ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XI-1-2026
XXV ISPRS Congress 2026 “From Imagery to Understanding”, Commission I, 4–11 July 2026, Toronto, Canada

Weakly-Supervised Learning for Tree Instances Segmentation in Airborne Lidar Point Clouds

Swann Emilien Céleste Destouches[1], Jesse Lahaye[1], Laurent V. Jospin [1], Jan Skaloud [1]

[1] Environmental Sensing & Observation Laboratory (ESO), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland – swann.destouches@alumni.epfl.ch, (jesse.lahaye, laurent.jospin, jan.skaloud)@epfl.ch

Keywords: ALS, Lidar, Tree Instance Segmentation, Weakly supervised learning, Remote sensing

Abstract

Tree instance segmentation of airborne laser scanning (ALS) data is of utmost importance for forest monitoring but remains challenging due to variations in the data caused by factors such as sensor resolution, vegetation state at acquisition time, terrain characteristics, etc. Moreover, obtaining a sufficient amount of precisely labeled data to train fully supervised instance segmentation methods is expensive. To address these challenges, we propose a weakly supervised approach where labels of an initial segmentation result obtained either by a non-finetuned model or a closed-form algorithm are rated by a human operator. The labels produced during the quality assessment are then used to train a rating model, whose task is to classify a segmentation output into the same classes as specified by the human operator. Finally, the segmentation model is finetuned using feedback from the rating model. This in turn improves the original segmentation model by 34% in terms of correctly identified tree instances while considerably reducing the number of non-tree instances predicted. Challenges still remain in data over sparsely forested regions characterized by small trees (less than two meters in height) or within complex surroundings containing shrubs, boulders, etc. which can be confused as trees causing the performance of the proposed method to be reduced.

1. Introduction

Quantifying tree distribution in forests is an important application of 3D vision to either estimate its economic potential (*e.g.*, exploitable wood volume) (Ma et al., 2020), to measure the impact of climate change (Fouqueray et al., 2020), to monitor the impact on slope stability (Jiang et al., 2023), or to estimate potential for carbon capture (Dalponte and Coomes, 2016).

Individual tree crown segmentation (ITCs) in lidar points clouds remains a challenging task. While deep learning models have improved it to some extent, they still struggle to deal with variability in data resolution, species, seasonal changes (leaf on/off) and artifacts present in the terrain, especially in mountainous areas where cliffs and boulders can interfere with the segmentation. Despite recent improvements toward generalization (Wielgosz et al., 2024), fine-tuning remains a necessity which poses a challenge for general applicability of existing methods. Producing manual labels for segmentation tasks is time consuming and error prone, both for semantic (Shimoda and Yanai, 2019) and instance (Cheng et al., 2023) segmentation. In the case of remote sensing data, this can sometimes be mitigated by using 2D maps instead of 3D labels, but is still an exhaustive process (Ruoppa et al., 2025). On the other hand, access to high quality labeled data is required by deep-learning approaches for their fine-tuning on novel target area(s) of interest (*e.g.*, , alpine ecotones), which remains an issue (Fan et al., 2024). This can be partially mitigated by using noisy training data, but current methods like the one from (Weinstein et al., 2019) still rely on some hand labeled data.

To address these challenges, we propose an approach inspired by Reinforcement Learning from Human Feedback (RLHF (Chaudhari et al., 2025)). Instead of labeling the data with a segmentation mask, we provide feedback on samples from the segmentation results, which is much faster to generate by a human operator. Then we train a classification model (rating model) to imitate the rating of the human operator, which in turn provides feedback to the segmentation model; see Figure 1. While this approach is technically generalizable to any segmentation task, it is especially useful for tasks where existing algorithms yield initial results with sufficient quality to kickstart the training process, which is the case in tree instance segmentation.

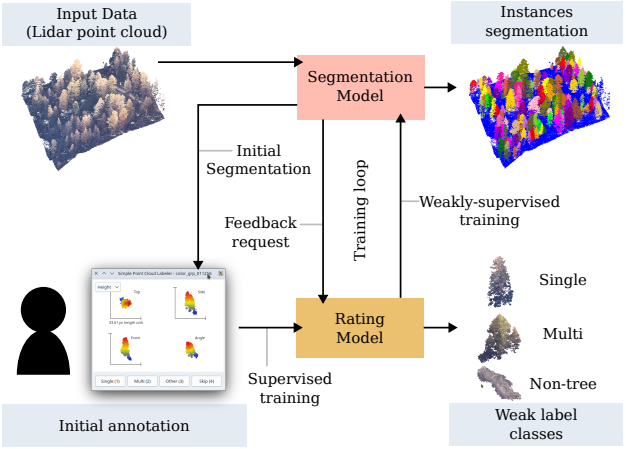


Figure 1. Proposed approach for weakly supervised training of tree segmentation.

In summary our contributions are:

- We propose employing a rating model as “a weak supervision” in training segmentation models. Our classifier obtains an accuracy of approximately 90%, which can be obtained from segmentation quality rating labels established reasonably quickly by a human operator via data inspection.
- We validate the proposed method by demonstrating an increase of 34% in the number of segmented single-tree instances in a challenging alpine dataset, and an important

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-1-2026-271-2026

decrease in the proportion of instances that do not correspond to a real tree.

- We evaluate the performance of the joint segmentation and rating model against a hand-labeled subset of the dataset. We confirm that our method successfully segments between 80% to 90% of trees present in dense forests in the study area, with major performance improvements compared to the state of the art.

2. Related work

2.1 Point cloud classification

Point classification models can be broadly divided into two categories. The first category transforms the data into a regular structure, such as 2D image grids (Rizaldy et al., 2018), 3D voxel representations (Maturana and Scherer, 2015) or slices (Huang et al., 2018), before further processing. Regular structures can be processed by classical architectures such as 2D CNN (Rizaldy et al., 2018), 3D CNN (Maturana and Scherer, 2015), RNN (Huang et al., 2018), etc. While CNN have proven easy to use and reliable for image classification tasks, their application on 3D point-clouds is not without an additional inconvenience. Indeed, in 3D data, CNN either impose using a voxel grid, which can be memory intensive in fine resolution or cause information loss in coarse resolution; or projecting 3D data onto a 2D grid, which further removes information from the 3D structure (Qi et al., 2017a). Nevertheless, these limitations can be somewhat addressed through the use of Sparse Voxel grids, which have both linear processing and memory complexity (Chen et al., 2023).

An alternative approach is to process lidar point clouds as an irregular data structure. Pioneering this idea was the PointNet architecture, which processes each point individually and uses an ordering invariant function, e.g. maximal or average pooling, to extract features (Qi et al., 2017a). A major drawbacks of this approach is the difficulty to capture local detail, while focusing on important structural points (Qi et al., 2017a). To partially mitigate this undesirable effect, PointNet++ employs a nested local partitioning of its input (Qi et al., 2017b). Self-Attention offers another efficient approach to process unstructured collections of objects, which is implemented for point clouds in architectures such as the Point Transformer (Zhao et al., 2021).

When choosing a classification architecture for the proposed learning system, different criteria are of importance, including but not limited to, accuracy, ease of training, and computational speed. We will evaluate three architectures: (i) 3DmFV (Ben-Shabat et al., 2018), a dense Voxel Net that mitigates the lower voxel grid resolution by encoding rich local features using 3D Modified Fisher Vectors, (ii) Point Transformer (Zhao et al., 2021), as a state of the art unstructured model, and (iii) a larger resolution 3D CNN we derived from VoxNet (Maturana and Scherer, 2015).

2.2 Individual tree segmentation

Closed form point cloud segmentation solutions such as watershed (Yang et al., 2020), graph cut (Lee et al., 2016) or region growing (Ma et al., 2020) offer a reasonable level of accuracy and therefore remain popular as baseline models with robust generalization ability for the task of Individual Tree Detection (ITD) and Individual Tree Crown segmentation (ITC). But they have gradually been phased out in favor of deep learning models, which yield better performances when correctly finetuned and can deal with more challenging datasets (*e.g.*, dense overlapping trees (Xiang et al., 2023)), but at the cost of more challenging generalization (albeit more recent models have shown improved results in this direction (Xiang et al., 2025)). Close form algorithms can also be used in unsupervised learning models like *e.g.*, , region growing (Ruoppa et al., 2025).

Different deep learning architectures have been adapted to the ITD and/or ITC tasks, including PointNet (Chen et al., 2021), PointNet++ (Liu et al., 2023) and 2D CNN version like YOLOv5 (Straker et al., 2023) or Mask R-CNN (Fan et al., 2024) (for the latter, the point cloud data is projected onto a raster depth map) or 3D UNET (Xiang et al., 2023).

Despite the increase of computational speed, the processing of very large outdoor datasets for ITD and ITCs remains challenging. To address this issue, (Xiang et al., 2023) proposed an architecture using Bottom-up instance grouping. Unlike Top-down instance detection, where objects have to be detected first before getting segmented and where tiling could split objects apart, the Bottom-up instance grouping assigns a feature vector to each point within a cloud which are later segmented using an unsupervised method. A major drawback of Bottom-up instance grouping is that the unsupervised segmentation step can be quite unreliable (Jiang et al., 2020). To alleviate this, the model is based on PointGroup, where multiple clustering variants are used to produce multiple candidate clusters. These are later filtered out using a ScoreNet and the overall process increases the reliability of the segmentation step.

The segmentation approach proposed by Xiang *et al.* was later combined with geometric filtering and data augmentation tailored to forests to create ForAINet (Xiang et al., 2024). This panoptic segmentation model (i.e. a model outputting both individual tree segmentation and semantic segmentation of the point cloud) is specialized for forestry. The model was later augmented with multi-resolution data to increase its generalization ability and became the *SegmentAnyTree* model (Wielgosz et al., 2024).

The ForestFormer3D model has been proposed by (Xiang et al., 2025) as a replacement for ForAINet (Xiang et al., 2024) and SegmentAnyTree (Wielgosz et al., 2024). While all these developments still rely on a 3D UNet for encoding, ForestFormer3D replaces the unsupervised segmentation step with a transformer-based layer, making the model fully trainable end-to-end. However, at the time of writing, its code base is not yet publicly available and the method has not been peer reviewed; as such is not evaluated in this work.

Although the method detailed in Sec. 3 can be adapted to employ other (or any) backbone segmentation architectures, we have selected *SegmentAnyTree* (Wielgosz et al., 2024) as the state of the art. This is due to it being the most recent peer-reviewed method and due to its training via data augmentation which should make it generalizable to datasets with varying resolution. Both aspects are interesting for the comparison in Sec. 4 and following conclusions (Sec. 5).

3. Methods

Rating the output of a segmentation algorithm is an order of magnitude faster than manually marking the exact outlines of

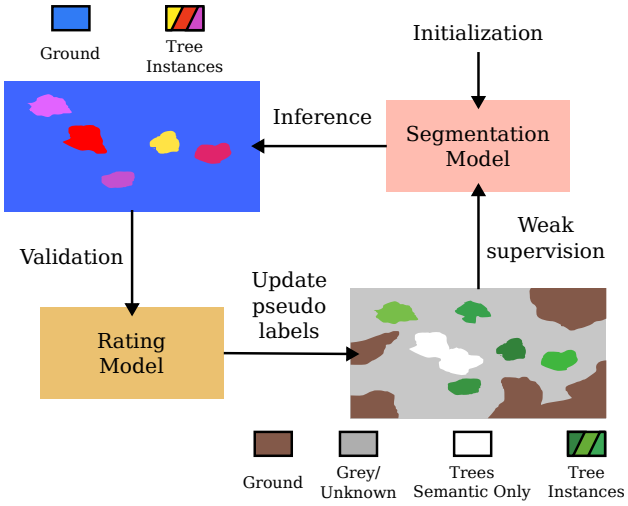


Figure 2. Training loop: knowledge is transferred from the rating model to the segmentation model via an interactive process where the output from the segmentation model is rated and used to build pseudo-labels.

(tree) clusters. But how can the information encoded in the rating be used to maximally improve the segmentation task? We propose achieving this through an iterative process shown in Figure 2 and described as follows:

1. First, we build an initial segmentation of the data, using either a pre-trained version of the segmentation model (as long as its performance is of sufficient accuracy) or a closed form solution such as, *e.g.*, watershed (Yang et al., 2020).

2. A human operator will then rate a representative sample of the initial clusters obtained in Step 1. For the task of ITCs, the proposed clusters are rated as belonging to one of three pseudo-label tree classes (***Single***, ***Multi*** or ***Non-Tree***), Sec. 3.1.

3. Then, a rating model is trained, *i.e.*, a classification model predicting the rating of a cluster, based on the “ground-truth” ratings annotated by the human operator (one of the rating models in Sec. 3.1).

4. The rating model is subsequently used to rate all clusters predicted in the initial segmentation output. From there, pseudo-label maps are built.

5. The previously obtained pseudo-labels are used to train the segmentation model (Sec. 3.2).

6. The updated segmentation predictions are fed to the rating model, the output of which is used to update the pseudo-labels (Sec. 3.3). From here, we iterate back to step 5, until the number of newly identified tree instances stabilizes.

The rating tool is described in more detail within the supplementary material. The rest of the section will focus on the rating and segmentation models.

3.1 Supervised rating model

To simplify the rating process, instead of attributing a score (which would be somewhat arbitrary and would require the operator to take more time to think), we reduce the rating to a classification problem. Each cluster is assigned to a class which is either ***Single***, when they are a single tree, ***Multi***, when multiple trees are captured in the same cluster, or ***Non-tree***, when other non-tree elements are clustered, or at least the trees are not the dominant elements in the clusters. The human operator in charge of the initial rating only needs a few seconds to make a decision for most clusters.

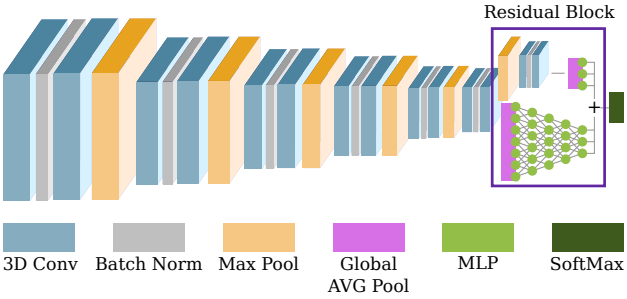


Figure 3. Architecture of proposed rating model

As for the automated rating model, as stated before, three different architectures were evaluated for the rating model. 3DmFV (Ben-Shabat et al., 2018) and Point Transformer (Zhao et al., 2021) architectures were used ”as is”, with only the final layer resized to match the three rating classes.

We also propose a VoxNet (Maturana and Scherer, 2015) architecture, shown in Figure 3, which trades the rich features of Fisher Vectors used by 3DmFV (Ben-Shabat et al., 2018) for a higher resolution. Instead of a simple occupancy function (Maturana and Scherer, 2015), we build our voxel grid using Kernel Density Estimation (KDE), which, at the cost of a slightly less sparse voxel grid, provides a much more precise estimate of the local point density, which is in itself an important feature that can be used by the classifier (Hu et al., 2022) (*e.g.*, to account for the distance between the points and the sensor, which will influence the point density).

Given a point cloud $\boldsymbol{P}$, KDE computes the value of a voxel $\boldsymbol{i}$ in the voxel grid $\boldsymbol{V}$ as:

$$\boldsymbol{V}[\boldsymbol{i}] = \sum_{\boldsymbol{p} \in \boldsymbol{P}} k(\boldsymbol{p} - \boldsymbol{i}), \tag{1}$$

with k the kernel function. We used:

$$k(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^3}} \exp\left(-\frac{1}{2}\langle \boldsymbol{x}, \boldsymbol{x} \rangle\right), \tag{2}$$

which is a Gaussian kernel with unit variance.

The voxel grid is then processed by a sequence of 3D convolutions and Max-pooling layers, halving the resolution but increasing the number of features. The last layers at $1/32$ resolution aggregates all the features using a residual block. The first connection of the residual block uses two convolution layers to reduce the voxel grid to 3 features at $1/64$ resolution which are aggregated using global average pooling. The second connection of the residual block aggregates 1024 features at $1/32$ resolution directly using global average pooling, before processing these features using a multi-layer perceptron. The features from the two branches are summed, and final activations are obtained via a Softmax layer.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-1-2026-271-2026 | © Author(s) 2026. CC BY 4.0 License.

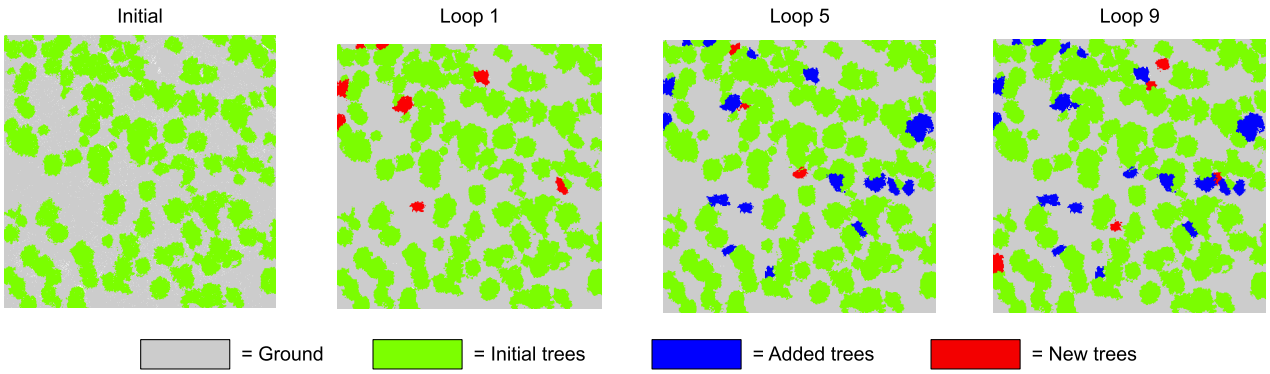


Figure 4. Iterative detection of new trees

Training is done using the ADAM optimizer. We use Batch normalization and weight decay for normalization. For all three models, we also used data augmentation to limit the risk of overfitting by adding random rotations along the z axis. The classification loss is the Cross Entropy Loss, weighted such that the total weights of each class are equivalent (as some classes, especially ***Non-tree***, can be overrepresented in the initial data), *i.e.*, the weights w are computed such that:

$$w_i c_i = w_j c_j, \tag{3}$$

for any classes i and j, where c_i is the number of samples in class i.

3.2 Self-supervised segmentation model

While all of our experiments will focus on the state of the art *SegmentAnyTree* (Wielgosz et al., 2024) method, the methodology we present here should be adaptable to any method which performs panoptic segmentation of trees, so long as it accepts a binary mask for semantic segmentation, instance mask for tree instances and does not require the training data to be fully labeled (for most loss functions, this can be implemented by setting a weight of 0 to all areas without information).

Pseudo labels are built from the tree clusters as such:

1. First, all points in the point cloud are marked as being Ground.
2. Then, all clusters or points that have been recognized as trees are marked as Gray/Unknown.
3. Finally, instances of single trees are marked on the data.

Gray/Unknown areas are assigned a weight of 0 in the loss, while Ground and Trees are processed as usual by the network (*SegmentAnyTree* supports the Gray/Unknown label by default, albeit any method should be adaptable by just setting a weight to the loss, which is supported by all usual loss functions).

The pseudo labels are used for training the model for a number of epochs n, before being updated. We performed a simple grid search to tune n and in our experiments we used $n = 3$. Albeit the optimal value will vary depending on the size of the dataset and other external factors and will need to be re-estimated in different settings.

3.3 Update to the pseudo labels

After one training loop (step 5), the point cloud is processed with the current parameters of the segmentation model, and the clusters are classified using the rating model. Clusters classified as single trees are then tested to be added to the pseudo labels.

If the candidate single tree does not intersect with any other tree (*i.e.*, it comprises only points classified as Ground or Gray/Unknown), then it is added to the pseudo labels map as a newly identified single tree.

In the case where the candidate overlaps with existing tree points, a series of tests is applied to determine whether the cluster corresponds to a new tree or an already known instance. In particular, to be considered as a new tree, the clusters are evaluated to conform to the following criteria:

1. The coordinates of the highest point in the candidate cluster, assumed to be the tip of the tree, should be different than the highest point of all intersecting clusters, as it is assumed that two trees cannot have the same tip.
2. The tree cannot intersect with other trees at a distance larger than a fixed threshold. We fix the threshold at 2m, as 25% of the expected width of a tree in the training set which was estimated to be 8m.
3. The intersection over cluster (IoC), for each cluster, is less than 0.7, with

$$\text{IoC} = \frac{\text{\# points of intersection}}{\text{\# points of cluster}}. \tag{4}$$

the intersection over cluster is used instead of the intersection over union to avoid small trees intersecting with large trees to be filtered out.

Once a cluster has passed all tests, it is added to the pseudo labels. Points in the intersection of clusters are split between the old and new clusters based on their distance with respect to the centroid of each cluster. Over time, more and more trees will be detected and added to the pseudo labels, as shown in Figure 4, until no more trees are detected.

4. Experiments

To validate the proposed procedure, we test it on a new and somewhat atypical remote sensing dataset, as it would be applied in practice. This section presents the dataset used and

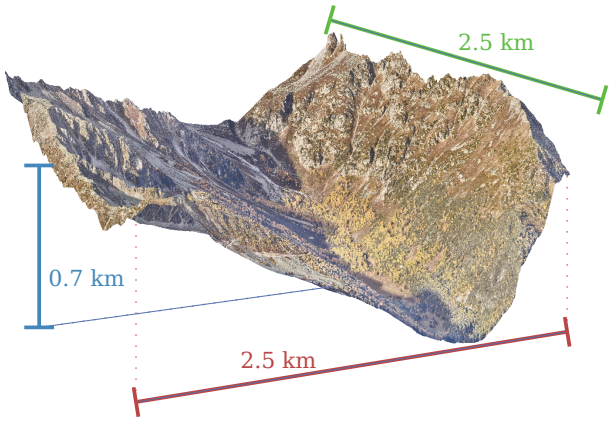


Figure 5. Colorized Lidar point cloud of the study site in the Val d’Arpette region, Switzerland.

experimental evaluation of both the rating model and segmentation model.

4.1 Dataset

The dataset we will use for this study is a lidar point cloud acquired by an aircraft over a small alpine region in the western part of Switzerland; see Figure 5, using a state-of-the-art dual channel ALS at the lowest-possible flying height over the mountain ridges to maximize point density. The data was acquired alongside a medium-format aerial camera, which was used to colorize the point cloud.

The area covers approximately $6.5\,km^2$ with vertical differences up to $\approx 700\,m$, includes 272 million points, and has an average density of $38\,pt/m^2$. The total number of trees in this small valley is estimated at around $30'000$ to $40'000$. For comparison, the FOR-instance aerial dataset used for training *SegmentAnyTree* has a point density ranging from around $500\,pt/m^2$ up to $10'000\,pt/m^2$, contained only $0.002\,km^2$ of labeled data, and $1'130$ labeled trees (a ground based dataset was also used for the initial training of *SegmentAnyTree* (Wielgosz et al., 2023), with up to $20'000\,pt/m^2$).

Our point cloud poses multiple challenges for tree segmentation methods (either closed form or deep learning) due to the following reasons:

- The very rugged terrain with steep slopes can negatively impact segmentation models. In certain cases, rocks can be wrongly classified as trees, or slope portions can be included in segmented tree instance clusters.
- The time of day and shading, which makes segmentation based on color more challenging.
- The mean point density is significantly lower in terms of what would be possible to achieve over a flat terrain from the same ALS (almost by an order of magnitude), and even lower compared to point densities obtained by drone or ground based measurements. It is also on the low end of the synthetic density used for training *SegmentAnyTree*, the lowest density being $10\,pt/m^2$ and second lowest $100\,pt/m^2$.
- The point cloud data was collected from an airplane flying at a constant altitude over steep terrain with high alpine ridges on its side. As a result, the distance between the LiDAR sensor and the ground varied, leading to variable point density in the data and adding complexity to the model training process.

Single	$3'790$	28.88%
Multi	$1'448$	10.31%
Non-tree	$7'985$	60.81%

Table 1. Initial hand classified clusters at Val d’Arpette test.

	Accuracy	Weighted Accuracy
3DmFV (Ben-Shabat et al., 2018)	83.4%	73.0%
Point Transformer (Zhao et al., 2021)	87.3%	85.5%
KDE (ours)	$\mathbf{91.6\%}$	$\mathbf{88.6\%}$

Table 2. Accuracy of the 3 studied classifier models

An initial segmentation of the full dataset was performed using a commercial implementation of the Watershed algorithm (Yang et al., 2020), yielding $38'609$ clusters. Of these clusters, $13'316$ were manually rated as representing one of the three classes described in Section 3.1. Table 1 reports the proportion of the different classes for the hand-classified data.

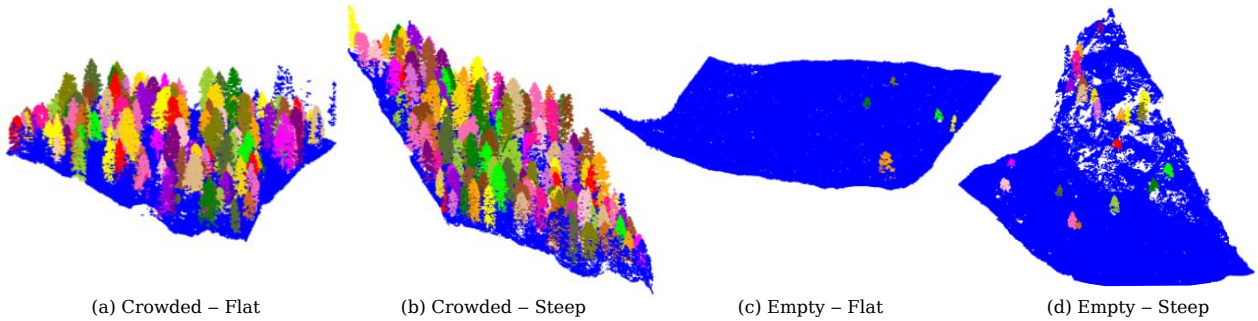
Due to the computational resources required to train the segmentation model, the dataset was tiled. The size of the tile was set at $100\,m \times 100\,m$, a compromise between the size of the tile, ensuring that it contains enough trees, and the processing power of the GPU used for training. The selected tiles were split into a train set, for our method to be executed on, and a test set, reserved for evaluation purposes.

We manually labeled all tree instances of four titles from the test set. To study the behavior of the proposes pipeline in different conditions, we selected a tile in a flat and crowded region, a flat and steep region, an empty and flat region and an empty and steep region; see Figure 6. The ground truth labels are a mix of manually selected tree instances and instances that were missed by the human operator but automatically detected and later validated visually, showing the limit of hand labeling tree clusters. Indeed, manually segmenting trees is not only time consuming, it is also error prone. The major source of error we observed during the labeling process is related to trees that are very close together and that the human operator labeled as a single tree.

4.2 Classification accuracy

For each of the three models under consideration, a grid search was performed to tune the most important hyper-parameters of the model. The accuracy and weighted accuracy (*i.e.*, accuracy of the model assuming each class having the same weight) are reported in Table 2. The KDE base model we developed reached the highest accuracy ($\sim 90\%$) that is, however, closely followed by the Point Transformer architecture. One of the main reasons is that local texture is very important when detecting trees in lidar data. Indeed, airborne laser scanners often operate in frequencies that allow their laser to traverse the upper layer of vegetation and detect deeper branches and leaves, and sometimes even the ground below the vegetation. The 3DmFV (Ben-Shabat et al., 2018) model struggles to capture this due to its lower resolution, and the Point Transformer architecture, as an architecture optimized for irregular data structure processing, also struggles a little bit.

Note that the performance of all models decreases when each class is re-weighted. Despite the weights in the Cross Entropy



(a) Crowded – Flat (b) Crowded – Steep (c) Empty – Flat (d) Empty – Steep

Figure 6. Labeled ground truth tiles.

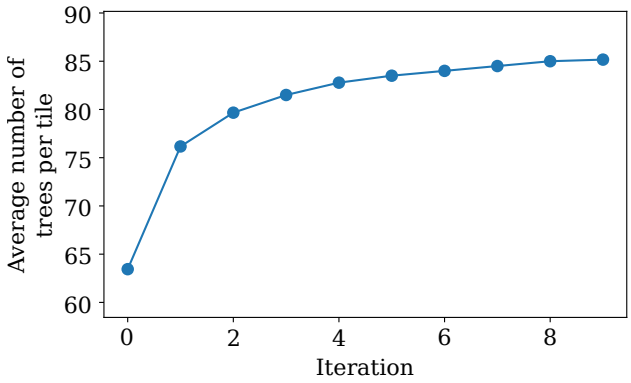


Figure 7. Average trees per tile at each iteration of the pipeline

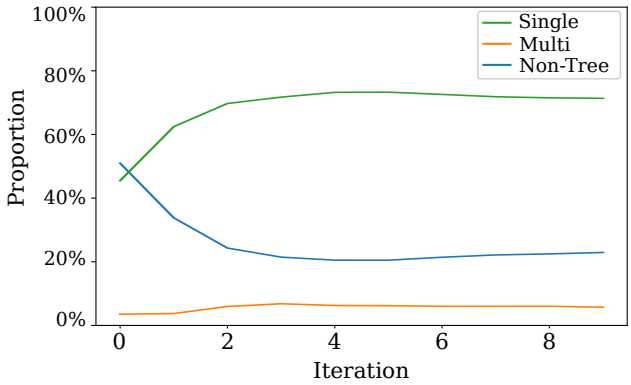


Figure 8. Proportion of the different rating classes predicted by the rating model as the training progresses

loss, the models did overfit the ***Non-tree*** class slightly, as this class is overrepresented in the training data. Nevertheless, the effect is minimal, so we opted to proceed with the KDE VoxNet based classifier as our rating model.

4.3 Single tree segmentation

To evaluate the proposed iterative segmentation approach, we will use the classification rating model to estimate the quality of the segmentation. This is to check if the number of segmented single trees increases per iteration, as well as if the number of multi-tree and non-tree clusters decreases. We retrained the *SegmentAnyTree* model on our data using the pseudo labels established by the rating model at a learning rate of $5 \cdot 10{-}5$, which was found to provide the best compromise between speed and stability of training.

As shown in Figure 7, the average number of identified trees per tile (surface of one hectare) increased from 63.4 at iteration 0 to 85.2 at iteration 9, an increase of 34.3 trees per hectare. The increase in the number of detected trees is very sharp at the beginning and stabilizes over time. Figure 8 also shows that the proportion of ***Non-tree*** samples goes down as training progresses, while the proportion ***Multi*** clusters remains more or less constant. This indicates that the proposed training scheme increases the number of retrieved ***Single*** tree instances and reduces the number of clusters assigned as ***Non-tree***.

4.4 Segmentation with respect to manual ground truth

While the increase in detected trees by the segmentation model is encouraging, it is also important to compare it with the actual number of trees on the ground. For that we employ the manually labeled ground truth depicted in Figure 6.

We take the output of either the original *SegmentAnyTree* model, or its evolved version by our approach, and process it with the described rating model so that non-***Single*** tree clusters are filtered out. This shows another benefit of our approach: the rating model can be combined with the segmentation model at inference time to refine its output. We finally compare the number of predicted trees in the cluster against the ground truth.

	# instances
(a) Crowded - flat	
SegmentAnyTree Original	136
SegmentAnyTree + Ours	**162**
Ground truth	182
(b) Crowded - steep	
SegmentAnyTree Original	162
SegmentAnyTree + Ours	**189**
Ground truth	200
(c) Empty - flat	
SegmentAnyTree Original	2
SegmentAnyTree + Ours	2
Ground truth	5
(d) Empty - steep	
SegmentAnyTree Original	10
SegmentAnyTree + Ours	**15**
Ground truth	22

Table 3. Instances detected with respect to ground truth for labeled ground truth

The results, reported in Table 3 per terrain steepness and tree population density, show that while our method does not reach a coverage of 100%, it increases the number of detected instances. The performance is not as strong for sparsely populated tiles containing few isolated trees. This indicates that the method is challenged by the presence of small shrubs that are out of domain compared to the typical tree in the dataset.

A more detailed analysis of the results is shown in Figure 9. For each cluster in the ground truth, we selected the cluster in

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-1-2026-271-2026 | © Author(s) 2026. CC BY 4.0 License.

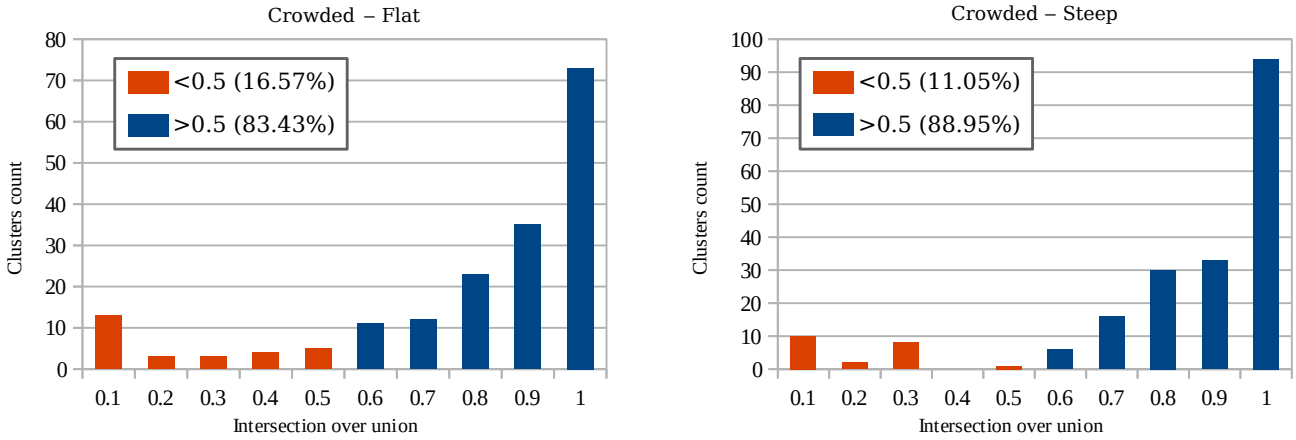


Figure 9. IOU between ground truth clusters and corresponding predictions of the finetuned model on ground truth crowded tiles

the prediction with the highest IOU and plot the resulting IOU distribution. The predicted clusters seems to be in very good agreement with the ground truth for densely populated areas independent of slope steepness. Only a certain number of trees from the ground truth were missed, leading to very small IOU. For crowded tiles more than 80% of ground truth clusters have an IOU of more than 0.5 with their corresponding predicted clusters. The results are more concerning for “empty” tiles (discussed in the supplementary material), where not only were most of the present small trees not detected, but also some proportion of the segmented trees do not match the ground truth.

5. Conclusion

We proposed a method which enables employing efficiently obtained rating labels by a human operator to facilitate instance segmentation of trees in lidar point cloud data obtained in a challenging environment. The proposed method resulted in a significant increase in the proportion of single trees detected by a state-of-the-art deep learning model in alpine terrain close to a tree limit and, when combined with our automatic rating model, enabled the recovery of more than 80% of the trees in the dense validation tiles that were annotated by hand. Yet, challenges remain in detecting small and isolated trees growing in rough terrain. This is partially due to the fact that the trees in these regions are very small and out of distribution compared to the main population of trees in the forest. Future research might address this by building an ensemble of models fit for different type of vegetation cover rather than focusing on a single model.

Our results demonstrate the potential benefits of using a simple rating model for weak supervision within instance segmentation of tree individuals. The rating model can also be used in conjunction with the segmentation model to filter out invalid clusters, providing additional value to the segmentation model.

6. Acknowledgments

Swann Emilien Céleste Destouches conducted the experiments described in this paper under the supervision of Jesse Lahaye and Laurent Jospin. Experimental design was done by Laurent Jospin with additional contributions by Swann Emilien Céleste Destouches. The tool for rating point clusters was developed by Laurent Jospin. The tool used for hand labeling the clusters in the point cloud data was developed by Swann Emilien Céleste Destouches. The overall research was directed by Jan Skaloud. Thanks to Swiss Flying Services and SixSense Helimap SA for facilitating the acquisition and preprocessing of our dataset. The dissemination of open data in this research was supported by the Open Research Data Program of the ETH Board.

References

Ben-Shabat, Y., Lindenbaum, M., Fischer, A., 2018. 3DmFV: Three-Dimensional Point Cloud Classification in Real-Time Using Convolutional Neural Networks. *IEEE Robotics and Automation Letters*, 3(4), 3145-3152. 2, 3, 5

Chaudhari, S., Aggarwal, P., Murahari, V., Rajpurohit, T., Kalyan, A., Narasimhan, K., Deshpande, A., Castro da Silva, B., 2025. RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs. *ACM Comput. Surv.* 1

Chen, X., Jiang, K., Zhu, Y., Wang, X., Yun, T., 2021. Individual Tree Crown Segmentation Directly from UAV-Borne LiDAR Data Using the PointNet of Deep Learning. *Forests*, 12(2). https://www.mdpi.com/1999-4907/12/2/131. 2

Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J., 2023. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21674–21683. 2

Cheng, T., Wang, X., Chen, S., Zhang, Q., Liu, W., 2023. Boxteacher: Exploring high-quality pseudo labels for weakly supervised instance segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3145–3154. 1

Dalponte, M., Coomes, D. A., 2016. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. *Methods in Ecology and Evolution*, 7(10), 1236-1245. 1

Fan, W., Tian, J., Troles, J., Döllerer, M., Kindu, M., Knoke, T., 2024. Comparing Deep Learning and MCWST Approaches for Individual Tree Crown Segmentation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-1-2024, 67–73. https://isprs-annals.copernicus.org/articles/X-1-2024/67/2024/. 1, 2

Fouqueray, T., Charpentier, A., Trommetter, M., Frascaria-Lacoste, N., 2020. The calm before the storm: How climate change drives forestry evolutions. *Forest Ecology and Management*, 460, 117880. 1

Hu, J. S. K., Kuai, T., Waslander, S. L., 2022. Point density-aware voxels for lidar 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8469–8478. 3

Huang, Q., Wang, W., Neumann, U., 2018. Recurrent slice networks for 3d segmentation of point clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2

Jiang, H., Zou, Q., Zhou, B., Jiang, Y., Cui, J., Yao, H., Zhou, W., 2023. Estimation of Shallow Landslide Susceptibility Incorporating the Impacts of Vegetation on Slope Stability. *International Journal of Disaster Risk Science*, 14(4), 618-635. 1

Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.-W., Jia, J., 2020. Pointgroup: Dual-set point grouping for 3d instance segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2

Lee, J., Cai, X., Lellmann, J., Dalponte, M., Malhi, Y., Butt, N., Morecroft, M., Schönlieb, C.-B., Coomes, D. A., 2016. Individual Tree Species Classification From Airborne Multisensor Imagery Using Robust PCA. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(6), 2554-2567. 2

Liu, Y., You, H., Tang, X., You, Q., Huang, Y., Chen, J., 2023. Study on Individual Tree Segmentation of Different Tree Species Using Different Segmentation Algorithms Based on 3D UAV Data. *Forests*, 14(7). 2

Ma, Z., Pang, Y., Wang, D., Liang, X., Chen, B., Lu, H., Weinacker, H., Koch, B., 2020. Individual Tree Crown Segmentation of a Larch Plantation Using Airborne Laser Scanning Data Based on Region Growing and Canopy Morphology Features. *Remote Sensing*, 12(7). 1, 2

Maturana, D., Scherer, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 922–928. 2, 3

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds), *Advances in Neural Information Processing Systems*, 30, Curran Associates, Inc. 2

Rizaldy, A., Persello, C., Gevaert, C. M., Oude Elberink, S. J., 2018. FULLY CONVOLUTIONAL NETWORKS FOR GROUND CLASSIFICATION FROM LIDAR POINT CLOUDS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2, 231–238. 2

Ruoppa, L., Oinonen, O., Taher, J., Lehtomäki, M., Takhtkeshha, N., Kukko, A., Kaartinen, H., Hyyppä, J., 2025. Unsupervised deep learning for semantic segmentation of multispectral LiDAR forest point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 228, 694-722. https://www.sciencedirect.com/science/article/pii/S0924271625003089. 1, 2

Shimoda, W., Yanai, K., 2019. Self-supervised difference detection for weakly-supervised semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 1

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 Journal of Photogrammetry and Remote Sensing*, 9, 100045. 2

Weinstein, B. G., Marconi, S., Bohlman, S., Zare, A., White, E., 2019. Individual Tree-Crown Detection in RGB Imagery Using Semi-Supervised Deep Learning Neural Networks. *Remote Sensing*, 11(11). https://www.mdpi.com/2072-4292/11/11/1309. 1

Wielgosz, M., Puliti, S., Wilkes, P., Astrup, R., 2023. Point2Tree(P2T)—Framework for Parameter Tuning of Semantic and Instance Segmentation Used with Mobile Laser Scanning Data in Coniferous Forest. *Remote Sensing*, 15(15). 5

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. *Remote Sensing of Environment*, 313, 114367. 1, 2, 4

Xiang, B., Peters, T., Kontogianni, T., Vetterli, F., Puliti, S., Astrup, R., Schindler, K., 2023. TOWARDS ACCURATE INSTANCE SEGMENTATION IN LARGE-SCALE LIDAR POINT CLOUDS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-1/W1-2023, 605–612. 2

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 Sensing of Environment*, 305, 114078. 2

Xiang, B., Wielgosz, M., Puliti, S., Král, K., Krůček, M., Missarov, A., Astrup, R., 2025. Forestformer3d: A unified framework for end-to-end segmentation of forest lidar 3d point clouds. 2

Yang, J., Kang, Z., Cheng, S., Yang, Z., Akwensi, P. H., 2020. An Individual Tree Segmentation Method Based on Watershed Algorithm and Three-Dimensional Spatial Distribution Analysis From Airborne LiDAR Point Clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 1055-1067. 2, 3, 5

Zhao, H., Jiang, L., Jia, J., Torr, P. H., Koltun, V., 2021. Point transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16259–16268. 2, 3, 5