraction and four feature propagation layers, followed by two fully-connected layers and softmax output to the per-point binary class outputs. The number of MLP layers and inter-layer sampling is consistent with the original PointNet++ semantic segmentation architecture shown in Qi et al. (2017).

Models were trained from scratch (randomly initialised weights) using a cross-entropy loss function using the Adam optimiser (Kingma and Ba, 2015) with a learning rate of 1e−3 and a batch-size of 5. Models were trained for 30 epochs. During training, data augmentation was performed using random variations in sampling of N=4096 points (from 32k available per tree), random z-axis rotation and random noise added to training points.

3.4. Domain adaptation of synthetic-trained models

In order to improve the results of synthetically-trained models when deployed on real target datasets, an unsupervised domain adaptation method was implemented which exploited unlabelled examples of target trees from the real datasets. Building from approaches in Caine

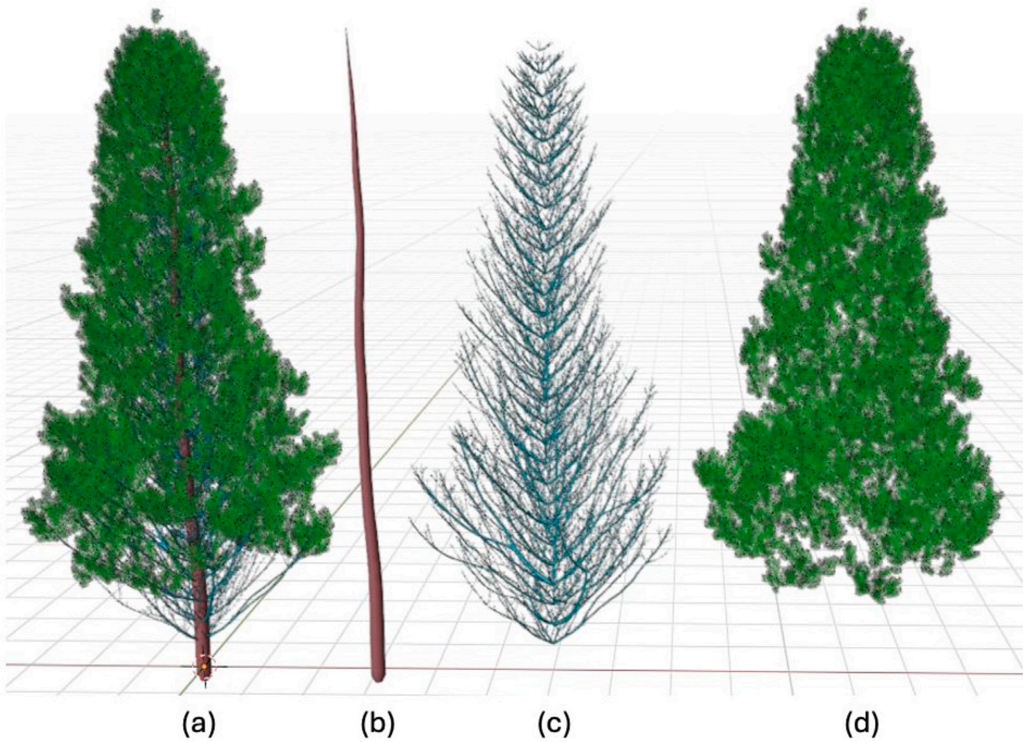

Fig. 3. Simulated Tree Meshes: (a) Example full tree mesh simulated using 'The Grove', with separate mesh components for the main stem (b), small branches (c) and needles (d).

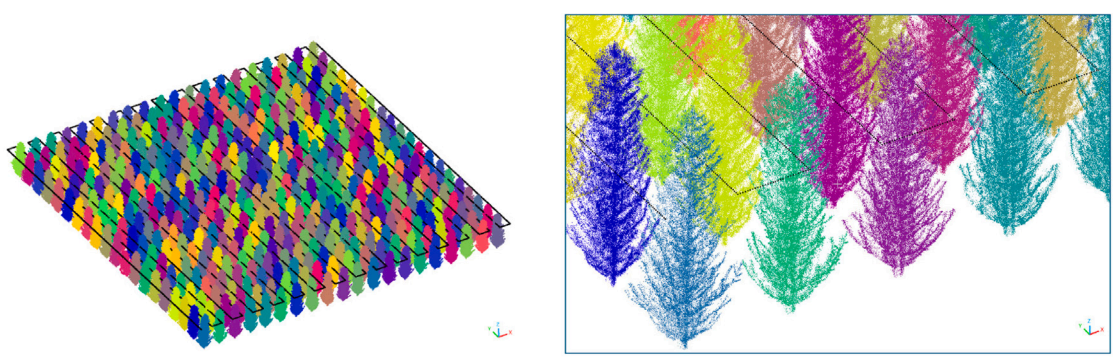

Fig. 4. LiDAR scanning/point cloud generation process: synthetic tree models are placed into a virtual forest stand and simulated point clouds generated by a simulated low-altitude aerial scanner using Helios++ (scanning platform trajectory shown in black).

et al. (2021), Wang and Bryson (2023), we implemented a teacher–student knowledge distillation approach which trains a model using a combination of labelled synthetic trees and unlabelled real trees. Although the process does use real LiDAR data from the training portion of a target site, the advantage of our unsupervised method is that these points are unlabelled, hence still not requiring any manual human effort in annotating real LiDAR data.

Fig. 6 illustrates the domain adaptation training process. Two models (one teacher and one student) use the same PointNet++ architecture described in Section 3.3. At each iteration of the training process, the teacher model is used to produce pseudo label predictions of per-point classes of real LiDAR trees from the target environment (using just the LiDAR points, without any associated labels or annotation). From these predictions, the entropy $S_{i,j}$ of each predicted point i in each tree j is calculated from the softmax output $p_{i,j}$ from the last layer of the model using:

$$S_{i,j} = -p_{i,j} \log_2 p_{i,j} \tag{1}$$

The entropy represents the inherent uncertainty of the prediction of this class from the current iteration of the teacher model. For each example tree point cloud j, the points are ranked by their entropy and only the highest entropy points are used as training data, which are sent to the student model. The student model is trained using a combination of the labelled synthetic tree examples and high entropy psuedo labels. At each iteration, the weights of the teacher model θ_t are updated using an Exponential Moving Average (EMA) of the student model weights θ_s:

$$\theta_t = \gamma\theta_t + (1-\gamma)\theta_s \tag{2}$$

where γ is the momentum. To initialise the process, we began by training the student model on labelled synthetic examples only for 30 epochs (*i.e.* same training process as in Section 3.3) and initialise the teacher weights to be equal to the student weights. We then ran the domain adaptation training strategy for a further 30 epochs, after which the student model is used as the final model for making predictions on new, unseen data. In our experiments, we used equal weights for

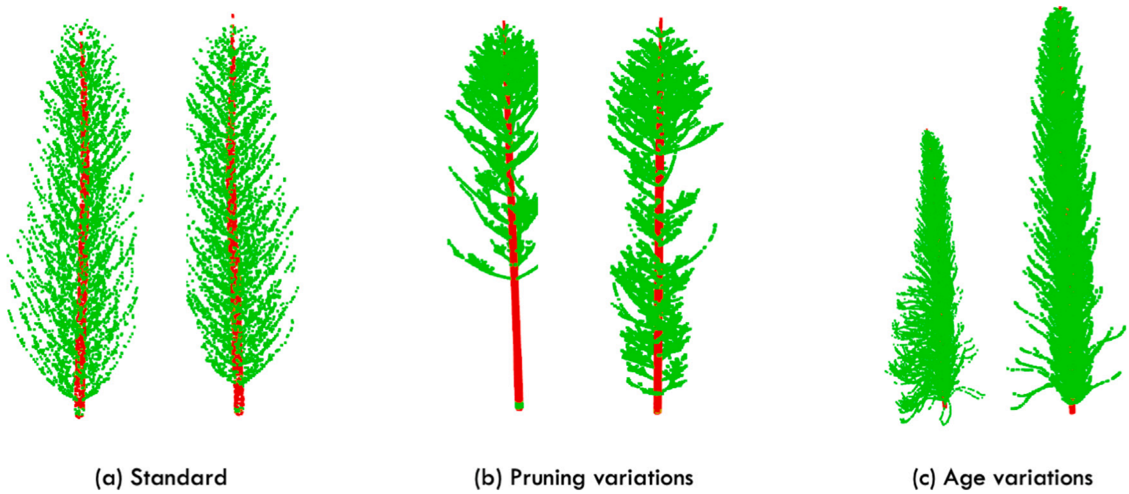

Fig. 5. Examples of synthetic tree point clouds generated using our simulation pipeline with various tree forms: (a) standard mature trees, (b) mature trees with variations in branch pruning and (c) variations in the age class of trees.

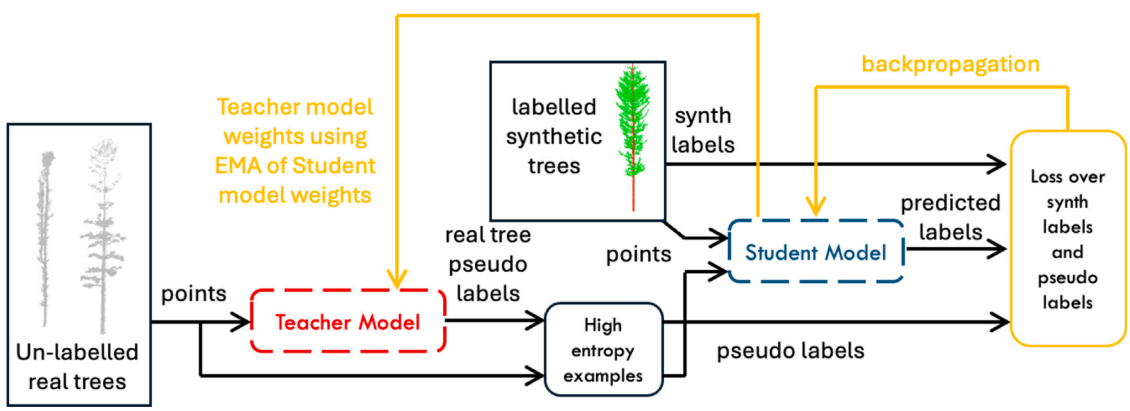

Fig. 6. Unsupervised domain adaptation using a teacher–student architecture: a teacher model is used to generate psuedo labels on unlabelled real examples. A student model is trained using labelled synthetic data and selected high-entropy psuedo labels. Weights in the teacher model are updated from the student weights using an exponential moving average. At the end of the training process, the resulting student model is used to predict on new, unseen real LiDAR data.

both synthetic and psuedo-labelled training examples, a momentum of $\gamma = 0.97$ and high entropy examples comprised of the top 75% of points for each real, psuedo-labelled tree.

4. Results

4.1. Experimental setup

Using real LiDAR test trees (Section 3.1), we evaluated the performance of models trained using our synthetic tree data generation pipeline. Several different experimental analyses were conducted:

1. We explored several variations to the synthetic data generation process (see Section 3.2.3) to see which process generated the best performing models when tested on real LiDAR data (Section 4.2).
2. Our best synthetic data trained model was compared to models trained on real data from the same target sites, and models trained on real data from non-target real datasets (Section 4.3).
3. We evaluated if the proposed unsupervised domain adaptation strategies described in Section 3.4 could further improve model performance on real target datasets (Section 4.4).

4.2. Results: Variation in synthetic data generation process

Table 2 illustrates the performance of models trained on the various types of synthetic data generation described in Section 3.2.3. For each tree type (Standard, AgeVariation, PruneVariation) we tested the two different scanning processes (Helios vs. random sample). We also tested two combined approaches, the PruneVariation and AgeVariation tree datasets, scanned using Helios (CombHelios) and the PruneVariation tree dataset using a combination of Helios scanning and random sampling (CombPrune). For each model, the average stem class Intersection over Union (IoU) was calculated for each of the three real LiDAR test sets; we show stem class IoU due to its importance in forest tree segmentation, where stem points are typically used for downstream tasks such as the determination of stem diameters and volume.

Results illustrated that the best synthetic-trained models were those using trees that exhibited variations in pruning, and that the use of high-fidelity Helios scanning slightly improved accuracy for most datasets over random surface sampling. The best overall combination of synthetic data for training came from combining Helios scanning and random sampling on the PruneVariation trees (CombPrune), which had the best performance for two of the three test sets, and second-best performance on the third test set (Table 2).

Table 2
Real test set performance (stem class IoU) for the three real datsets (Tumut, DogPark, Kinleith) using various PointNet++ models trained on different synthetic datasets. Each row indicates the performance of a single model/synthetic data generation process. Best results in each test set shown in bold.

Synthetic-trained models	Test set stem class IoU		
	Tumut	DogPark	Kinleith
Standard/Helios	0.447	0.661	0.450
Standard/random sample	0.307	0.651	0.392
AgeVariation/Helios	0.090	0.233	0.207
AgeVariation/random sample	0.313	0.661	0.325
PruneVariation/Helios	0.497	0.761	**0.518**
PruneVariation/random sample	0.570	0.743	0.448
Prune+AgeVariation/Helios (CombHelios)	0.502	0.737	0.505
PruneVariation/Helios+random sample (CombPrune)	**0.572**	**0.774**	0.507

Table 3
Real test set performance (stem class precision, recall and IoU) for three of the real datsets (Tumut, DogPark, Kinleith) using PointNet++ models trained on different real and synthetic datasets, including the Unsupervised Domain Adaptation (UDA) model. Each row indicates the performance of a single model. Best results shown in bold. Best non-target model shown in italics.

Model trained on	Tumut			DogPark			Kinleith		
	Prec.	Recall	IoU	Prec.	Recall	IoU	Prec.	Recall	IoU
Tumut (real)	**0.891**	**0.842**	**0.766**	*0.847*	0.919	*0.787*	0.687	0.613	0.482
DogPark (real)	0.638	0.602	0.460	**0.915**	0.947	**0.872**	*0.690*	0.697	0.531
Kinleith (real)	0.766	0.494	0.446	0.808	0.885	0.733	**0.778**	**0.844**	**0.685**
CombPrune (synth)	0.711	0.732	0.572	0.790	0.974	0.774	0.576	0.799	0.507
CombPrune (synth)+UDA	*0.834*	*0.737*	*0.648*	0.782	***0.991***	0.777	0.604	*0.863*	*0.553*

Table 4
Overall Results Summary: Comparison of real test set performance (stem class IoU) for models trained using real data, synthetic data and synthetic data combined with our Unsupervised Domain Adaptation (UDA) approach. Best non-target model shown in bold.

Model trained using	Test set stem class IoU		
	Tumut	DogPark	Kinleith
Trained on real target	0.766	0.872	0.685
Trained on real, non-target (best result)	0.460 (DogPark)	**0.787** (Tumut)	0.531 (DogPark)
Trained on synth data (CombPrune)	0.572	0.774	0.507
Trained on synth data (CombPrune) with UDA	**0.648**	0.777	**0.553**

4.3. Results: Synthetic data vs. real data during training

Table 3 compares the performance of models trained on the various real datasets with the best performing synthetic data trained model (CombPrune) and this model applied with the Unsupervised Domain Adaptation (UDA) technique in Section 3.4, across the three real test sets. Model performance varied for each of the test sets, but for almost all metrics the best performing models were consistently those trained on real data which came from the target site (*i.e.* see bold values). The best performing model trained on synthetic datasets (CombPrune) still performed with a very reasonable level of accuracy (stem class IoUs of 0.774 to 0.507) and were competitive with, or in the case of the Tumut test set, out-performed models trained on non-target real datasets (*i.e.* a model trained on examples of real trees from another forest/scanning site).

Fig. 7 illustrates some example predicted segmentations of real LiDAR trees using the various models shown in Table 3. Results illustrate that the synthetically-trained models exhibited the ability to segment stem points in real LiDAR scans with comparable quality to models trained on real data.

4.4. Results: Unsupervised domain adaptation results

Table 3 also shows the performance of models trained using synthetic data with our unsupervised domain adaptation process (last row) and Table 4 further summarises the main results from this table. Models trained on the real target sites continued to provide the best performance of all other training approaches. On all three test sets, our domain adaptation process improved the performance of synthetically trained models (with increases in IoU from 0.003 to 0.076) and in two of the three test sets (Tumut and Kinleith), our synthetic + UDA models out-performed all models trained on real, non-target data (IoU).

4.5. Results: training and inference time

All models were trained on a desktop computer using a single EVGA GeForce RTX 3080 Ti GPU. Model training took approximately 30 min for standard models, and approximately 60 min for UDA-training, for which models were trained for twice as many epochs. Trained models had an inference time of approximately 26 ms (per input of N=4096 points), which meant we would perform segmentation at a rate of approximately 160k points per second.

5. Discussion

5.1. Synthetic data generation process

The type and variation in simulated tree form had a significant effect on the performance of models trained on the synthetic data (Table 2). The PruneVariation trees (mature trees with variation in removed/pruned branches) were most effective at training models; this is likely due in part to the fact that real trees often present similar characteristics. For example, mature plantations of Radiata pine often lose their lower branches due to overshadowing by upper branches or combinations of mechanical damage (wind/hitting other trees), rot and insect damage. The synthetic PruneVariation trees also presented a much richer set of variation in the data during training, which likely results in more flexible/robust models being learnt with a broader set of

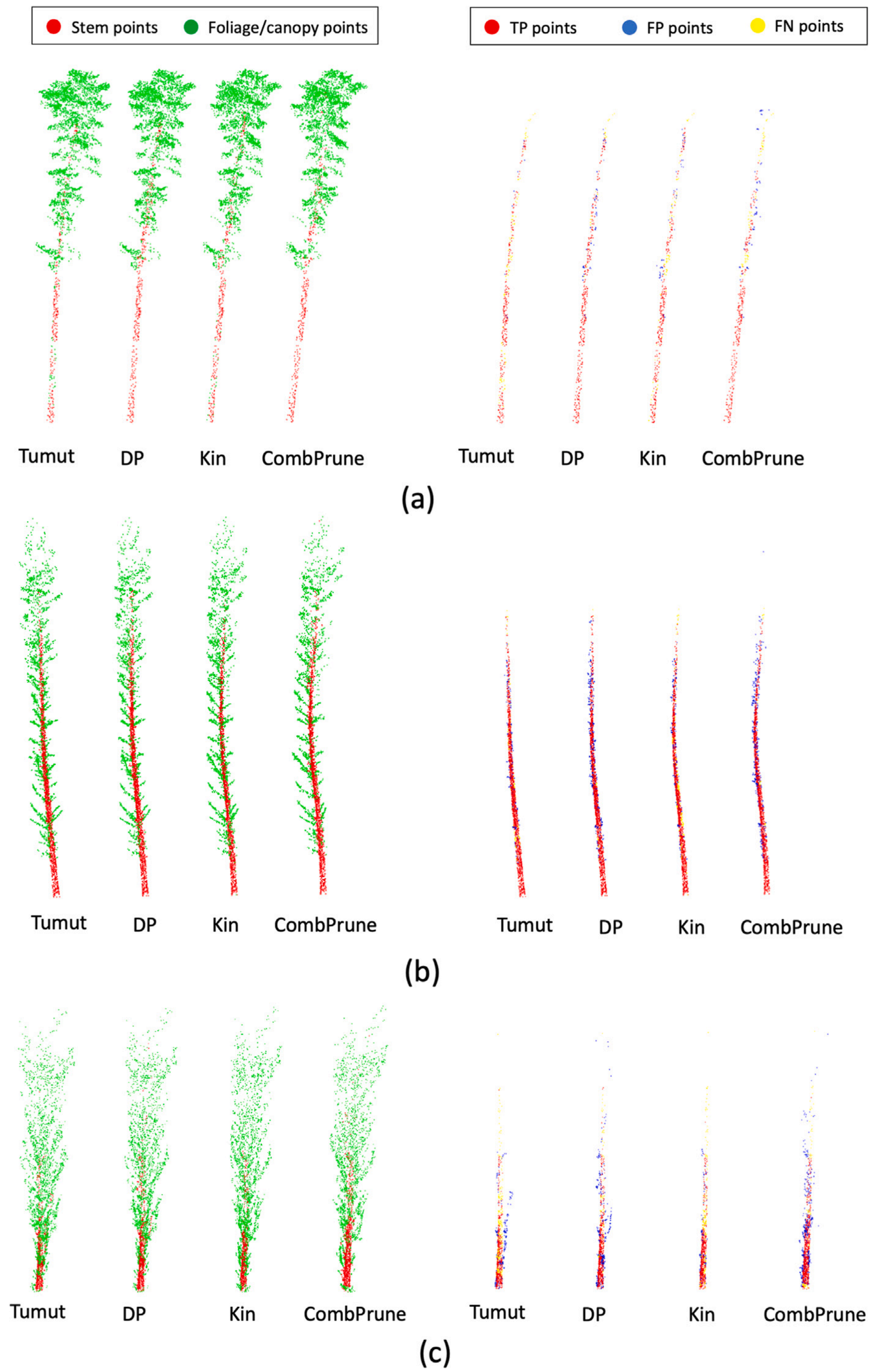

Fig. 7. Example results of predicted class labels and prediction errors using different models: (a) example tree from Tumut, (b) DogPark and (c) Kinleith datasets. Predicted class labels are shown in the left-hand figures and True Positive (TP, red), False Positive (FP, blue) and False Negative (FN, yellow) shown on the right. Models trained on Tumut, DogPark (DP), Kinleith (Kin) and PruneVariation-Helios+random sample (CombPrune) datasets.

features and mechanisms internal to the network that can interpret tree structure during segmentation. The AgeVariation models performed the worst of all tree forms considered. Juvenile trees in this dataset had much thicker canopies relative to their stem size (see Fig. 5(c)) resulting in fewer stems strikes in the simulated LiDAR. Models trained on these datasets likely had trouble optimising parameters as the segmentation of these trees was particularly difficult.

The use of a high-fidelity process for simulating the real scanning patterns of LiDAR (*i.e.* use of Helios++) had less of an impact in the overall success of models trained from the synthetic data, but still played an important contribution to the best model (Table 2). On average, models trained using the Helios++ simulated points performed slightly better than those that simulated points using simple random sampling of the tree meshes, but this was not always true for all models. The overall best model (CombPrune) used a combination of data consisting of both Helios++ simulated scans and simple random sampling. Results here indicated that successful training from synthetic data is not just a matter of ensuring simulated data is high-fidelity relative to the real environment, but also ensuring that the synthetic data exposes the model to a wide variety of target object form, while ensuring that the task (in this case semantic segmentation) is not overly difficult to solve on the synthetic data. It is possible that high-fidelity scanning processes may play a more important role in synthetic-trained models when using much lower-resolution aerial scanning data, where occlusions and scanning patterns are a more important consideration in data interpretation.

5.2. Synthetic data vs. real data for training

Results of our work indicated that models trained on our synthetic data pipeline had the ability to out-perform models trained on real data, in particular when real data came from non-target sites, and our unsupervised domain adaptation method further improved performance (Table 4). Training on synthetic has similarly been shown to be a feasible alternative to training on manually-labelled real data in other domains (Ros et al., 2016; Wood et al., 2021). Similarly to other studies (Ros et al., 2016; Bryson et al., 2023; Wang and Bryson, 2023), our results indicated that the best performing models were those trained on the real data that came from the same target sites for which the model was deployed; however, training with synthetic data was often the next best option (*i.e.* as an alternative to training on manually-labelled real data from another non-target site, Table 4).

Unsupervised domain adaptation methods based on a teacher-student knowledge distillation method have been shown to effectively adapt models from domain-shifted scenarios in real data (*e.g.* Caine et al. (2021), Wang and Bryson (2023)) and results from our paper also indicate that this method is effective at closing the sim-to-real gap in the tasks/environments presented here (Table 4). The teacher–student knowledge distillation approach to domain adaptation used in our study has several potential advantages over alternative domain adaptation techniques. The approach exploits knowledge on the appearance of real LiDAR point clouds, but being an unsupervised method, it does not rely on any supervised labels or annotation of real target data. The method is relatively simple to implement: although the overall process for training differs from that used for normal model training (*i.e.* see Fig. 6), this training process builds on and uses the existing model (*i.e.* PointNet++ architecture described in Section 3.3) without needing to make structural changes to the model itself, as opposed to methods that re-structure the model architecture (*e.g.* Saleh et al. (2019), Xiao et al. (2022), Tzeng et al. (2017), Imbusch et al. (2022)). This potentially allows for the straight-forward use of the same technique on different model architectures by placing any point cloud-consuming neural network into the place of the teacher and student models.

5.3. Comparison with state-of-the-art methods for forest point cloud learning

Direct comparisons with the performance of our approach to other state-of-the-art deep learning models for forest point clouds is difficult because of the lack of common datasets and point labelling schemes. Results from (Krisanski et al., 2021) demonstrate stem class IoUs of up to 0.91, but using much higher-resolution terrestrial laser scanning point clouds than in our study. Windrim and Bryson (2020) report stem class IoUs of 0.524 for similarly challenging aerial LiDAR datasets as we have considered, for example in the Tumut dataset. The state-of-the-art model for both tree detection and semantic segmentation in Xiang et al. (2024) achieves a mean IoU for segmentation of 0.735 (across 5 classes), which is a little higher than the average IoUs of our best models, but using a more sophisticated network architecture (sparse 3D convolutions with a UNet-like architecture) and is trained and evaluated over a broader range of different tree types than in our study.

5.4. Implications for LiDAR-based forest analysis

The outcomes of our work have implications to reducing the cost, manual burden and effort of labelling LiDAR data for training supervised machine learning models. In scenarios in which datasets are captured in new forests or using different scanning techniques/resolutions (*i.e.* aerial vs. ground), our results indicate that use of synthetic-trained models is comparable or preferable to using models trained on real LiDAR datasets from past sites. The use of synthetic data generation also lends itself to generating training datasets for new tasks, or if labelling schemes change (*i.e.* addition of new classes/finer-scale schemes for labelling) as it is likely easier to edit the automatic label generation method than to re-label/annotate an entire LiDAR training dataset by hand. The modest but useful gains offered by unsupervised domain adaptation were also notable as they allow the often large volumes of unlabelled forest point cloud data to be leveraged.

5.5. Limitations of our approach

Although the techniques presented here have several demonstrated advantages, there are a few potential limitations to the approach. Knowledge of the target tree types/species (*i.e.* Radiata Pine) allows for synthetic tree data to be generated in a targeted way, however in scenarios involving forests of unknown composition or largely heterogeneous tree species and ages, it is not clear how well our synthetic training data would transfer from one tree species to another. Generation of synthetic data also involves the use of several different pieces of software (*e.g.* The Grove, Helios++ etc.) with potentially specialised knowledge in the use of each. We expect that as interest in the benefits of synthetic data-based training grow, the ease of use of these software pipelines will also improve.

6. Conclusions and future work

We have presented a new simulation/synthetic data generation pipeline for LiDAR scans of forest trees, and validated its utility for training supervised deep learning models for an individual tree part segmentation task. Models trained using our synthetic data were shown to be competitive with models trained on real data from non-target sites, and we also demonstrated an unsupervised domain adaptation approach which further improves the results of our synthetically-trained models. Our approach has implications for reducing the burden required in manual human expert annotation of large LiDAR datasets required to achieve high-performance from deep learning methods for forest analysis. The use of synthetically-trained models shown here provides a potential way to reduce the barriers to the use of deep learning in large-scale forest analysis, with implications to applications ranging from forest inventories to scaling-up in-situ forest phenotyping.

There are a number of potential avenues for future work. We have explored the potential of unsupervised methods for domain adaptation; however, methods based on supervised domain adaptation or transfer learning (in which a small amount of *labelled* real target data is used) could have the potential to further improve results. Studies in other domains have shown that synthetic and labelled real data can be combined to further improve performance (*e.g.* Ros et al. (2016), Saleh et al. (2019)) and it seems likely that this is also true in forest analysis tasks using LiDAR.

Future work should also focus on extending the results shown here to other relevant tasks in forest point cloud analysis, such as individual tree detection and regression/estimation of finer-scale structural characteristics of trees from LiDAR. Although our work is focused on the semantic segmentation of points within individual trees, it is also possible to simulate how trees grow relative to each other in a full stand/forest under closed-canopy conditions using 'TheGrove11', and this would enable simulating high-fidelity data which could potentially be used for training models for other tasks such as tree delineation/instance detection (*i.e.* Windrim and Bryson (2020), Li et al. (2023), Wielgosz et al. (2024), Xiang et al. (2024)).

CRediT authorship contribution statement

Mitch Bryson: Writing – original draft, Validation, Supervision, Software, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Ahalya Ravendran:** Writing – review & editing, Software, Methodology, Investigation. **Celine Mercier:** Writing – review & editing, Software, Resources, Methodology, Investigation. **Tancred Frickey:** Writing – review & editing, Software, Resources, Methodology, Investigation. **Sadeepa Jayathunga:** Writing – review & editing, Resources, Methodology, Data curation. **Grant Pearse:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Robin J.L. Hartley:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was funded by the "Seeing the forest for the trees: transforming tree phenotyping for future forests" programme (C04X21 01) funded by the New Zealand Ministry of Business, Innovation and Employment (MBIE) Endeavour Fund. This work has been supported by MBIE as part of the Tree Interactions Programme, New Zealand (Catalyst Fund C09X1923). Kinleith mobile laser scanner data was captured under the Resilient Forests Research programme, funded by Scion's Strategic Science Investment Funding (SSIF), New Zealand, provided by MBIE, with additional co-funding from the NZ Forest Growers Levy Trust. We acknowledge Manulife Investment Management Forest Management NZ Ltd for providing access to their Kinleith Forest estate for the collection of that data. Thanks to David Herries, Susana Gonzales, Interpine New Zealand and HQPlantations Australia for providing access to airborne and mobile laser scanning datasets from the DogPark and Tumut sites, originally funded by National Institute for Forest Production Innovation, Australia grant NIF073-1819.

References

Allworth, J., Windrim, L., Bennett, J., Bryson, M., 2021. A transfer learning approach to space debris classification using observational light curve data. Acta Astronaut. 181, 301–315.

Amirkolaee, H.A., Shi, M., He, L., Mulligan, M., 2024. AdaTreeFormer: Few shot domain adaptation for tree counting from a single high-resolution image. ISPRS J. Photogramm. Remote Sens. 214.

Ayrey, E., Hayes, D., 2018. The use of three-dimensional convolutional neural networks to interpret LiDAR for forest inventory. Remote Sens. 10, 649.

Brede, B., Terryn, L., Barbier, N., 2022. Non-destructive estimation of individual tree biomass: Allometric models, terrestrial and UAV laser scanning. Remote Sens. Environ. 280, 113180.

Bryson, M., 2017. PointcloudITD: A Software Package for Individual Tree Detection and Counting. in Deployment and integration of cost-effective, high spatial resolution, remotely sensed data for the Australian forestry industry, FWPA Technical Report, (1), pp. 1–19.

Bryson, M., Wang, F., Allworth, J., 2023. Using synthetic tree data in deep learning-based tree segmentation using LiDAR point clouds. Remote Sens. 15 (9), http://dx.doi.org/10.3390/rs15092380.

Caine, B., Roelofs, R., Vasudevan, V., Ngiam, J., Chai, Y., Chen, Z., Shlens, J., 2021. Pseudo-labeling for scalable 3D object detection. http://dx.doi.org/10.48550/ArXiv.2103.02093.

Chen, J., Chen, Y., Liu, Z., 2021. Classification of typical tree species in laser point cloud based on deep learning. Remote Sens. 13, 4750.

Chinthammit, W., Bryson, M., Stone, C., Cao, Z., Kang, B., 2023. Operational Immersive Visualisation and Measurement Of Dense Point Cloud Data In Forest Inventory, Project Report PNC464-1718, Forest and Wood Products Australia.

Davidson, S.J., Frickey, T., Mercier, C., Wuraola, A., Tan, A., Sellier, D., 2023. From tree growth modelling to synthetic point clouds: Applications to data science. In: 10th International Conference on Functional-Structural Plant Models. FSPM2023.

Digumarti, S., Nieto, J., Cadena, C., Siegwart, R., Beardsley, P., 2018. Automatic segmentation of tree structure from point cloud data. IEEE Robot. Autom. Lett. 3.

Dobbs, H., Batchelor, O., Green, R., Atlas, J., 2023. Smart-tree: Neural medial axis approximation of point clouds for 3D tree skeletonization. In: Pattern Recognition and Image Analysis: 11th Iberian Conference, IbPRIA 2023, Alicante, Spain, June 27–30.

Gobakken, T., Naesset, E., 2004. Estimation of diameter and basal area distributions in coniferous forest by means of airborne laser scanner data. Scand. J. Forest Res. 19, 529–542.

Griffiths, D., Boehm, J., 2019. SynthCity: A large-scale synthetic point cloud. http://dx.doi.org/10.48550/ArXiv.1907.04758.

Hamraz, H., Jacobs, N., Contreras, M., Clark, C., 2018. Deep learning for conifer/deciduous classification of airborne LiDAR 3D point clouds representing individual trees. ISPRS J. Photogramm. Remote Sens. 158, 219–230.

Hao, Y., Widagdo, F., Liu, X., Liu, Y., Dong, L., Li, F., 2022. A hierarchical region-merging algorithm for 3-D segmentation of individual trees using UAV-LiDAR point clouds. IEEE Trans. Geosci. Remote Sens. 69, 5701416.

Hartley, R.J.L., Jayathunga, S., Massam, P.D., De Silva, D., Estarija, H.J., Davidson, S.J., Wuraola, A., Pearse, G.D., 2022. Assessing the potential of backpack-mounted mobile laser scanning systems for tree phenotyping. Remote Sens. 14 (14).

Henrich, J., van Delden, J., Seidel, D., Kneib, T., Ecker, A., 2024. TreeLearn: A Comprehensive Deep Learning Method for Segmenting Individual Trees from Ground-Based LiDAR Forest Point Clouds. http://dx.doi.org/10.48550/ArXiv.2309.08471.

Imbusch, B., Schwarz, M., Behnke, S., 2022. Synthetic-to-Real Domain Adaptation using Contrastive Unpaired Translation. In: IEEE International Conference on Automation Science and Engineering.

Kim, D., Ko, C., Kim, D., Kang, J., Park, J., Cho, H., 2023. Automated segmentation of individual tree structures using deep learning over LiDAR point cloud data. Forests 14 (6).

Kingma, D., Ba, J., 2015. Adam: a method for stochastic optimization. In: International Conference on Learning Representations.

Krisanski, S., Taskhiri, M., Gonzalez-Aracil, S., Herries, D., Turner, P., 2021. Sensor Agnostic Semantic Segmentation of Structurally Diverse and Complex Forest Point Clouds Using Deep Learning. Remote Sens. 13 (1413).

Lamprecht, S., Stoffels, J., Dotzler, S., Hab, E., Udelhoven, T., 2015. aTrunk - An ALS-based trunk detection algorithm. Remote Sens. 7, 9975–9997.

Li, Z.W.P., Cui, Y., Lei, S., Kang, Z., 2023. Automatic detection of individual trees in forests based on airborne LiDAR data with a tree region-based convolutional neural network (RCNN). Remote Sens. 15 (4).

Li, Y., Ma, L., Zhong, Z., Liu, F., Cao, D., Li, J., Chapman, M.A., 2020. Deep learning for LiDAR point clouds in autonomous driving: A review. http://dx.doi.org/10.48550/arXiv.2005.09830.

Lin, Y.-C., Shao, J., Shin, S.-Y., Saka, Z., Joseph, M., Manish, R., Fei, S., Habib, A., 2022. Comparative analysis of multi-platform, multi-resolution, multi-temporal LiDAR data for forest inventory. Remote Sens. 14, 649.

Lines, E., Allen, M., Cabo, C., Calders, K., Debus, A., Greive, S., Miltiadou, M., Noach, A., Owen, H., Puliti, S., 2022. AI applications in forest monitoring need remote sensing benchmark datasets. http://dx.doi.org/10.48550/ArXiv.2212.09937.

Liu, B., Chen, S., Huang, H., Tian, X., 2022. Tree species classification of backpack laser scanning data using the PointNet++ point cloud deep learning method. Remote Sens. 14, 3809.

Lombardi, E., Rodríguez-Puerta, F., Santini, F., Chambel, M.R., Climent, J., Resco de Dios, V., Voltas, J., 2022. UAV-LiDAR and RGB imagery reveal large intraspecific variation in tree-level morphometric traits across different pine species evaluated in common gardens. Remote Sens. 14 (22).

Lu, D., Xie, Q., Wei, M., Gao, K., Xu, L., Li, J., 2022. Transformers in 3D point clouds: A survey. URL https://arxiv.org/abs/2205.07417.

Maltamo, M., Suvanto, A., Packalén, P., 2007. Comparison of basal area and stem frequency diameter distribution modelling using airborne laser scanner data and calibration estimation. Forest Ecol. Manag. 247, 26–34.

Moorthy, S., Calders, K., Vicari, M., Verbeeck, H., 2020. Improved supervised learning-based approach for leaf and wood classification from LiDAR point clouds of forests. IEEE Trans. Geosci. Remote Sens. 58 (5), 3057–3070. http://dx.doi.org/10.1109/TGRS.2019.2947198.

Næsset, E., Gobakken, T., Holmgren, J., Hyyppä, H., Hyyppä, J., Maltamo, M., Nilsson, M., akan Olsson, H., Persson, Å., Söderman, U., 2004. Laser scanning of forest resources: the nordic experience. Scand. J. Forest Res. 19 (6), 482–499. http://dx.doi.org/10.1080/02827580410019553.

Neumann, M., Borrmann, D., Nuchter, A., 2023. Semantic classification in uncolored 3D point clouds using multiscale features. In: Petrovic, I., Menegatti, E., Markovic, I. (Eds.), Intelligent Autonomous Systems 17. Springer Nature, Switzerland, pp. 342–359.

Neuville, R., Bates, J., Jonard, F., 2021. Estimating forest structure from UAV-mounted LiDAR point cloud using machine learning. Remote Sens. 13, 352.

Olofsson, K., Holmgren, J., Olsson, H., 2014. Tree stem and height measurements using terrestrial laser scanning and the RANSAC algorithm. Remote Sens. 6, 4323–4344.

Persson, A., Holmgren, J., Soderman, U., 2002. Detecting and measuring individual trees using an airborne laser scanner. Photogramm. Eng. Remote Sens. 68, 925–932.

Persson, P., Olofsson, K., Holmgren, J., 2022. Two-phase forest inventory using very-high-resolution laser scanning. Remote Sens. Environ. 271, 112909.
Peterson, B., Nelson, K., Seielstad, C., Stoker, J., Jolly, W., Parsons, R., 2015. Automated integration of lidar into the LANDFIRE product suite. Remote Sens. Lett. 6, 247–256.
Puliti, S., Pearse, G., Surovy, P., Wallace, L., Hollaus, M., Wielgosz, M., Astrup, R., 2023. FOR-instance: a UAV laser scanning benchmark dataset for semantic and instance segmentation of individual trees. http://dx.doi.org/10.48550/ArXiv.2309.01279.
Qi, C., Yi, L., Su, H., Guibas, L., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: Conference on Neural Information Processing Systems. NIPS.
Ramirez, P., Paternesi, C., Luigi, L.D., Lella, L., Gregorio, D.D., Di Stefano, L., 2020. Shooting Labels: 3D Semantic Labeling by Virtual Reality. In: IEEE International Conference on Artificial Intelligence and Virtual Reality. AIVR.
Raumonen, P., Kaasalainen, M., Akerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., 2013. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. Remote Sens. 5 (2).
Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A., 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In: IEEE Conference on Computer Vision and Pattern Recognition. CVPR.
Saleh, K., Abobakr, A., Attia, M., Iskander, J., Nahavandi, D., Hossny, M., Nahvandi, S., 2019. Domain Adaptation for Vehicle Detection from Bird's Eye View LiDAR Point Cloud Data. In: International Conference on Computer Vision Workshop. ICCVW.
Straker, A., Puliti, S., Breidenbach, J., Kleinn, C., Pearse, G., Astrup, R., Magdon, P., 2023. Instance segmentation of individual tree crowns with YOLOv5: A comparison of approaches using the ForInstance benchmark LiDAR dataset. ISPRS Open J. Photogramm. Remote Sens. 9 (100045).
Tsai, Y., Sohn, K., Schulter, S., Chandraker, M., 2019. Domain Adaptation for Structured Output via Discriminative Patch Representations. In: International Conference on Computer Vision.
Tzeng, E., Hoffman, J., Saenko, K., Darrell, T., 2017. Adversarial Discriminative Domain Adaptation. In: Computer Vision and Pattern Recognition. CVPR.
Uggla, G., Horemuz, M., 2021. Towards synthesized training data for semantic segmentation of mobile laser scanning point clouds: Generating level crossings from real and synthetic point cloud samples. Autom. Constr. 130.
Vandendaele, B., Fournier, R., Vepakomma, U., Pelletier, G., Lejeune, P., Martin-Ducup, O., 2021. Estimation of northern hardwood forest inventory attributes using UAV laser scanning (ULS): transferability of laser scanning methods and comparison of automated approaches at the tree- and stand-level. Remote Sens. 13, 2796.
Wang, D., 2020. Unsupervised semantic and instance segmentation of forest point clouds. ISPRS J. Photogramm. Remote Sens. 165, 86–97. http://dx.doi.org/10.1016/j.isprsjprs.2020.04.020.
Wang, F., Bryson, M., 2023. Tree Segmentation and Parameter Measurement from Point Clouds Using Deep and Handcrafted Features. Remote Sens. 15 (4).
Ward, D., Moghadam, P., Hudson, N., 2018. Deep Leaf Segmentation Using Synthetic Data. In: British Machine Vision Conference. BMVC.
Westling, F., Bryson, M., Underwood, J., 2021. SimTreeLS: Simulating aerial and terrestrial laser scans of trees. Comput. Electron. Agric. 187 (106277).
Wielgosz, M., Puliti, S., Xiang, B., Schindler, K., Astrup, R., 2024. SegmentAnyTree: A sensor and platform agnostic deep learning model for tree segmentation using laser scanning data. arXiv:2401.15739.
Windrim, L., Bryson, M., 2020. Detection, segmentation, and model fitting of individual tree stems from airborne laser scanning of forests using deep learning. Remote Sens. 12 (9), 1–29.
Winiwarter, L., Esmorís Pena, A.M., Weiser, H., Anders, K., Martínez Sánchez, J., Searle, M., Höfle, B., 2022. Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning. Remote Sens. Environ. 269, http://dx.doi.org/10.1016/j.rse.2021.112772.
Wood, E., Baltrusaitis, T., Hewitt, C., 2021. Fake it till you make it: face analysis in the wild using synthetic data alone. In: International Conference on Computer Vision. ICCV.
Wu, B., Zheng, G., Chen, Y., 2020. An improved convolution neural network-based model for classifying foliage and woody components from terrestrial laser scanning data. Remote Sens. 12, 1010.
Xi, Z., Hopkinson, C., Chasmer, L., 2018. Filtering Stems and Branches from Terrestrial Laser Scanning Point Clouds Using Deep 3-D Fully Convolutional Networks. Remote Sens. 10, 1215.
Xiang, B., Wielgosz, M., Kontogianni, T., Peters, T., Puliti, S., Astrup, R., Schindler, K., 2024. Automated forest inventory: analysis of high-density airborne LiDAR point clouds with 3D deep learning. Remote Sens. Environ. 305, 114078.
Xiao, A., Huang, J., Guan, D., Zhan, F., Lu, S., 2022. Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation. In: AAAI Conference on Artificial Intelligence. AAAI.
Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2021. Point transformer. In: International Conference on Computer Vision. ICCV.