Science of Remote Sensing 14 (2026) 100513

Contents lists available at ScienceDirect

Science of Remote Sensing

journal homepage: www.sciencedirect.com/journal/science-of-remote-sensing

Full length article

NormalView: Tree species classification from backpack and aerial lidar data using geometric projections

Juho Korkeala [a], Jesse Muhojoki [a,*], Josef Taher [a], Klaara Salolahti [a], Matti Hyyppä [a], Antero Kukko [a,b], Juha Hyyppä [a,b]

[a] Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute FGI, The National Land Survey of Finland, Vuorimiehentie 5, Espoo, FI-02150, Finland

[b] Department of Built Environment, School of Engineering, Aalto University, P.O. Box 11000, Espoo, FI-00076, Finland

ARTICLE INFO

Dataset link: https://zenodo.org/records/21039101

Keywords:
Tree species classification
Point cloud
Mobile laser scanning
Backpack laser scanning
Deep learning
Multispectral ALS

ABSTRACT

Laser scanning has proven to be an invaluable tool in assessing the decomposition of forest environments. Mobile laser scanning (MLS) has shown to be highly promising for extremely accurate, tree level inventory. In this study, we present NormalView, a projection-based deep learning method for classifying tree species from point cloud data. NormalView embeds local geometric information into two-dimensional projections, in the form of normal vector estimates, and uses the projections as inputs to an image classification network, YOLOv11. In addition, we inspected the effect of multispectral radiometric intensity information on classification performance. We trained and tested our model on high-density MLS data (7 species, ~ 6000 pts/m^2), as well as high-density airborne laser scanning (ALS) data (9 species, >1000 pts/m^2). On the MLS data, NormalView achieves an overall accuracy (macro-average accuracy) of 95.5% (94.8%), and 91.8% (79.1%) on the ALS data. We found that having intensity information from multiple scanners provides benefits in tree species classification, and the best model on the multispectral ALS dataset was a model using intensity information from all three channels of the multispectral ALS. This study demonstrates that projection-based methods, when enhanced with geometric information and coupled with state-of-the-art image classification backbones, can achieve exceptional results. Crucially, these methods rely only on geometric information, and thus are compatible with most sensors. Additionally, we publicly release the MLS dataset used in the study, containing 1915 samples.

1. Introduction

Tree species classification is a key tool in understanding forest and urban ecosystems. Understanding the composition of species helps monitor biodiversity, manage forest risks, and to understand the ecological richness of forest environments (Chirici et al., 2012; Gamfeldt et al., 2013; Huuskonen et al., 2021). Accurate species classification also has economic impact (Haara et al., 2019). In boreal forests, it is found that aspen (*Populus tremula*) holds an important role as a keystone species (Kivinen et al., 2020; Kouki et al., 2004), serving as a biodiversity hotspot hosting hundreds of species, including woodpeckers and flying squirrels (Angelstam and Mikusiński, 1994; Tikkanen et al., 2006; Remm et al., 2017). In recent years, automated forest inventory has started to replace more traditional methods in forest inventory. Recently, we have seen deep learning-based methods for semantic segmentation of forests (see e.g. Wielgosz et al. (2024), Henrich et al. (2024), Ruoppa et al. (2025), Xiang et al. (2025)). In combination, these segmentation methods and species classification methods can provide us with forest information almost completely automated.

Mobile laser scanning (MLS) and airborne laser scanning (ALS) have proven to be invaluable tools in forestry. The applications of MLS include estimation of stem curves (Bauwens et al., 2016; Hyyppä et al., 2020; Gollob et al., 2020), extracting branch information (Winberg et al., 2023), and collecting forest inventories (Liang et al., 2014; Mokroš et al., 2021). ALS data has been used for individual tree detection and their characteristics' assessment (Hyyppä and Inkinen, 1999; Windrim and Bryson, 2020; Jarahizadeh and Salehi, 2026), change detection (Hyyppä et al., 2003; Yu et al., 2004; Ma et al., 2018; Arumäe et al., 2020; Kozniewski et al., 2022; Sun et al., 2022), biomass estimations (Dalponte et al., 2018; Zhao et al., 2018) and semantic segmentation of forests (Xiang et al., 2024; Ruoppa et al., 2025). Both data types have been used for the classification of tree species (Puliti et al., 2025b; Taher et al., 2026). Accurate tree species classification

* Corresponding author.
E-mail address: jesse.muhojoki@nls.fi (J. Muhojoki).

https://doi.org/10.1016/j.srs.2026.100513
Received 22 July 2026; Received in revised form 16 September 2026; Accepted 23 September 2026
Available online 5 October 2026
2666-0172/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

from dense point clouds — such as MLS or close-range ALS data — enables collection of reference data efficiently for large-area forest inventories. The combination of highly accurate stem curves, efficient collection, and tree species classification is required, as the stem curve and taper models are highly dependent on the species (Salekin et al., 2021; McTague and Weiskittel, 2021) and are changing over time due to, e.g. climate change (Schneider et al., 2018), and the tree stems are not visible from ALS used for large-area laser scanning, and therefore estimating the stem attribute from other attributes (mostly canopy) is required.

To classify trees from point cloud data, various methods have been presented previously. The methods utilising artificial intelligence can be roughly divided into three categories: machine learning based-methods, projection-based deep learning methods, and point-based 3D deep learning methods. Machine learning (ML) methods derive hand-crafted features from tree point clouds, such as tree height, crown area, height percentiles, and intensity and reflectance statistics. For an extensive review of various features used in tree species classification, see Koenig and Höfle (2016). Common ML classifiers include random forests (Breiman, 2001), support vector machines (Cortes and Vapnik, 1995), and linear discriminant analysis (Izenman, 2013). These algorithms have been successfully applied in several tree species classification studies, for example by Hakula et al. (2023), Dalponte et al. (2012), Ørka et al. (2012), Tockner et al. (2025), Yu et al. (2017) for random forests, by Dalponte et al. (2012), Ørka et al. (2012) for support vector machines, and by Ørka et al. (2012), Hardenbol et al. (2021) for linear discriminant analysis.

The recent rapid developments in the field of deep learning (DL) have made DL methods surpass more traditional ML methods, as seen in the benchmarks by Puliti et al. (2025b) and Taher et al. (2026). Projection-based methods operate by creating two-dimensional representations of tree point clouds, for example by taking orthographic projection images from multiple angles and colouring the images black-and-white (B&W) (Marinelli et al., 2022; Seidel et al., 2021; Blackburn et al., 2025; Puliti et al., 2025b; Taher et al., 2026), according to depth (Allen et al., 2022), intensity (Blackburn et al., 2025) or geometric features (Blackburn et al., 2025). The classification task is largely studied in the field of image-based artificial intelligence, and one benefit of the 2D-models is the performance of state-of-the-art image models, such as YOLO (Jocher and Qiu, 2024). In addition to methods using point cloud data as input, satellite imagery has also been used to classify tree species, for example by Gazzea et al. (2022), Wang et al. (2025).

3D-DL models operate directly on the point clouds, taking as input subsampled point coordinates and possible additional attributes, such as intensity and reflectance values. The pioneers of deep learning in the realm of point cloud understanding were PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b). Several studies have shown their ability to classify tree species (Liu et al., 2022a,b; Sun et al., 2023; Liu et al., 2022c; Taher et al., 2026; Puliti et al., 2025b). Newer 3D-architectures include PointMLP (Ma et al., 2022), 3D-CNNs (Tran et al., 2015), DGCNN (Wang et al., 2019) and PointTransformer (Zhao et al., 2021). All of these architectures have been utilised to classify tree species: Liu et al. (2022c,b) used PointMLP, 3D-CNNs were utilised by Liu et al. (2021, 2022c), Puliti et al. (2025b), DGCNN by Liu et al. (2022c), Puliti et al. (2025b), and PointTransformer in Sun et al. (2023), Taher et al. (2026). These studies show the prominence of 3D methods in tree species classification.

Recently, two benchmark studies on tree species classification have been published, namely one by Puliti et al. (2025b) and another by Taher et al. (2026). The former is carried out on the FOR-species20k[1] public dataset (Puliti et al., 2025a), and the other on the MS-ALS-SPECIES[2] dataset (Hyyppä et al., 2025, 2026), which is used as a reference dataset in this study. The FOR-species20k dataset contains more than 20 000 samples from 33 species, which are gathered by different methods (terrestrial laser scanning, MLS, unmanned ALS). In their study, Puliti et al. (2025b) found that image-based models outperformed certain point-based 3D models. In contrast, Taher et al. (2026) found that point-based 3D models, especially transformer-based architectures, outperform image-based models. Hence, it seems currently unclear whether the best classification performance can be achieved by projection-based methods or 3D-architectures.

Normal vector estimations are regularly used in computer vision and point cloud-based applications, for example in segmentation (Yang and Lee, 1999), shape modelling (Pauly et al., 2003) and point cloud registration (Serafin and Grisetti, 2015). Recently, Ferreira et al. (2024) combined normal vectors estimated from ALS data with satellite images to classify tree species, while Sha et al. (2026) used normal vectors as a part of a point-based architecture for tree species classification from point cloud obtained from varying sources. Both observed improvements in classification performance, with Ferreira et al. (2024) arguing that normal vectors capture species-specific leaf and crown geometries. This makes sense also from a biological viewpoint: local geometry may reflect species-dependent features, such as branching directions and the aforementioned crown geometry. In their study, Sha et al. (2026) found that in their architecture normal vectors yielded the most significant initial performance boost, and commenting that surface orientation is a key feature in local geometric awareness. Thus, we chose to investigate whether normal vectors provide discriminative information in the form of 2D-representations. Normal vectors were chosen as the descriptors of local geometry, as they capture local geometric information, while being relatively simple to compute, and descend naturally to RGB images, as explained in Section 2.4.1.

To address these gaps, this study investigates the potential of projection-based deep learning methods for tree species classification from point cloud data. We evaluate the suitability of integrating geometric information into projection images through surface normal vector estimation and assess how effectively this geometric cue alone can support species classification compared to radiometry-based approaches. Furthermore, we examine the influence of multispectral intensity information from multiple scanners and test model performance across both mobile and airborne laser scanning datasets.

The original contributions of this study are the following.

- We present NormalView, a projection-based deep learning method for classifying tree species from point cloud data. NormalView embeds local geometric information into two-dimensional projections, in the form of normal vector estimates, and uses the projections as inputs to an image classification network.
- We extensively inspect the effect of radiometric information on tree species classification via projection-based methods. We compare images coloured by intensity to projections done without intensity information, as well as examine the effect of multispectral intensity information compared to single channel intensity information.
- We inspect the aptitude of our model on ($\sim$6000 pts/m^2) MLS data, as well as high-density ALS data ($>$1000 pts/m^2). We compare the performance of our projection-based models on the ALS data with the results of state-of-the-art point-based 3D models (Taher et al., 2026), which are trained on the same ALS data.
- We release a high-density, close-range, and geometrically highly accurate eight species dataset of MLS tree segments containing 1915 samples (Korkeala et al., 2026). The data is available at 10.5281/zenodo.21039101.

[1] Available on Zenodo: https://doi.org/10.5281/zenodo.13255198
[2] Available on Zenodo: https://doi.org/10.5281/zenodo.17077256

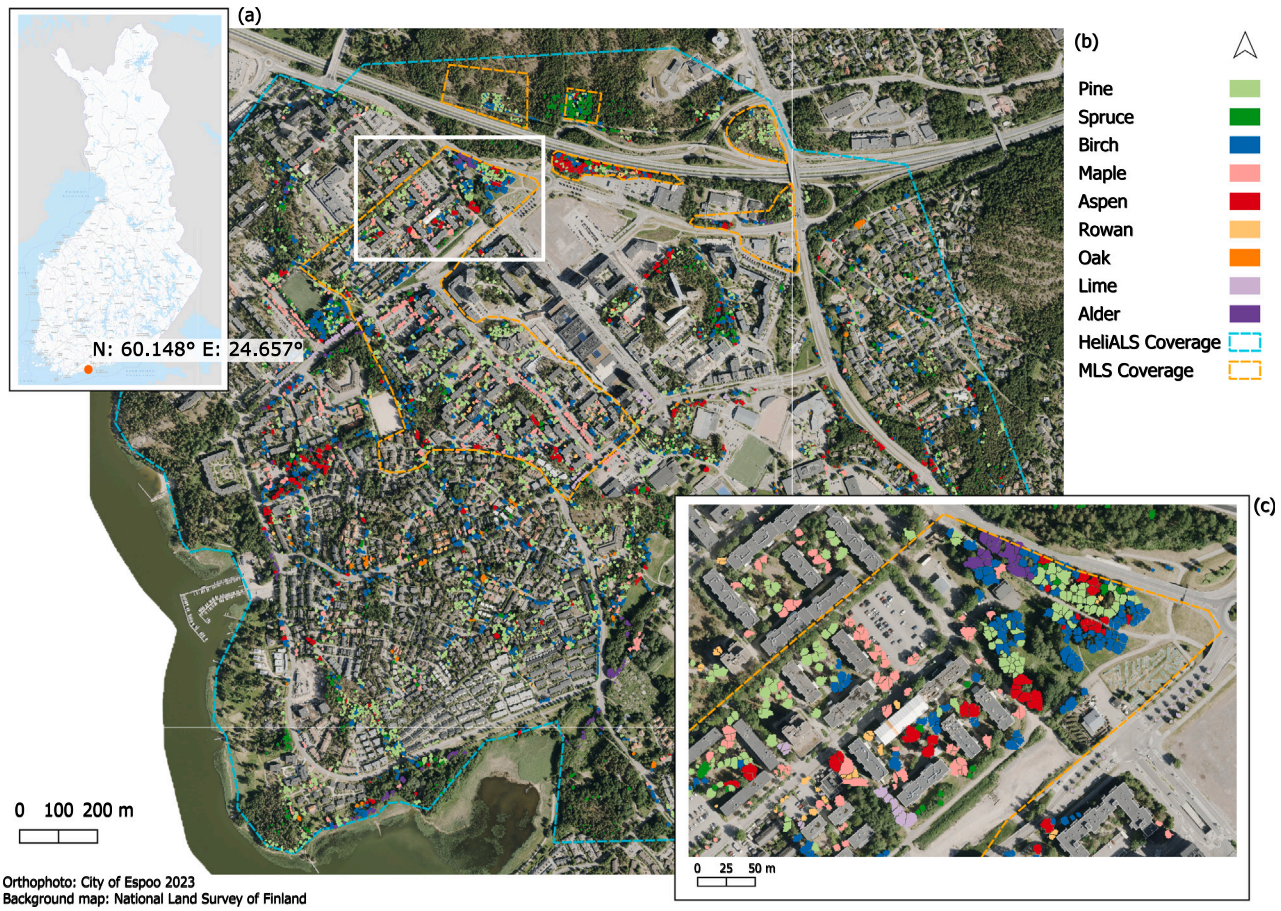


Fig. 1. Visualisation of the test site in Espoonlahti. (a): Location of the test site. (b): Orthophoto showing Espoonlahti. The tree segments, as well as the coverage of the different scanning methods, are visible. (c): A zoomed-in view of the orthophoto in (b), showing the tree segments in more detail.
Source: Figure adapted from Taher et al. (2026).

2. Materials and methods

2.1. Test site

The test site is located in Southern Finland, in Espoonlahti, Espoo (60.148° N, 24.657° E) and is shown in Fig. 1. The area contains a wide variety of trees in urban, semi-urban, and natural environments. The urban areas contain some planted trees, such as lindens, oaks, and maples. The sites with more naturally occurring trees consist mainly of pines, spruces, birches, and some aspens. The MLS data used in this study contains eight species: pine (*Pinus Sylvestris*), spruce (*Picea sp.*), birch (*Betula sp.*), maple (*Acer platanoides*), aspen (*Populus tremula*), rowan (*Sorbus sp.*), alder (*Alnus sp.*) and lime (*Tilia sp.*). Alder was not used in the MLS models, as there were too few labelled samples. The ALS data contains nine species: oak (*Quercus robur*) and the eight aforementioned species.

2.2. Measurements

The MLS data was collected using a FARO Orbis laser scanner (FARO Technologies Inc., Lake Mary, Florida, USA). The Faro Orbis is equipped with a Hesai XT32 lidar (Hesai Technology Co., Ltd., Shanghai, China). The measurements were conducted on the 3rd and 11th of June 2024. 19 measurements were made in total, with a single measurement lasting 7–20 min. The MLS coverage can be seen in Fig. 1. The laser scanner was mounted on a backpack. The key scanner properties are presented in Table 1.

The ALS data was gathered in the summer of 2023 by a helicopter equipped with three Riegl (Riegl GmbH, Horn, Austria) laser scanners: VUX-1HA, miniVUX-1DL, and VQ-840-G. The different scanners operate on different wavelengths, see Table 1. The flight altitude of the helicopter was approximately 100 m. Further information on ALS data collection is available in Taher et al. (2026), Hyyppä et al. (2025).

The field reference was collected using a browser-based solution by the employees of the Finnish Geospatial Research Institute (FGI) in 2024 and 2025. The browser-based application allowed for users to see the segmented trees, overlaid on an orthophoto of the area. With the application, the users could identify the species of the tree on the field, as well as add notes. The application was also used to verify and correct initial labels. The species of some planted trees were identified from the database of City of Espoo (2024).

2.3. Data preprocessing

The preprocessing of the ALS data was conducted on RiProcess (Riegl) and TerraScan (Terrasolid Ltd., Helsinki, Finland) software. Importantly, the data used in this study contains intensity information from each of the three channels for each point. This was achieved by using nearest-neighbour interpolation to fill in missing intensity values. Reflectance was used as the "source" for the intensity in RiProcess, i.e. the intensity values were distance-adjusted. The trees were segmented using a watershed segmentation-based algorithm introduced first in Yu et al. (2010) and tested in Kaartinen et al. (2012), and called Local Maxima Finding (FGI_LOCM). In addition, noise points were removed using a statistical outlier filter (Rusu et al., 2008) and ground points identified with a cloth simulation filter (Zhang et al., 2016). For further information on data processing and reference data

Table 1
Scanner properties.

	ALS			MLS
Property	VUX-1HA	miniVUX-1DL	VQ-840-G	Hesai XT32
Divergence V × H (mrad)	0.5	0.5 × 1.6	1	0.7 × 1.7
Pulse Repetition Rate (KHz)	1017	100	200	20
Range (m at N% reflectivity)	85 (10%)	140 (20%)	150 (20%)	80 (10%)
Wavelength (nm)	1550	905	532	905
Beam diameter at ground[a] (mm)	50	50 × 160	100	7 × 17

[a] At a 100 m (ALS) and 10 m (MLS) range from scanners.

Table 2
Species distribution and model training splits for the different data acquisition methods.

	Data type					
	ALS			MLS		
Species	Train	Test	Total	Train	Test	Total
Majority species						
Pine (Pinus Sylvestris)	1864	465	2329	639	116	755
Spruce (Picea sp.)	537	134	671	248	62	310
Birch (Betula sp.)	1224	307	1531	309	65	374
Maple (Acer platanoides)	540	136	676	188	46	234
Minority species						
Aspen (Populus tremula)	378	94	472	87	21	108
Rowan (Sorbus sp.)	163	41	204	48	9	57
Oak (Quercus robur)	53	13	66	–	–	–
Lime (Tilia sp.)	126	32	158	52	14	66
Alder (Alnus sp.)	104	27	131	–	–	11[a]
Total	4989	1249	6238	1582	333	1904

[a] Not used in MLS models.

collection, we refer the reader to Taher et al. (2026), Hyyppä et al. (2025, 2026).

The raw MLS data from the scanner was first processed by the FARO Connect software, which applies Faro's 3D SLAM algorithm. The point clouds were then randomly subsampled to a spatial resolution of 2 cm, where the minimum distance between any two points is at least 2 cm. Noisy points were removed using a Statistical Outlier Removal -filter (SOR filter, Rusu et al. (2008)). The SOR filter evaluates the average distances to k neighbours for each point and removes points for which the average distance is greater than the average distance of all points plus n times the standard deviation of the average distances. We used $k = 10$ and $n = 2$. Then, a Cloth Simulation Filter (Zhang et al., 2016) was applied to the point cloud, to identify the ground points. The measurement sites contained urban elements, such as buildings, cars, and traffic signs, which were manually removed from the point clouds. At this stage, the point clouds contained only the trees. The preprocessing steps were performed using open source CloudCompare software (CloudCompare development team, 2024).

Individual tree segmentation of the MLS trees was done with the TreeIso algorithm (Xi and Hopkinson, 2022), which is based on a cut-pursuit algorithm for weighted graphs (Landrieu and Obozinski, 2017). The segments produced by TreeIso were manually checked, and unsatisfactory segments were manually corrected using CloudCompare.

To compare the segmented MLS trees with the reference data, they need to be transformed from the local coordinates of the scanner to the global ETRS-TM35FIN coordinate system. To achieve this, we calculated a rigid transformation between the coordinates of each MLS-scan, x_{local}, and global ALS coordinates based on anchor points, such as traffic signs, that were found in both point clouds. We calculated an optimal transformation $x_{\text{global}} = Rx_{\text{local}} + t$, where R is a rotation matrix, t a translation vector, and x_{global} the new transformed coordinates, that minimised the squared distance between the ALS and MLS anchor points.

After transforming the points to the global coordinates, segmented trees were matched with the reference data. We adopted the same strategy as in Hakula et al. (2023), and utilised a Hausdorff distance based closest neighbour algorithm, as presented in Yu et al. (2006). Two points (trees), $a \in A$ and $b \in B$, in different sets (measurement data and reference data), A and B, were considered a match if a is the closest point to b in A, and b is the closest point to a in B. In addition, the distance between the matched points was required to be less than 3 m for them to be considered a match. The reference data for the MLS data is the same as for the ALS data.

In total, we arrive at 1915 MLS segments and 6238 ALS segments. The species-wise distributions in the data can be seen in Table 2. The four majority species (pine, spruce, birch, and maple) constitute 87.4% of the MLS dataset, and 83.5% of the ALS dataset. The difference between the data from the different acquisition methods is visualised in Fig. 2, where the left tree (MLS) has a much higher point density, and especially the trunk and branch structure is clearer. The tree on the right (ALS) on the other hand has comparable, or even more refined, canopy structure, due to it being scanned from above, as opposed to the MLS scans. In the MLS data, there are some trees which were captured by multiple different measurements due to overlapping measurement paths. As distinct scans are inherently different, and hence contain useful unique data, we included all viable scans of such trees in the dataset. There are 178 trees which are present in two segments and 11 trees which are present in three segments. To avoid bias in the model testing, we ensured that all such trees belong to the training set and that the models were not tested on such trees. For each point cloud representing a tree, we evaluated approximate coordinates of the tree trunk. The trunk coordinates were evaluated using a heuristic algorithm, which bins the points in x- and y-directions, and calculates a score for each bin based on point density, vertical point spread, vertical gaps, and height. The mean coordinates of the bin with the highest score are the approximate trunk coordinates.

2.4. Methodology

From each tree point cloud, we created projection images by four distinct methods: (1): an orthographic projection with intensity values as colours (OPI), (2): an orthographic projection with RGB-colours coming from normal vector estimations, denoted as NormalView (NV), (3): an orthographic projection with the colour values corresponding



Fig. 2. The same pine tree as captured by (a) the backpack mobile scanner, and (b) ALS. The point counts of the segments are 399 239 and 94 586 respectively. The MLS point cloud has been coloured by gamma corrected ($\gamma = 1.15$) single-channel intensity values (wavelength = 905 nm), which were first mapped to RGB colours by a colourmap. The ALS point cloud has been coloured by gamma corrected ($\gamma = 4$) intensity values (RGB colour channels: red = 1550 nm, green = 905 nm, blue = 532 nm). The colours are purely for visualisation purposes.

to linear distance from the viewing plane (depth), and (4): an orthographic projection where each pixel is either black or white (WOP), which is in essence a silhouette image. Further explanations of the colouring schemes are in Section 2.4.1. The models trained on WOP-images serve as a baseline, to see how well the trees can be classified with only projection images, without any radiometric input. The depth images are an established method in the field (see e.g. Goyal et al. (2021)), and serve as geometry-based baseline. With the projected images, we trained a classification model that takes images as input. To this end, we used the YOLOv11x-cls[3] architecture, as provided by Ultralytics (Ultralytics Inc. Frederick, Maryland, USA) (Jocher and Qiu, 2024).

From the MLS data, we create images according to the four methods mentioned above. From the ALS data we create two image datasets: one with images from seven species to compare the models trained on ALS data more directly with the models trained on MLS data, and the other with images from the full dataset with nine species, to compare our models with the benchmark in Taher et al. (2026). Both ALS datasets contain images created by the OPI, NV, and WOP methodologies outlined above. We analyse the viability of the image creation approaches separately for both the seven and nine species datasets. Finally, we combined the MLS and ALS datasets into a single combined dataset, and created images according to the OPI, NV, and WOP methods mentioned above. The purpose of the combined dataset is to analyse the viability of the proposed method on data from varying acquisition sources.

2.4.1. *Image creation*

For all colouring schemes, the orthographic projection images are created by projecting the 3D points of the tree onto a projection plane. The pixel coordinates are obtained by finding the orthogonal, or perpendicular, projection of the point onto the plane. The projection plane is the xz-plane, and hence the projection corresponds to extracting the x- and z-coordinates of the points. The x- and y-coordinate axes correspond to easting and northing in the georeferenced data, and the z-axis to the upwards direction. The projections are taken from multiple angles, as the silhouette of the tree varies depending on the viewing

[3] https://github.com/ultralytics/ultralytics



(a) A silhouette image of a birch

(b) A sliced silhouette image of a birch

(c) Images of the same rowan, scaled down to different sizes.

Fig. 3. (a)–(b) Two examples of images used in training the models. The images are WOP images created from the same birch from the same angle. (a) All the points are used for the projection. (b) Points nearest to the "viewing plane" are removed to reduce occlusion on the trunk and inner branches. (c) Images in different resolutions of the same rowan. The resolution is achieved by bilinearly interpolating the image of size 1024 × 1024.

angle. The projections from different angles are achieved by rotating the point cloud around the z-axis and projecting onto the xz-plane. Prior to taking the projections, we centred the tree in the origin, by translating the approximated coordinates of the trunk to the origin. In this study, we used five different viewpoints, corresponding to rotations of $0°, 72°, 144°, 216°$ or $288°$ around the z-axis. In the case that multiple points have the same projected coordinates, and would hence overlap in the projection image, the point nearest to the projection plane was used as the pixel source.

From each angle, we created two images: the full projection containing all points and a sliced image in the depth direction to gain insight of the structure of the tree closer to the trunk. This information includes the growing direction of branches, for example. To generate the sliced images, we excluded all points in the rotated point cloud whose y-coordinate satisfied $y > t_y + k$, where t_y is the estimated y-coordinate of the tree trunk, and k is a used-defined threshold. The remaining points were used to create the image. In this study, we used $k = 0.7\,\text{m}$. This selection provides useful structural information of the insides of the tree, while being robust against errors in the trunk position estimation. The differences between the full images and the sliced images can be seen in Figs. 3(a) and 3(b). Notably, the trunk and inner branches are more visible in the sliced image than in the full image. Consequently, we created ten 512 × 512 images of each tree per colouring scheme, which are discussed below. In addition, to test the effect of image size on classification accuracy, we created images of size 1024 × 1024 from the MLS data. Multiple images with varying size were produced from the full image via bilinear interpolation, see Fig. 3(c), and performance was tested on each resolution individually.

In the OPI images, we used the intensity values of the points as a single-channel colour input. The intensity values provided by the scanners belong to the interval $[0, 65\,536]$ and were normalised by dividing by the maximum value, 65 536, since the absolute scale of the intensity can reveal relevant information about the tree. The scaled intensity values were then linearly scaled to the interval $[0, 255]$, and rounded to integers. For the MLS data, these values were then used as input for each of the three 8-bit colour channels in an RGB-image, resulting in a greyscale image. The ALS data contains information from three intensity channels, from which we created intensity-coloured images by using different permutations of the intensities. We created greyscale intensity images from each individual channel as described for the MLS data, and also three types of images containing two different channels, and finally images using all three channels. For the images containing two different channels, the third colour value in the images was set to zero. For the images containing data from all three channels, we performed a similar normalisation as above and used the three channels as input to the red, green, and blue colour channels. The various colouring schemes are illustrated in Fig. 4.

For NV images, we first estimated a normal vector for each point. The normal at each point was estimated using the `estimate_normals` function of the Open3D library,[4] which determines a local

[4] https://www.open3d.org/docs/0.7.0/python_api/open3d.geometry.estimate_normals.html

tangent plane from the N nearest neighbours, identified using a KD-tree, using principal component analysis. No upper limit for the search radius was used. The unit normal vector of this plane is then chosen as the normal vector at the point. In this study, $N = 20$ was used for all experiments. This allows for enough points so that the plane is approximately tangential to the surface at the point, while taking points from a small enough area so that we preserve local geometry. In the ablation studies it was found that the choice of $N = 20$ provides the best balance between MLS and ALS data, although the effect of the number of nearest-neighbours used is rather minuscule, as seen in Table 8. We oriented the normals to point outward in a standardised manner, so that the spatial angle between the normal vector and the vector from the point to the estimated trunk is greater than 90°. Specifically, if the angle between the estimated normal vector and the vector from the point to the estimated trunk is less than 90°, we switched the sign of the normal vector. Subsequently, the components of the normal vector were used as the red, green, and blue colour channels, after a linear scaling to the interval $[0, 255]$, and rounding to integer values.

To inspect how Normalview fares against established geometry-based methods, we also generated depth-based images from the tree point clouds. We followed the protocols outlined in Allen et al. (2022) and the DetailView method in Puliti et al. (2025b) for creating the depth images. For each tree, the point coordinates were first median-centred and uniformly scaled to fit within $[-1, 1]^3$. The resulting point cloud was rotated around the vertical axis to obtain the same set of azimuthal views as the other projection methods. For each view, the x- and z-coordinates were orthographically projected onto the image plane, while the coordinate along the viewing direction was used as the pixel value. In other words, the linear distance (depth) from the viewing plane was used as the colour value. When multiple points projected to the same pixel, the depth of the point closest to the viewing direction was retained. Empty pixels were assigned a value of zero, and the resulting depth values were linearly mapped to the 8-bit range [0,255], and rounded to integers. For each viewing angle, an additional sliced projection was generated by the same slicing procedure as for the other projection modes. Unlike the original DetailView configuration, which combines side, top, bottom, and close-up trunk projections, only the side views were used here so that the experiments isolate the effect of the pixel representation while keeping the number and orientation of projections consistent across methods. The code to create the projection images was implemented following the public DetailView repository[5] (Puliti et al., 2025b).

In the WOP images, a pixel was simply coloured white, if there is a projected point corresponding to it. For the intensity and normal images, we applied a Gaussian convolution kernel, to make the images smoother. The convolution kernel is 3×3 with a standard deviation of 0.85. No filter was applied to the WOP images. In line with Allen et al. (2022), Puliti et al. (2025b), no filter was applied to the depth images either.

Projecting the points onto a two-dimensional plane inherently causes information loss, as we lose one degree of spatial freedom. Consequently, we lose intrinsic structural relationships in the data. The projection tactics described in this section attempt to combat this: multiple viewing angles provide a more thorough view of the tree than one singular angle, and the slice images allow the models to "see inside the trees", as can be seen in Fig. 3. The key idea behind the normal projections is to embed three-dimensional structural information into the images via colour. Normal vectors describe the local three-dimensional geometry of each point. Another way this could be achieved is by using eigenfeatures derived from the local structural tensor of each point, as is done by Blackburn et al. (2025).

Additionally, we examined the effect of image size in model training on model performance. Image size refers to the parameter passed on to the YOLO model in the training phase. YOLO scales the input images to the set image size by applying bilinear interpolation. All images in this experiment were created to have size 1024×1024, and were scaled using the bilinear interpolation to the set size when training the model. Ultralytics implements the scaling using OpenCV's resize-function (Bradski, 2000). We tested the effect of image size only on the intensity-coloured images. The differences in perceived visual quality are large for the smaller image sizes, as demonstrated by Fig. 3(c). We see that a significant portion of visual details are lost in images sized 128×128 and smaller.

2.4.2. Model training

As the classification model, we used the YOLOv11x-cls architecture, provided by Ultralytics (Jocher and Qiu, 2024). Although the YOLO family is primarily associated with object detection, dedicated classification variants have been developed and evaluated on large-scale image classification benchmarks. YOLOv11 builds upon the preceding YOLO models, starting from YOLOv1 (Redmon et al., 2016). For an evolution of the YOLO architectures, see Terven et al. (2023). YOLOv11 employs a convolutional backbone–neck architecture designed for hierarchical feature extraction. The backbone uses C3k2 blocks and spatial pyramid pooling, followed by the C2PSA attention module, while the feature-pyramid-based neck combines information from multiple spatial resolutions. In the classification variant used here, the detection head is replaced by a classification head that produces image-level class probabilities. The source code for the YOLOv11 models is available at the Ultralytics Github repository.[6]

The choice of architecture was mainly motivated by a favourable balance between computational requirements and classification performance, as well as the usability and performance of the training and inference frameworks provided by Ultralytics. YOLO models have also been used in tree species classification, for example in the benchmarking studies by Puliti et al. (2025b) and Taher et al. (2026), with reasonably high performance, compared to other CNN backbones. We emphasise however, that the proposed methodology is independent of YOLO, in the sense that any image classification network could be used as the downstream classifier. If one has access to sufficient hardware, the Vision Transformer (ViT) (Dosovitskiy et al., 2021) could be used an alternative classifier. The smallest ViT model contains 86 million parameters, and would have been computationally too demanding for the hardware used in this study. For comparison, the used YOLOv11x-cls model has 28.6 million parameters, requires 13.6 billion floating point operations per second (FLOPs), and achieves an overall accuracy of 79.5% on the ImageNet dataset (Deng et al., 2009). Other alternative CNNs which could be used include the various variants of EfficientNet (Tan and Le, 2020), which have between 5.3 and 66 million parameters, and achieve an overall accuracy between 77.1% and 84.3% on the ImageNet dataset, and the different ResNet (He et al., 2015) models.

The training parameters can be seen from Table 3. Parameters not mentioned in the table are left to the default values. We divided the data so that approximately 80% of the trees are used for model training, and 20% are used for testing. Taher et al. (2026) use a 5-way k-fold split in their studies. The train/test split we used for the ALS data corresponds to the train/test split of the first fold in Taher et al. (2026). The stratified tree-level train–test split, the species counts of which are visible in Table 2, was used throughout the experiments, such that all observations from an individual tree were assigned exclusively to either the training or test set. This type of single-split evaluation is commonly used in tree-species classification studies, including Puliti et al. (2025b), Liang et al. (2014), Marinelli et al. (2022), Tockner et al. (2025). The model weights were initialised to the weights of the model pretrained on the ImageNet dataset.

[5] https://github.com/JulFrey/DetailView

[6] https://github.com/ultralytics/ultralytics

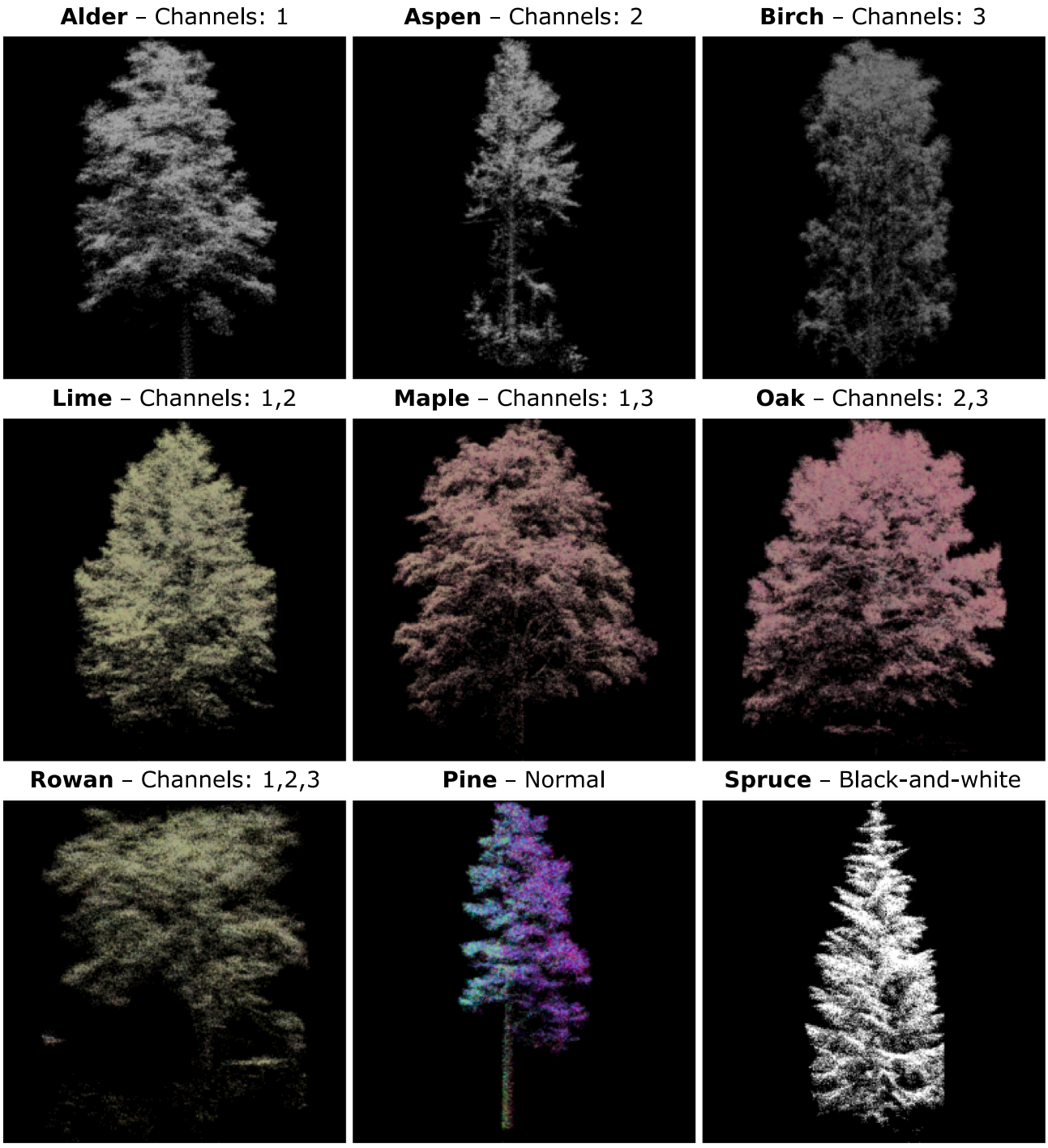


Fig. 4. Example images of the nine different tree species and image colouring methods used in model training. All samples are from the ALS data. For explanations for the colours, see Section 2.4.1.

For each tree, ten projection images were generated during training, corresponding to five viewing directions and both the full and sliced projections. These images were treated as individual training samples, with the tree-level class label assigned to each image. During inference, 50 images were generated for each tree by using 25 viewing directions and both full and sliced projections. The image-level class probabilities were summed across all 50 images, and the species with the highest aggregated probability was assigned as the tree-level prediction. Using a high number of viewing angles in the inference phase increases model robustness, at the cost of computational efficiency. In the training phase, we used early stopping to prevent overfitting and to save time. Early stopping means stopping training prior to reaching the set number of epochs, if no improvement in validation metrics is observed for a certain number of epochs. Ultralytics provides this functionality via the `patience` parameter, which was set to 30.

Image augmentations in the training phase of image classification models are recognised to be of significant benefit to model robustness and to avoid overfitting (Shorten and Khoshgoftaar, 2019). The augmentations can include geometric transformations (translations, scaling, and horizontal flips), colour transformations (hue, saturation, and brightness adjustments), blocking parts of the image, and combining multiple images into one. During training, we used the RandAugment policy (Cubuk et al., 2020), implemented by Ultralytics, which in turn

Table 3
Parameters used in model training. For the entries for image and batch size, the first one refers to ALS models, and the second one to MLS models. The values inside parentheses for CPU, RAM and GPU refer to the machine used to train models with image sizes 768 and 1024 in the image size tests.

Parameter	Value
Epochs	400
Image size	256 / 512
Batch size	16 / 5
Optimizer	AdamW[a]
lr0[b]	0.0005
Dropout	0.15
Patience[c]	30
Momentum[d]	0.8
Learning rate scheduler	Cosine[e]
Augmentation policy	RandAugment
Python	3.11.
PyTorch	2.7.0.
CUDA	12.4.
RAM	32 GB (256 GB)
CPU	Intel Core i7-12850HX(Intel Xeon Gold 6234)
GPU	12 GB Nvidia RTX A3000(24 GB Nvidia Quadro RTX A6000)

[a] See Loshchilov and Hutter (2017).
[b] Initial learning rate.
[c] Number of epochs to wait before stopping training early if performance does not improve.
[d] β_1 parameter in AdamW; controls incorporation of past gradients.
[e] See Loshchilov and Hutter (2016).

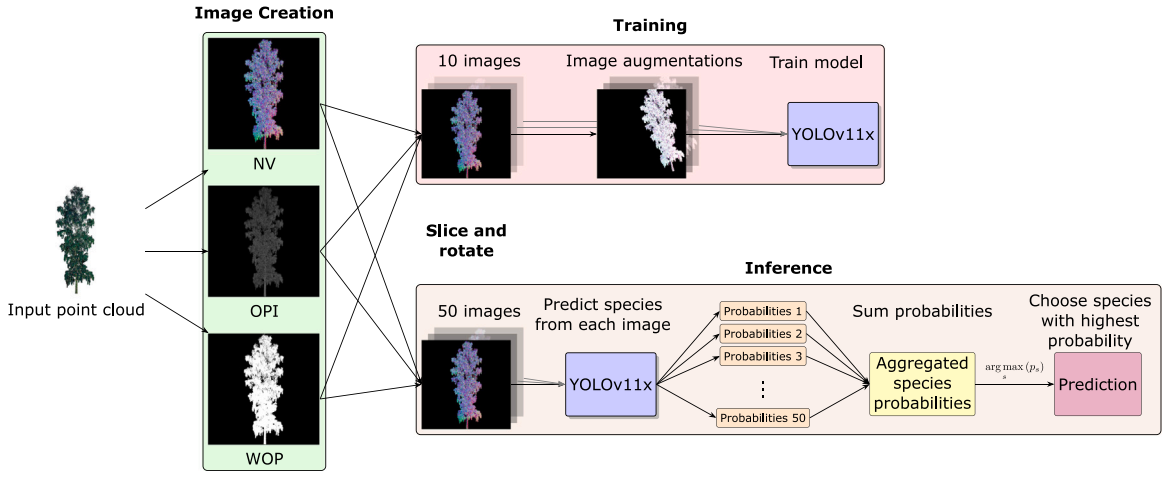


Fig. 5. A flowchart representing the workflow of the used methods. First, we create images from the input point cloud (green). Then, we either train a model using the created images (red), or use a trained model to predict the species of the input (brown).

is based on the PyTorch[7] implementation. No additional augmentation was applied during validation or inference. The used software versions and hardware specifications can be seen in Table 3. In the image size tests, models using sizes 768 and 1024 were trained on a more powerful machine, due to high VRAM consumption.

The workflow presented in Fig. 5 illustrates that the proposed image creation procedure is independent of the downstream classifier and could therefore be combined with other image-based architectures. From the input data, we create images according to the methods described in Section 2.4.1. From these images, we can either train a new model, or predict the species of the input point cloud using a trained model. Notably, in the training phase we create 10 images per tree point cloud, but when doing inference, 50 images are created. All code used for projection generation, model training, and evaluation is publicly available at Korkeala (2026), and the model weights are available at Korkeala et al. (2026), enabling the experiments and reported results to be reproduced.

[7] https://docs.pytorch.org/vision/main/generated/torchvision.transforms.RandAugment.html

2.5. Reference method: Point transformer

In order to benchmark the proposed NormalView method against existing tree species classification methods, we selected the Point Transformer (Zhao et al., 2021) as a baseline due to its high accuracy in this task (Taher et al., 2026). For consistency, our experimental setup closely followed that of the individual tree species classification benchmark by Taher et al. (2026).

The baseline experiments included four scenarios: (1) Only x-, y- and z-coordinate information (geometry), (2) geometry and intensity values, (3) geometry and normal vector information, and (4) geometry, intensity and normal vector information. The normal vectors for each point were computed equivalently to the NormalView approach described in Section 2.4.1.

Prior to training and inference the individual tree point clouds were mean centred, and the overall scale was normalised by dividing all spatial coordinates by a scaling constant that effectively fit the point cloud inside a unit sphere while preserving the natural aspect ratio. Likewise, the intensity values were normalised to reside in the interval $[0, 1]$ by scaling the original 16-bit intensity values by a factor of $1/2^{16}$. Random sampling was applied to standardise the point density across

samples. The target point density of 1500 points/m^2 was obtained by sampling points from the original point clouds with respect to the effective tree canopy area. The effective canopy area was defined as the area of convex hull enclosing the orthographically projected tree points on the Easting-Northing (x- and y-coordinates, respectively) plane. The target point density in the baseline experiments was selected as a trade-off between the available GPU memory capacity and point density performance scaling law established in Taher et al. (2026). The Point Transformer experiments were also conducted using a constant point count of 8192 points for each tree. The results for constant point count experiments are available in Appendix B.

The Point Transformer model was trained in each experiment for 200 epochs and the model that obtained the highest overall accuracy in the validation set was used to evaluate the baseline performance metrics. For comparability, the test set split was identical to the NormalView experiments. The validation set was selected among the training and validation samples by stratified sampling, such that 10 percent of the samples in the training and validation set were used for validation and 90 percent of the samples were available for training. The model was trained using regular cross-entropy loss function with AdamW optimizer (Loshchilov and Hutter, 2017). The learning rate was initially set to 0.001 and dynamically adjusted using a cosine annealing scheduler (Loshchilov and Hutter, 2016) across $T_{\max} = 200$ epochs down to a minimum learning rate of 10^{-6}. Training time data augmentations consisted of random translation (uniform distribution in the interval $[-0.1, 0.1]$), isotropic scaling (uniform distribution with scaling factor between 0.8 and 1.2), rotation (uniform distribution with rotations limited to $\pm$ 10 degrees along each principal axis) and jittering of the x-, y- and z-coordinates (normally distributed with $\mu = 0$ and $\sigma = 0.005$). Additionally, the point-wise intensity values were randomly scaled (uniform distribution with scaling factor in the interval $[0.99, 1.01]$) and zero-mean Gaussian noise ($\sigma = 0.005$) was added to the values. Batch size 16 was used throughout the experiments.

Training and evaluation of the baseline model was carried out on a computer equipped with an Intel Xeon Gold 6244 CPU, 512 GB of RAM, and two NVIDIA RTX A6000 GPUs with 48 GB of VRAM each.

2.6. Experiments

Below are listed the experiments that were conducted in this study.

- Comparison of models trained on OPI, NV, and WOP images created from MLS data.
- Comparison of models trained on OPI, NV and WOP images created from ALS data using seven species and the full nine species data set.
- Comparison of models trained on OPI, NV and WOP images created from a mixed dataset consisting of both the MLS and ALS datasets.
- Conducting comparative studies using depth images and the Point Transformer architecture on the MLS dataset.
- Conducting ablation studies on the effects of point density, nearest neighbour parameter in normal vector estimation, and slice images on model performance.
- Evaluating the effect of image size on model performance.
- Evaluating the computational efficiency of the image creation pipeline of the proposed NormalView method.

To compare the models using different image types, we trained models with the same initial parameters and evaluated their performance using the metrics listed in Section 2.7. The OPI images created from ALS data contain all possible seven permutations of different intensity channels as described above. The motivation behind these tests is to evaluate the performance of NormalView against models using radiometric information, as well as to inspect the differences between models using intensity information from a single source compared to models using data from multiple sources.

Additionally, we examined the effect of image size in model training on model performance. Image size refers to the parameter passed on to the YOLO model in the training phase. The input images are scaled to the set image size by applying bilinear interpolation. All images were created to have size 1024×1024, and were scaled using the bilinear interpolation to the set size when training the model. Ultralytics implements the scaling using OpenCV's resize-function (Bradski, 2000). We tested the effect of image size only on the intensity-coloured images. The differences in perceived visual quality are large for the smaller image sizes, as demonstrated by Fig. 3(c). We see that a significant portion of visual details are lost in images sized 128×128 and smaller. However, the visual differences are not as prevalent between the larger images. The aim was to find out how much the image size parameter affects model performance, and what is the optimal image size.

When training on the mixed MLS + ALS dataset, we used image size 256×256 in training. As discussed above, the images are scaled to the set resolution before being passed on to the model. Thus, nothing is done to the ALS images, as they already have resolution 256×256, but the MLS images are scaled down from their 512×512 resolution. Further, on the mixed dataset we used the data from scanner 2 (miniVUX-1DL) as the intensity data for the ALS segments. The reason for this choice is that out of the three models using single-channel intensity data on the nine species ALS data, the one using channel 2 data performed the best, see Table 5.

2.7. Model evaluation

To evaluate the trained models, we utilised four macro-level metrics: overall accuracy (OA), macro-average accuracy (MAA), macro-average F_1-score (MAF_1), Cohen's kappa (K) (Cohen, 1960), and three species-level metrics: precision (P), recall (R) and species-wise F_1 (F_1). We denote true positives by TP_s, true negatives by TN_s, false negatives by FN_s, and false positives by FP_s, where s denotes the species.

Overall accuracy is defined as the ratio of correct predictions over all test samples

$$\mathrm{OA} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathbb{1}(\hat{y}_i = y_i),$$

where N_{test} is the number of samples in the test set, $\mathbb{1}$ is the indicator function, $\hat{y}_i$ is the predicted label of the ith segment, and y_i is the correct label of the ith segment. OA is a straightforward metric for classification models, but can be a poor indicator of performance, for example if the underlying class distribution is imbalanced (He and Garcia, 2009).

The species-wise precision, recall, and F_1-score are defined as

$$\mathrm{P}_s = \frac{\mathrm{TP}_s}{\mathrm{TP}_s + \mathrm{FP}_s}, \quad \mathrm{R}_s = \frac{\mathrm{TP}_s}{\mathrm{TP}_s + \mathrm{FN}_s}, \quad \mathrm{F}_{1,s} = \frac{2\mathrm{P}_s\mathrm{R}_s}{\mathrm{P}_s + \mathrm{R}_s}$$

To gain insight of model performance in the presence of the aforementioned imbalance in the species distribution, we also calculated the macro-average accuracy and macro-average F_1-score for each model. These metrics are highly informative, since both of the used datasets have imbalances between the species counts. The macro-averaged metrics give equal weight to each species, and thus they are sensitive to poor performance on the less represented species. The MAA is defined as the arithmetic mean of species-wise recalls, while the MAF_1 is defined as the arithmetic mean of species-wise F_1 scores.

$$\mathrm{MAA} = \frac{1}{|S|} \sum_{s \in S} \mathrm{R}_s, \quad \mathrm{MAF}_1 = \frac{1}{|S|} \sum_{s \in S} \mathrm{F}_{1,s}$$

where S denotes the set of all species in the dataset and $|S|$ the number of species in the set. The Kappa score is defined as

$$\mathrm{K} = \frac{p_o - p_e}{1 - p_e},$$

where p_o is the empirical agreement observed between the predictions and the ground truth and p_e is the proportion of samples agreed upon if the predictions were given by chance.

Table 4
Performance metrics of models trained on MLS data. Bolded values indicate the best performance in each metric.

Type	OA (%)	MAA (%)	MAF_1 (%)	K (%)
Intensity	94.6	91.6	91.7	93.1
Normal	**95.5**	**94.8**	**93.1**	**94.2**
Black-and-white	94.0	90.7	89.8	92.3

Table 5
Comparison of model performance metrics on ALS data for 7 and 9 species. Bolded values indicate the best performance in each metric respectively for 7 and 9 species. Channel 1 refers to intensity information provided by VUX-1HA, Channel 2 to miniVUX-1DL, and Channel 3 to VQ-840-G.

	7 Species				9 Species			
Type	OA (%)	MAA (%)	MAF_1 (%)	K (%)	OA (%)	MAA (%)	MAF_1 (%)	K (%)
Channels: 1	**93.6**	90.1	**91.1**	**91.5**	91.8	79.7	82.4	89.2
Channels: 2	93.2	89.5	90.5	91.0	92.2	80.3	83.1	89.8
Channels: 3	93.4	89.5	90.3	91.2	91.2	77.9	80.8	88.5
Channels: 12	93.1	89.6	90.5	90.9	91.8	83.2	85.4	89.3
Channels: 13	93.1	88.8	89.9	90.8	92.3	84.2	85.3	90.0
Channels: 23	93.2	88.3	89.4	91.0	92.1	82.3	83.6	89.6
Channels: 123	**93.6**	**90.6**	**91.1**	**91.5**	**92.5**	**84.9**	**86.8**	**90.2**
Normal	93.4	88.9	89.9	91.2	91.8	79.1	81.8	89.2
Black-and-white	93.5	89.5	90.5	91.3	91.6	79.5	82.9	89.0

Table 6
Performance metrics of models trained and evaluated on the combined MLS and ALS dataset. Bolded values indicate the best performance in each metric.

Type	OA (%)	MAA (%)	MAF_1 (%)	K (%)
Intensity	92.2	**78.1**	**81.3**	89.7
Normal	92.1	76.4	79.5	89.6
Black-and-white	**92.4**	77.2	81.0	**90.0**

As the models were trained using a single run, the reported metric values should be interpreted carefully, primarily as descriptive comparisons, rather than as evidence that small differences in metric signify statistical superiority.

3. Results and discussion

In this section, we evaluate the performance of models and provide an analysis of the results. In Section 3.1, we show the performance of the models trained on MLS data and seven species, on ALS data, both with seven and nine species, and on a mixed nine species dataset which is made up of the MLS and ALS datasets. To further evaluate the performance of our proposed method, we run comparative experiments using established methods in the field, namely depth-coloured images and Point Transformer, the results of which are in Section 3.2. The results of the ablation studies are presented in Section 3.3. Section 3.4 presents our analysis of the effect of image size on model performance, as well as an investigation of the computational efficiency of the proposed NormalView pipeline.

3.1. Model performance evaluation

The macro-level model performance metrics for the models trained on MLS data are presented in Table 4, and the results for the ALS models are visible in Table 5, for both seven and nine species. Table 6 contains the performance metrics for models trained on a mixed dataset, which contains both the MLS and ALS data. The species-level metrics are visible in Figs. 6 and 7, for the MLS and ALS models respectively, while Fig. 8 contains the metrics of the models trained on the MLS + ALS dataset. In the following, Channel 1 refers to intensity information provided by VUX-1HA, Channel 2 to miniVUX-1DL, and Channel 3 to VQ-840-G.

NormalView demonstrates the strongest performance on the MLS data, as can be seen in Table 4. It achieves the highest macro-level performance metrics across the board. The strong results in MAA and MAF_1 highlight the ability of NormalView to take into account even minority species in dense MLS data. This observation is supported by the species-wise performance presented in Fig. 6. NormalView achieves the highest F_1 for all species except for spruce and aspen.

The intensity-based model performs better than the black-and-white model, but the differences in performance are marginal. For the majority species, the species-wise performance is similar across the models. However, as Fig. 6 shows, the differences start to show for the minority species. The black-and-white model performs notably worse than the other two models on rowan and lime.

The overall metrics for species classification for the ALS data are slightly lower than on the MLS data, as illustrated in Table 5. This can likely be attributed to the higher point density and superior segment quality in the MLS data. The results on the seven species ALS data remain strong, however, with the OA (MAA) being 93.1% (88.3%) or higher for each model. The average F_1-scores and Cohen's kappas all also exceed 90%, except for the macro-F_1 scores for the normal model, and the models using intensity channels 1 and 3, and 2 and 3. On all the models, the MAA is noticeably lower than OA.

On the nine species dataset, the OAs (MAAs) are 91.2% (77.9%) or higher for each model. There is more notable separation between the OA and MAA for the models trained on nine species, when compared to the MLS models or models trained on seven species. On nine species, the models using multiple intensity channels perform more consistently than the others.

Overall on the ALS data, the model using all three intensity channels achieves the highest macro-level metrics, achieving an OA (MAA) of 93.6% (90.6%) on seven species, and 92.5% (84.9%) on nine species. On seven species, the model using only intensity channel 1 achieves similar performance in all other macro-level metrics, except having a 0.5% lower MAA than the three-channel model. On nine species, the model using intensity channels 1 and 3 achieves the second-highest values for the metrics, with the only notable difference to three channel model being a 1.5% lower MAF_1. The overall accuracies of all models are within a percentage point for seven species, and within 1.3 percentage points for nine species. There is a bit more separation in the MAA and MAF_1 metrics, which indicates that there are more

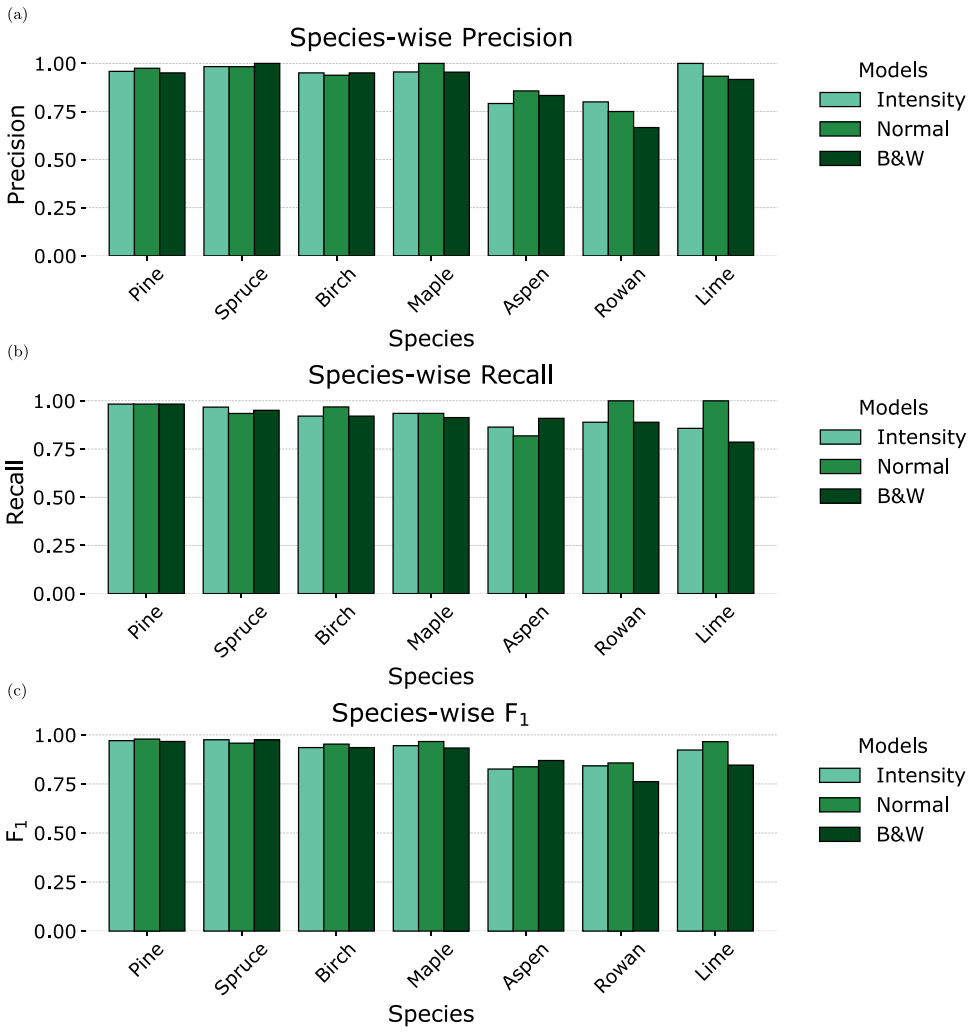


Fig. 6. Species-wise metrics of the models trained on MLS data. (a) Species-wise precision, (b) species-wise recall, (c) species-wise F_1 scores. B&W refers to the model trained on black-and-white images.

noticeable differences in how the models perform across species. This is particularly relevant, as both the MLS and ALS have imbalanced species distributions. The difficulty of predicting minority species can be seen in Fig. 7: the metrics are lower for aspen, rowan, and lime, and especially for oak and alder. On the majority species, especially on pine and birch, the species-wise metrics are extremely strong. The differences in OA and the macro-averaged metrics are at least partly attributable to the weaker performance of the models on the less represented species.

The model using all intensity channels has the most balanced performance across species. The models using two intensity channels perform extremely similarly on both the seven species and nine species datasets, and on the nine species dataset they outperform the single-channel and geometry-based models. This, and especially the adeptness of the three-channel model, indicate that multispectral information is beneficial on datasets with numerous species.

With the seven species dataset, NormalView and the black-and-white model perform comparatively with all the other models, with the MAAs being within two percentage points of the highest-performing model. On nine species, the models using multiple intensity channels consistently achieve higher macro-level metrics. Their performance is on an equal footing with the models using a single intensity channel. Their macro-level metrics are substantially worse than those of the three-channel model. However, the OAs (MAAs) of NormalView and the black-and-white model, 91.8% (79.1%) and 91.6% (79.5%) respectively, indicate strong performance, only slightly losing out to the best model (FGI-PointTransformer-DL-3D) in Taher et al. (2026), which is trained on the same data, and uses full radiometric information.

Table 6 presents the results of the models trained on the combined MLS and ALS dataset. The differences between the three projection methods are very small. All models achieve an overall accuracy of approximately 92%, with differences below 0.3 percentage points. Similarly, the macro-average metrics differ by less than two percentage points. The results are very similar to the results on the full ALS data in Table 5. Classification of the minority species, especially oak, and to a lesser degree rowan, lime, and alder, is again difficult, as suggested by Fig. 8. Across both the MLS and ALS datasets, the minority species have the lowest species level performance metrics. This suggests that the lower performance on them is not solely due to the acquisition modality, but also to some degree on the imbalance in the dataset.

Interestingly, the black-and-white model performs on par, if not better, than the intensity and NormalView models on the mixed data. This suggests that silhouette information captures structural characteristics that generalise well across both acquisition modalities. The silhouette information remains more consistent between the data acquisition methods, than the local geometry used by NormalView, or the intensity information. In contrast, intensity values and local surface geometry appear to be more sensor-dependent, making their advantages less pronounced when MLS and ALS observations are combined into a single dataset.

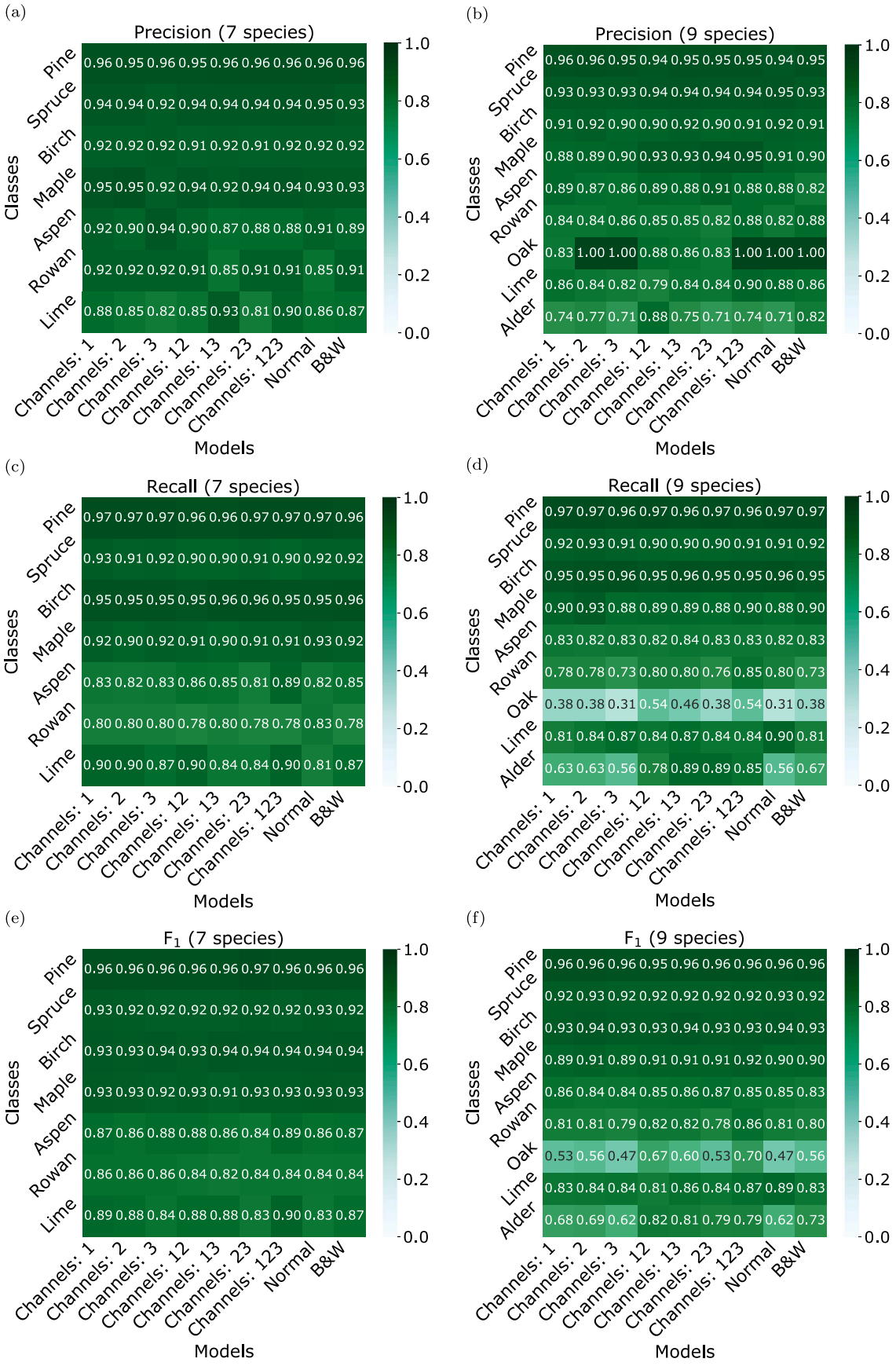


Fig. 7. Species-wise precision, recall, and F_1-scores for models trained on ALS data with 7 (left) and 9 (right) species: (a)–(b) precision, (c)–(d) recall, (e)–(f) F_1-scores.

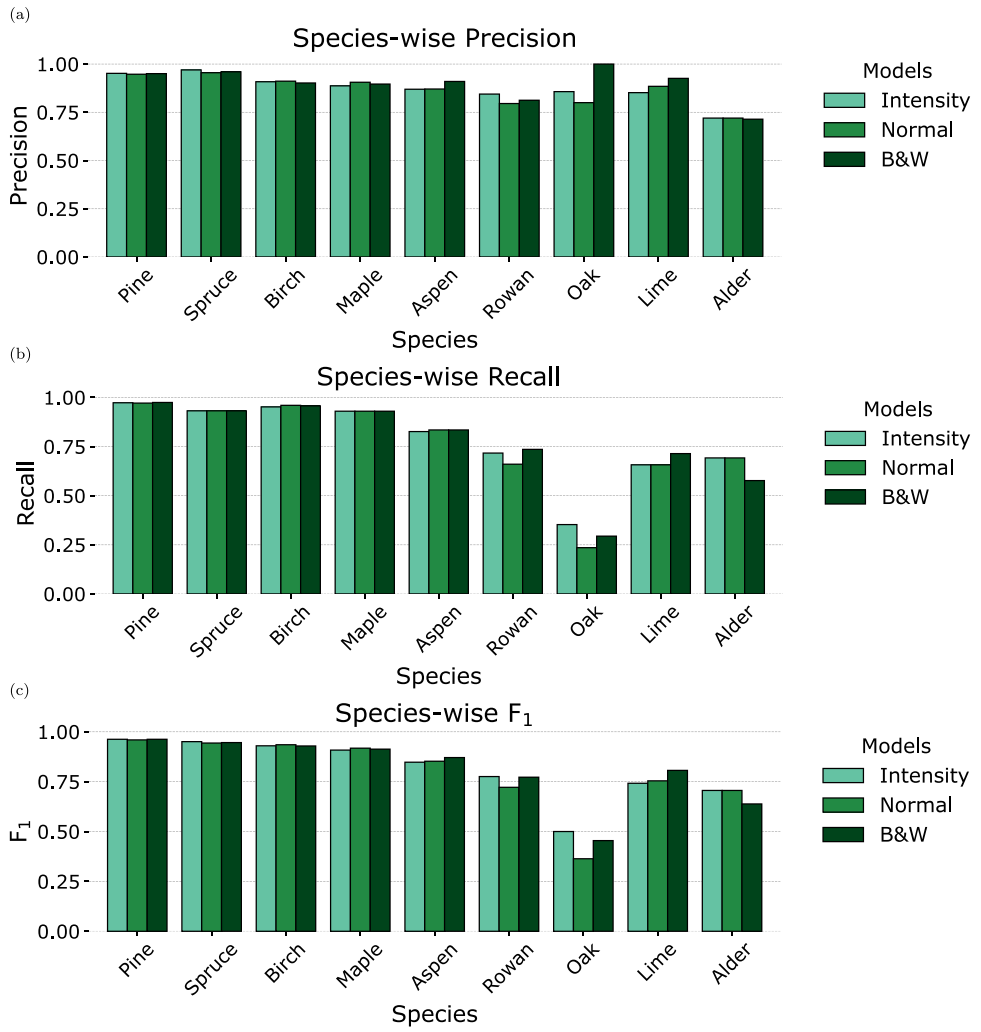


Fig. 8. Species-wise metrics of the models trained and evaluated on the mixed MLS + ALS data. (a) Species-wise precision, (b) species-wise recall, (c) species-wise F_1 scores. B&W refers to the model trained on black-and-white images.

3.2. Comparative experiments

To compare our proposed method with established methods in the field, we trained and tested models using depth images, as well as three Point Transformer models on the MLS data. In their comparison study, Puliti et al. (2025b) found that the best method DetailView, uses depth values to colour the projection images. On the other hand, Taher et al. (2026) found the Point Transformer model to be the best performer in their tree species classification study. Thus, we chose to compare NormalView with depth projection using the same protocol as used by DetailView, and with the Point Transformer architecture by using the normal vector representation of NormalView as additional point attributes. The performance metrics are visible in Table 7. The metrics and a brief discussion for the Point Transformer using point clouds downsampled to a constant point count (8192) can be found in Appendix B.

The results of the depth-based model are shown in Table 7. The depth model achieves an overall accuracy of 95.2%, which is comparable to that of the OA of NormalView, 95.5%. Its macro-average F1-score is also very similar, at 93.3% compared with 93.1% for NormalView. NormalView achieves a slightly higher macro-average accuracy (94.8% vs. 94.2%) and Cohen's kappa (94.2% vs. 93.8%). Thus, the results are extremely similar, and as such the results do not suggest a substantial difference between the projection methods, when it comes to species classification performance.

The strong performance of the depth model is nevertheless informative for evaluating NormalView. Both approaches encode geometric information into a two-dimensional representation, but they do so in fundamentally different ways. Depth represents the distance of the visible surface from the projection plane, and therefore primarily retains information about the three-dimensional arrangement of the points along the viewing direction. In contrast, the normal representation encodes the local orientation of the surface at each point. Consequently, the two representations provide different geometric information: depth describes where the surface is located, whereas surface normals describe how the surface is locally oriented.

The similar performance of the two models suggests that both cues contain sufficient information to help in distinguishing the tree species in the MLS dataset. This also indicates that the high performance of NormalView cannot be attributed solely to the use of a geometric representation rather than radiometric information. The results indicate that normal-vector projections constitute a competitive alternative to depth-based projections for converting local three-dimensional geometry into a representation suitable for conventional image classification.

The performance metrics of Point Transformer in Table 7 reveal a clear relationship between model performance and the number of input features. The version using all three available point features, point coordinates, intensity values, and normal vectors, obtains the best performance, with an OA and MAA of 92.8% and 92.0%, respectively. This version is outperformed by NormalView by approximately

Table 7
Performance metrics of Point Transformer trained on point clouds subsampled to constant point density and a model trained using depth images on MLS data. XYZ refers to point coordinates, I to intensity values, and N to normal vectors. Bolded values indicate the best performance in each metric, respectively, and the values in parentheses indicate the difference relative to the NormalView model trained on the full-density MLS dataset (Table 4).

Type	OA (%)	MAA (%)	MAF_1 (%)	K (%)
Depth	**95.2** (−0.3)	**94.2** (−0.6)	**93.3** (+0.2)	**93.8** (−0.4)
Point Transformer				
XYZ	76.9 (−18.6)	66.6 (−28.2)	68.6 (−24.5)	69.6 (−24.6)
XYZ + I	86.5 (−9.0)	77.6 (−17.2)	79.6 (−13.5)	82.6 (−11.6)
XYZ + N	89.5 (−6.0)	83.7 (−11.1)	83.0 (−10.1)	86.5 (−7.7)
XYZ + I + N	92.8 (−2.7)	92.0 (−2.8)	90.3 (−2.8)	90.8 (−3.4)

three percentage points in each metric. When using only intensity information or normal vectors, the performance of Point Transformer degrades by a significant amount. Comparing with NormalView, the Point Transformer using point coordinates and normal vectors is substantially outperformed, with a difference of 6.0 and 11.1 percentage points in OA and MAA, respectively. Using only the point coordinates yields the worst results: an OA (MAA) of 76.9% (66.6%). Furthermore, the macro-average metrics decrease more rapidly than overall accuracy.

When comparing Point Transformer with the projection models, it should be kept in mind that the input data and the model architecture of Point Transformer are completely different to those of the projection models. The results do, however, reveal that projection-based methods are a viable alternative to point-based architectures on MLS data.

As mentioned above, adding auxiliary features improves the performance of Point Transformer. However, adding normal vector information to point coordinate information yielded noticeably stronger results than adding intensity information. Namely, adding intensity information improves the OA (MAA) by 9.6% (11.0%), while adding normal information to the point coordinates elevates the OA (MAA) by 12.6% (17.1%). This is on par with the findings in Section 3.1, where it was observed that NormalView achieved slightly better performance than the projection model using intensity data on the MLS dataset. The larger improvement obtained from adding normal vectors than from adding intensity information, particularly in MAA, suggests that local surface orientation provides useful discriminative information for tree species classification in high-density MLS data. The larger gain in MAA is particularly noteworthy, as it indicates that the benefit of normal information extends across all species. Importantly, normal information provided benefits in both projection-based and point-based architectures. Adding both normal vectors and intensity to point coordinates further improves the OA and MAA by 15.9 (25.4) percentage points, respectively, which is a notable improvement suggesting that normal vectors and intensity offer complementary information.

3.3. Ablation studies

We conducted ablation studies to inspect the effect of different parameters and data quality on the performance of the models. Namely, we tested how the nearest-neighbour parameter N in normal vector estimation affects the models, do the sliced images provide benefits as opposed to using no sliced images, and how significantly model performance drops, if at all, on lower-density MLS data.

As discussed in Section 2.4.1, to estimate the normal vector at a given point, we fit a plane to the nearest N points, and take the normal vector of the plane to be the normal vector at the point in question. To investigate the effect of this parameter, we created images by using a varying N in the normal vector estimation phase, and trained otherwise similar models on the images.

As the results in Table 8 show, the nearest-neighbour parameter does not affect NormalView's performance greatly. On dense MLS data, the best results are achieved with $N = 3$, with $N = 20$ achieving comparatively similar values. On MLS data, the results are all close to each other, with the exception of $N = 30$, where performance is noticeably weaker. On the ALS data the results are even more homogeneous, with $N = 5$ being the best, followed again closely by $N = 20$. However, the metrics of the various models are extremely similar. The results indicate that NormalView is relatively insensitive to the choice of the nearest-neighbour parameter in normal vector estimation. This is important, as it implies that extensive parameter optimisation is not necessary when applying the method.

We studied the effect of point density on the performance of NormalView on the MLS dataset. We conducted the study by randomly subsampling the MLS tree segments to a desired point density, and creating images from the subsampled segments, and training models on them.

The results, presented in Table 9, are perhaps a little surprising. The performance generally degrades as the point density decreases. On the lowest density data, which has a point density of $500\,\mathrm{pts/m^2}$, the overall accuracy drops only 2.4 percentage points, while the MAA drops by 4.9 percentage points. On $2000\,\mathrm{pts/m^2}$ the OA drops only by 0.6 percentage points and the MAA and MAF_1 decrease by 3.1 and 1.0 percentage points, respectively, when compared to the model trained on the full density data. The separation between the models trained on subsampled data is even more moderate. The results on the downsampled data for point densities $2000\,\mathrm{pts/m^2}$ and $1000\,\mathrm{pts/m^2}$ are slightly stronger than what NormalView achieved on the ALS data and seven species, which is a dataset consisting of the same species at approximately the same point density. The above findings suggest that a full coverage of the tree, as obtained from a below-canopy scanner, is more important for performance than a high point density. However, from the MAA and MAF_1 it can be inferred that it is easier to distinguish minority species when the point density is higher.

Finally, Table 10 shows the metrics on models trained on intensity, normal, and black-and-white MLS data without slice images. The slice images improve the performance of the intensity and NormalView models, particularly the latter, where removing slice images decreases the macro-average accuracy by 3.7 percentage points. This indicates that exposing the internal branch and trunk structure provides additional discriminative geometric information beyond the outer silhouette. In contrast, the black-and-white model slightly benefits from removing slice images, suggesting that silhouette-only representations cannot effectively exploit the additional internal structure and may instead be affected by the increased image complexity.

3.4. Effect of image size on model performance and computational efficiency

Fig. 9(a) shows that increasing image size improves model performance to a certain extent. The highest test accuracy (95.2%) and macro-average test accuracy (93.8%) are achieved by the model using 512 × 512 as the image size. The models using higher image size, 768 × 768 and 1024 × 1024, perform slightly worse than the model using size 512 × 512, and rather equally to the model using size 256 × 256. The training accuracies are markedly higher than test accuracies for all models. Curiously, the model using size 512 × 512 has the third-worst training accuracy, surpassing only the models using sizes 32 × 32 and 64 × 64. This hints that the model overfitted to the training data the least among the models. The model using image size

Table 8
Performance metrics of NormalView trained with varying number of nearest points N in normal vector estimation. The bolded values indicate the best performance in each metric, respectively.

	MLS				ALS			
N	OA (%)	MAA (%)	MAF_1 (%)	K (%)	OA (%)	MAA (%)	MAF_1 (%)	K (%)
3	**95.5**	**95.1**	**94.6**	**94.2**	91.2	78.0	80.9	88.5
5	94.6	92.7	92.9	93.0	**91.8**	**79.5**	**81.9**	**89.2**
10	94.3	93.6	92.3	92.7	91.7	78.9	81.5	89.1
15	94.3	92.9	91.7	92.7	91.1	77.8	80.3	88.4
20	**95.5**	94.8	93.1	94.2	**91.8**	79.1	81.8	**89.2**
30	92.8	91.2	89.7	90.7	**91.8**	78.8	80.4	**89.2**
40	94.5	92.1	91.6	92.7	91.7	78.7	80.4	89.1

Table 9
Metrics of NormalView trained on MLS data with varying point densities. Bolded values indicate the best performance in each metric, respectively, and the values in parentheses indicate the difference relative to the NormalView model trained on the full-density MLS dataset (Table 4), where the median and mean point densities are 6147 pts/m^2 and 6917 pts/m^2, respectively.

Point Density (pts/m^2)	OA (%)	MAA (%)	MAF_1 (%)	K (%)
5000	94.6 (−0.9)	**92.2** (−2.6)	91.3 (−1.8)	93.1 (−1.1)
4000	93.4 (−2.1)	90.4 (−4.4)	89.1 (−4.0)	91.5 (−2.7)
3000	93.7 (−1.8)	90.1 (−4.7)	89.6 (−3.5)	91.9 (−2.3)
2000	**94.9** (−0.6)	91.7 (−3.1)	**92.1** (−1.0)	**93.4** (−0.8)
1000	94.0 (−1.5)	90.0 (−4.8)	90.5 (−2.5)	92.3 (−1.9)
500	93.1 (−2.4)	89.9 (−4.9)	89.4 (−3.7)	91.1 (−3.1)

Table 10
Performance of models trained on MLS data without slice images. Bolded values indicate the best performance in each metric, respectively, and the values in parentheses indicate the difference relative to the models trained using images created by the same projection procedure, with the addition of slice images (Table 4).

Type	OA (%)	MAA (%)	MAF_1 (%)	K (%)
Intensity	92.3 (−2.3)	90.6 (−1.0)	90.7 (−1.0)	90.7 (−2.3)
Normal	93.7 (−1.8)	91.1 (−3.7)	90.2 (−2.9)	91.9 (−2.3)
Black-and-white	**94.0** (+0.0)	**92.9** (+2.2)	**91.5** (+1.7)	**92.3** (+0.0)

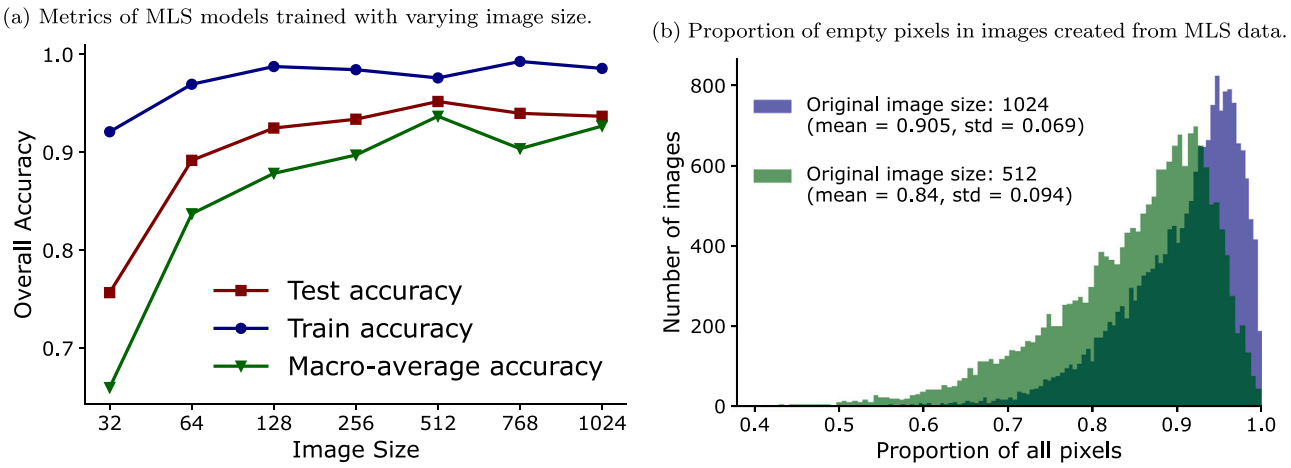


Fig. 9. (a) Performance metrics of models trained on MLS data and varying image sizes. (b) Proportions of empty pixels in images created from MLS data to sizes 1024 × 1024 and 512 × 512.

512 × 512 achieves better results than the model trained on intensity-based images, where the initial image size is set to 512 × 512, which are visible in Table 4. Computationally, the models using larger image sizes require more memory and processing power, both during training and inference.

Fig. 9(b) shows that the images created into size 1024 × 1024 tend to have more empty pixels. The pixel sizes are 1.64 cm (± 0.47 cm), and 3.28 cm (± 0.93 cm) for the sizes 1024 × 1024 and 512 × 512, respectively, when evaluated by the average heights of the trees.

To assess the computational requirements of the different projection representations, the complete inference pipeline is evaluated on an uniformly sampled subset of 500 trees in the MLS dataset for each of the four image creation modalities described in Section 2.4.1. For each tree, the point cloud is processed to generate the 50 projection images used for inference, after which the images are passed to the trained YOLO model to obtain the tree-level prediction. In what follows, preprocessing will refer to the following operations: loading the points of the tree segments into RAM memory, performing the necessary computations for each modality (normal estimation, normalisation and depth value estimation, etc...), image creation and writing the images to disc. Inference will refer to passing the created images through the YOLO model, and obtaining the species prediction. In combination, these steps comprise the inference pipeline described in Section 2.4.2, and consequently the sum of the preprocessing and inference times corresponds to the total time it takes for the method to obtain a prediction from a tree point cloud. We used 25 angles with sliced

Table 11
Average preprocessing and inference times per tree for the different projection representations, using 25 viewing angles and slice images (50 images in total per representation). Values are reported as mean ± standard deviation over the 500 trees used in the tests.

Type	Preprocessing (s)	Inference (s)	Total (s)
Intensity	2.78 ± 2.59	0.86 ± 0.16	3.64 ± 2.71
Normal	3.50 ± 3.38	0.88 ± 0.17	4.38 ± 3.50
Black-and-white	3.10 ± 2.90	0.85 ± 0.15	3.94 ± 3.00
Depth	5.69 ± 4.71	0.96 ± 0.20	6.65 ± 4.83

(a) Average inference times of NormalView as a function of the number of angles. The error bars signify one standard deviation.

(b) Preprocessing time of NormalView as a function of the number of points in the segment, and the related power-law fit, on a log-log-scale.

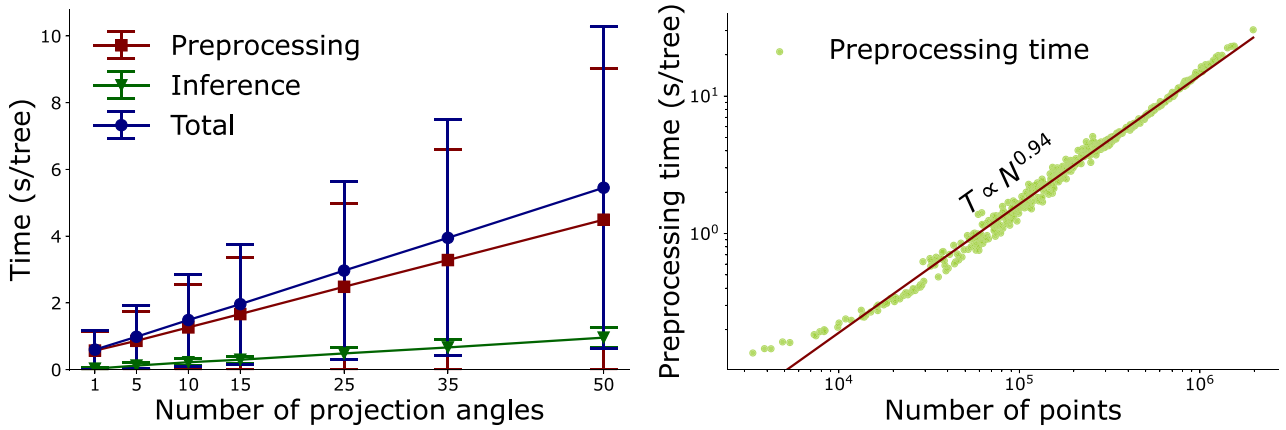


Fig. 10. (a) The average preprocessing, inference, and total times of the NormalView pipeline as a function of the number of angles. (b) The preprocessing times of NormalView as a function of the number of points in the tree segment and the associated power-law fit.

images, as in the regular inference, and thus create 50 images per tree. All computational efficiency tests were run on the same machine used in model training, with 32 GB of RAM, and a 12 GB Nvidia GPU (for further details, please refer to Table 3).

From Table 11 we see that the total inference times for each image creation modality are, on average, a few seconds on dense MLS data. The preprocessing times are 3–5 times larger than the inference times for each model. Thus, as expected, the computational bottleneck is in the image creation and writing phase, rather than in the model inference phase. Also notable is that the standard deviations are much larger for the preprocessing times than for the inference times. This is expected because loading the point cloud and the subsequent preprocessing operations depend on properties of the input tree, most notably its point count, whereas inference is performed on a fixed number of images with identical dimensions.

As mentioned, the substantial variation in preprocessing time is closely related to the size of the input point clouds. Fig. 10(b) shows the preprocessing time of NormalView as a function of the number of points in the corresponding tree segment on a log–log scale. The relationship is approximately linear on a log–log scale, indicating a power law dependence of the form $T \propto N^{\alpha}$, where T denotes preprocessing time and N the number of input points. A linear regression in the log–log space gives $\alpha = 0.94$, with a high coefficient of determination, $R^2 = 0.938$. Thus, the preprocessing time scales approximately linearly with the number of points, although the exponent being slightly below one indicates marginally sublinear scaling over the range of point-cloud sizes considered. From Fig. 10(b) we can also see that for small point counts the linear trend breaks down. This happens, because for small point counts the point loading, precomputation, and image creation steps are fast enough for the computational overhead from, for example, writing the images to disc to become significant. The time it takes to write the images to disc is not as significantly affected by the point count, as the other steps.

The fastest modality is the intensity model with the average total time being 3.64 s, followed by the black-and-white model. Third fastest is the normal model, and the slowest is the depth model, by some margin. The preprocessing phase of the normal model is expectedly slower than those of the intensity and black-and-white models. This can be attributed, at least in some part, to the normal estimation, which is computationally demanding. The normal model is 0.74 s slower in total time, when compared to the intensity model. One explanation for the relative slowness of the depth projection implementation could be the median centring and uniform scaling. These normalisation steps are used by both Allen et al. (2022) and the DetailView method in Puliti et al. (2025b), but we suspect that they would not be necessary to obtain similar performance. Notably, the differences in total times are almost completely due to the differences in preprocessing times. The average inference times are all within 0.11 s of each other.

Since NormalView aggregates predictions from multiple projection views, we additionally investigate how the computational cost scales with the number of angles used. The time to it takes complete pipeline from tree point cloud to species prediction was evaluated using 1, 5, 10, 25, and 50 projection angles per tree. For this experiment, we do not use the slice images, so only one image is created per angle.

Fig. 10(a) shows a clear increase in preprocessing and total runtime as the number of projection angles increases. The increase is approximately linear over the tested range, although the increase in total runtime is somewhat less than proportional to the number of angles. As preprocessing constitutes the dominant part of the computational cost, the observed trend is primarily driven by the additional projection images that must be generated and written for each angle. In contrast, inference time changes comparatively little. This is due to the fact that images can be passed to the YOLO architecture in batches, and thus, the inference time increases comparatively slowly, at least in the tested range.

3.5. Further discussion

The results in Section 3.1, as well as Fig. A.1, indicate that NormalView is highly adept at classifying tree species from high-density

MLS data. Especially the confusion matrices A.1 show that NormalView achieves relatively strong performance on the minority species, rowan and lime, compared with the black-and-white and intensity-based representations in this experiment. However, also the performance metrics of NormalView are lower on the minority species than on the majority species, likely due to the imbalances in the dataset. Compared to the model trained on black-and-white images, NormalView achieved a 4.1 percentage points better MAA and 3.3 percentage points better MAF_1, see Table 4. This indicates that if we only have geometric data on hand, the normal vector colouring improves classification performance. In particular, it seems that the normal vector colouring captures local information, which yields better classification performance compared to the black-and-white model, which uses only the silhouette information.

On the full ALS dataset, models using multiple intensity achieved the highest metric values. The model with all three intensity channels was a strong performer with the seven species and the nine species datasets. This is in accordance with the findings in Taher et al. (2026), where they found that especially low-density point cloud data benefits from multispectral information. On lower-density data, the importance of radiometric information becomes more significant, as the data is sparser, and the underlying geometry of the point cloud becomes inconsistent and more unreliable. The performance of the best models on the data is comparable to the best models of the benchmark conducted by Taher et al. (2026), where they achieved an OA (MAA) of 92.0% (85.1%) on a similar train–test split as here. Our best-performing model, the one with all three intensity channels, shows essentially equal performance. The corresponding metrics for NormalView are 91.8% (79.1%), which are worse, but not by a large margin. The key difference is that the best model in the benchmark utilised full radiometric data from all three scanners, while NormalView uses only geometric information. Images coloured by the normal vector estimations benefit from the tree point clouds having consistent and high-density trunks and branches, as this allows for smoothly varying and consistent normal vectors and hence smoother and bounded colour gradients. This likely explains the comparatively stronger performance of NormalView on MLS data, compared to ALS data. On the ALS dataset with the same seven species as in the MLS data, NormalView was not the strongest performer, unlike on the MLS data. The results on the MLS data suggest that especially in applications requiring high-density point clouds, such as exact inventory of natural forests, normal vector colouring is a promising representation.

There seems to be little performance difference between the models using ALS data and intensity information from a single channel. On the seven species subset, the model using channel 1 (VUX-1HA) achieved the highest performance metrics amongst the single channel models, while on the full nine species data set the model using channel 2 (miniVUX-1DL) achieved the strongest results. Overall, the performance results are similar and there seems to be little difference in which scanner is used. This suggests that using radiometric information with projection-based methods can be made viable on scanners with varying properties, such as the wavelength of the laser pulse.

We see from Fig. 7 that across the board the models perform the worst, by precision and F_1, on oak, and, to a lesser degree, alder. These species are the two with the smallest representation in our dataset, and hence the least variety. In addition, the segments of the alders and oaks have some impurities, as suggested by Figure 5 in Taher et al. (2026). Alders tend to grow in groups, making segmentation more difficult. A significant portion of oak segments are only partial segments of the trees. The low performance on these species can probably at least partially be attributed to low sample counts and lower-quality segments. From the confusion matrices in Appendix A we see that almost all models tend to systematically classify aspens as birches. Birches and aspens have rather similar silhouettes, which explains the errors. Similarly, rowans seem to get classified as maples and birches. Some examples of trees misclassified by the MLS models are shown in Fig. 11. It seems that poorly captured trees with lacklustre silhouettes are prone to misclassification, such as the aspen and birch trees in Fig. 11. This is not the only explanation for misclassification, as for example the spruce, lime, and maple trees in Fig. 11 are captured well by the scanner, and have full silhouettes. To enhance the performance of the models on species with similar silhouettes, one could, for example, include more detailed bark information, as is done in the DetailView method discussed by Puliti et al. (2025b).

The F_1-scores for aspens are satisfactory across all data modalities and models. On the MLS data, the scores are around 80%, while between 83 and 87% for the ALS data and all nine species. This shows that we are able to map aspens reasonably consistently from both MLS and ALS data. The use of multispectral information improves the scores by a small margin, but the intensity-free NormalView and black-and-white models achieve F_1 scores of 85% and 83% respectively, highlighting the capabilities of projection-based methods.

The black-and-white model is the worst performing model on MLS data, and among the worst performers on ALS data. On the MLS data and the nine species ALS data, it achieves an OA (MAA) of 94.0% (90.7%) and 91.6% (79.5%) respectively. These results show that even using purely black-and-white orthographic projections allows us to classify tree species on a reasonably high level. This is further supported by the performance of the black-and-white model on the MLS + ALS data. The performance is due to the high capabilities in which modern image-based architectures, such as YOLO, are able to extract meaningful representations and features from the data. The image classification backbones used in previous studies of tree species classification seem to be, in general, slightly outdated, such as the VGG-11 (Simonyan and Zisserman, 2015) used by Marinelli et al. (2022), or the DenseNet-201 (Iandola et al., 2014) used by the DetailView method in Puliti et al. (2025b).

The comparative experiments demonstrate that NormalView performs competitively with both depth-based projection and a point-based deep learning architecture on the MLS dataset. The close correspondence between the depth and NormalView results indicates that depth and surface-normal representations provide similarly informative geometric cues for species classification. NormalView also was found to be computationally more efficient than the used depth projection method. The Point Transformer experiments provide complementary evidence for the usefulness of normal vectors. Adding normal vectors to point coordinates produced a larger improvement in the evaluation metrics than adding intensity information alone. Despite using a two-dimensional projection representation, NormalView achieves higher performance than the evaluated Point Transformer configurations, indicating that the projection process does not prevent the model from exploiting the geometric information required for accurate species classification. One benefit of the normal vectors compared to, for example depth, is that they can be used with point-based 3D-architectures, while depth needs an explicit viewing direction, which seems to make its generalisation to non-projective archetypes nontrivial. Another benefit, at least in the experiment setup used here, is that normal vectors do not require a per-tree calibration step, unlike the depth representation used here, which median-centred and normalised the tree coordinates. This calibration likely also caused the depth projections to be computationally slower than the NormalView pipeline.

As mentioned previously, depth and normal information offer somewhat complementary geometric information. Namely, depth represents the relative position of the observed point cloud surface to the viewer, normal vectors encode local orientation and geometry. One possible approach could be to simply combine the normal vector and depth images into a single, 4-channel representation. These could be then used to train a convolutional network. Such a fusion remains a potential direction for future work.

An important finding of this study is that the relative advantages of the different projection methods depend on the acquisition modality. On dense MLS data, the model using normal vector information achieves the highest values across the reported metrics, demonstrating

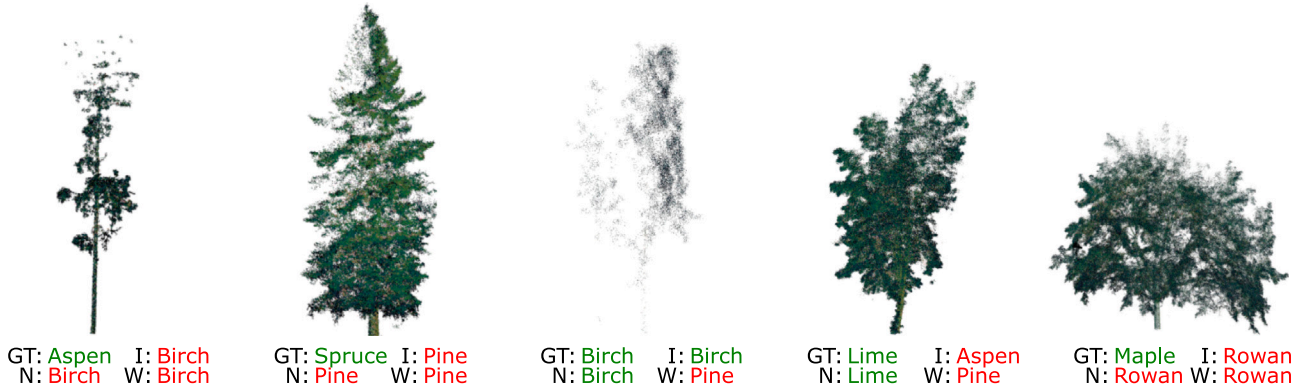


Fig. 11. Examples of MLS-trees misclassified by at least one model. GT refers to ground truth, I, *N* and W refer to the predictions given by the intensity, normal and black-and-white models respectively. Green indicates a correct prediction and red an incorrect one.

that local surface orientation contains useful information beyond point coordinates for species classification. On ALS data, however, multispectral intensity information proves more beneficial, particularly for distinguishing minority species in the nine-species dataset. When both acquisition methods are combined into a single dataset, the performance differences between the models largely disappears. This suggests that modality-specific variation limits the effectiveness of representations tailored to a single sensor type and highlights the importance of developing acquisition-invariant features for cross-platform tree species classification.

The ablation studies provide additional insight into the robustness of the proposed approach. NormalView is found to be largely insensitive to the choice of the number of nearest neighbours used for normal estimation, suggesting that the method does not require extensive parameter tuning when applied to new datasets. Similarly, the relatively small performance degradation observed after substantial point cloud downsampling indicates that the method primarily exploits larger-scale structural characteristics of trees rather than relying on extremely dense point clouds. This is encouraging from a practical perspective, as the density of resulting point clouds may vary. Furthermore, the experiments demonstrate that slice images provide meaningful additional information for the intensity-based and geometry-based representations by exposing the internal branch and trunk structure, whereas silhouette-only images benefit less from the additional views. Together, these findings indicate that NormalView is not only accurate but also robust with respect to several practical factors encountered in real-world applications.

As seen in Section 3.4, using an appropriate image size can noticeably affect the predictive power of the networks. Using larger images allows for more detailed information, at the cost of longer training times and higher hardware requirements. Hence, it is critical that before training the models, we have some notion of the appropriate size into which to create the images. For images created into size 1024×1024, one pixel size corresponds approximately to a square of size $1.64\,\text{cm} \times 1.64\,\text{cm}$ in the projective plane. In the preprocessing phase, we subsampled the point clouds to a spatial resolution of $2\,\text{cm}$, which is larger than the size of the pixels. As a result, there are many empty pixels, even at points where we would assume the tree to be visible. Certainly, as a general guideline, the resolution should be such that one pixel encompasses an area that is slightly larger than the spatial resolution of the underlying point cloud data. This is supported by our findings in Section 3.4, where the model with image size 512 was the best. For images of size 512×512, the pixel size is $3.28\,\text{cm} \times 3.28\,\text{cm}$, which is roughly one-and-a-half times larger than the spatial resolution of the densest parts of the point clouds. Fig. 9(b) shows that higher resolution images tend to have more empty pixels, which is, of course, expected. Convolutional neural networks (CNNs) are usually designed with the assumption that the input data will be dense (Graham et al., 2018). If the image size in the projections is large, the images will start to be rather sparse, in the sense that most of the image will be empty pixels, containing no information. At the very least, this will waste time and require more computational resources with little to no added performance benefit. In the worst case, empty pixels will negatively affect the model.

The computational efficiency experiments presented in Section 3.4 provide context for the applicability of NormalView in real use-case scenarios. The total time to arrive at a species prediction from a tree point cloud is, on average, approximately $4.5\,\text{s}$. In real world applications, for example in a real-time forest segmentation and species classification pipeline, the total inference times remain practical. As was also seen, the total inference time can be reduced by reducing the amount of projection angles used, and as such, the proposed methodology offers flexibility. Increasing the amount of viewing angles beyond a certain point will provide diminishing results, as the projection images taken from different angles which are close to each other, provide essentially the same information.

3.6. Limitations

The datasets used in this study, while containing the most predominant boreal tree species, are incomplete even for traditional Finnish ecosystems. Hence it is not applicable to use the models directly in practice. The training of the models also requires significant amounts of annotated data. Collecting and labelling training data for deep learning models is notoriously time-consuming and, as such, highlights the need for large multispecies datasets, such as the recently released FORspecies20k and the MS-ALS-SPECIES dataset (Puliti et al., 2025a; Hyyppä et al., 2025).

NormalView and the other models in this study were tested only on data gathered from a periurban ecosystem in Southern Finland. Thus, the generalisability of the proposed method, NormalView, to a different type of ecosystem, such as a tropical forest, warrants further study. Different forest types can exhibit considerably different characteristics, for example different branching patterns, crown structures, and epiphytic vegetation. NormalView was designed to embed local geometry of the tree into the projection images, which should support its generalisation to trees with different structural characteristics. This does not guarantee equivalent performance in other environments however, as performance depends on a multitude of factors, such as point density, segment quality, occlusion, and scanning conditions. For example, in denser forest environments the point clouds can become noisier, and the tree point cloud segments can have more inconsistent trunk and branch information, which can degrade the performance of NormalView.

Furthermore, the used datasets are somewhat imbalanced when it comes to the species counts. While macro-averaged metrics were

reported for each model, no explicit strategies were used to mitigate the species imbalance. The comparatively lower performance of practically every model on the minority species shows that the models do not learn the representations for the minority species as efficiently as for the majority species. Thus, further research is required to mitigate this issue. Some methods which could be used to address data imbalance are: adding data augmentations in the training phase in proportion to the rarity of the class in the dataset, as is done by the FBK-PointNet++-DL-3D -method in Taher et al. (2026), creating synthetic data from the minority species at the cost of more computation and longer training times, and using a weighted loss function which penalises misclassifications on minority classes more severely, such as the weighted cross-entropy, which was used by a Point Transformer model obtaining the best MAA in the benchmark by Taher et al. (2026). Using a weighted loss function can have the side effect of lowering overall accuracy, as it may in turn hinder the performance on the majority species. It should be noted that species imbalance is not necessarily undesirable as such, if the distribution in the dataset reflects the distribution in the application domain, as the classifier then can learn this distribution, but if the amount of training samples of the minority classes is small, then the combination of skewed data and a small amount of training samples does hinder classifier performance (López et al., 2013). The aforementioned weaker performance on the minority species suggests that this is possibly the case here.

In the MLS data, the segment quality is extremely high. Our empirical experience is that better segment quality improves classification performance. For real world applications, the segment quality might be significantly lower, and the accuracy of the models may suffer from these inaccuracies. The effect of segment quality on tree species classification warrants further study.

Due to the computational cost of training the models, each model was trained and tested using only a single training run. Thus, the variability between runs arising from the stochastic nature of the training phase is not accounted for. Since each model was trained only once, small differences in performance should be interpreted cautiously, moreso descriptive of performance, than as giving an absolute statistical ranking of the models. All experiments in this study use a fixed tree-level train–test split. This design ensures that the comparisons between representations and experimental conditions are performed under identical data partitions, but it does not allow us to quantify variability due to the random initialisation of the network or the choice of train–test partition. We note that in a study with similar data as here, Hyyppä et al. (2026) investigated the variability in performance of a species classifier arising from the test set by bootstrapping it with replacement 2000 times and reporting 95% confidence intervals for the performance metrics. They observed that the selection of the test set causes relatively little variability in the performance of the model. This does not account for the variability inherent to the training phase, but it does provide some context for the stability of experiments based on a single test set. Nevertheless, repeated experiments using different random seeds and, ideally, cross-validation would provide confidence intervals for the reported metrics and constitute an important direction for future work.

4. Conclusions

In this study, we presented NormalView, a projection-based intensity-independent tree species classification method designed for classification from MLS or high-density ALS data. The method embeds local geometric information of tree point clouds into projection images via colour. The colour values are derived from normal vector estimates of the points.

We compared our projection-based method with other models trained on projection images using intensity information, black-and-white projection images, and depth-based projection images, as well as the point-based Point Transformer model. We found that on high-density seven species MLS data our NormalView achieves the highest performance metrics compared to the other evaluated methods, with the overall accuracy (macro-average accuracy) reaching 95.5% (94.8%). The strong performance on dense MLS data, together with the robustness observed in the ablation studies, suggests that NormalView is particularly well suited for forestry related point cloud processing pipelines with high-density data.

For high-density helicopter ALS data, we found that our method achieves comparable performance with methods using intensity information. However, the best model on the ALS data was the one using intensity information from three channels of the multispectral ALS. It achieved an overall (macro-average) accuracy of 92.5% (84.9%) on the full dataset with nine species. These results are on par with the best models in Taher et al. (2026), showing that projection-based methods using strong CNN-backbones can achieve performance comparable to that of state-of-the-art 3D models. The experiments indicate that geometry-based representations perform particularly well on dense point clouds, while multispectral radiometric information remains beneficial for more challenging low-density airborne datasets.

Our tests show that using multispectral data, i.e. combining radiometric information from multiple scanners, improves the prediction ability of projection-based models. The benefits are best seen for species belonging to minority classes. We also inspected the effect of image size on model performance, and found that using larger images is beneficial up to a certain point, enhancing the accuracy of the model. For the high-density MLS data we found that image size 512×512 produces the best results.

CRediT authorship contribution statement

Juho Korkeala: Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Jesse Muhojoki:** Writing – original draft, Supervision, Resources. **Josef Taher:** Writing – review & editing, Conceptualization. **Klaara Salolahti:** Writing – review & editing, Validation, Data curation. **Matti Hyyppä:** Writing – review & editing, Data curation. **Antero Kukko:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Juha Hyyppä:** Writing – original draft, Supervision, Project administration, Funding acquisition.

Code availability

The code used for NormalView image creation and model training is available in Korkeala (2026).

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

The study has received funding from the Research Council of Finland (RCF), and the NextGenerationEU instrument through the following grants "Collecting accurate individual tree information for harvester operation decision making" (RCF 359554), "Digital technologies, risk management solutions and tools for mitigating forest disturbances" (RCF 353264, NextGenerationEU), and "Forest-Human–Machine Interplay – Building Resilience, Redefining Value Networks and Enabling Meaningful Experiences" (RCF 359175) using RCF research Infrastructure "Measuring Spatiotemporal Changes in Forest Ecosystem" (346382)".

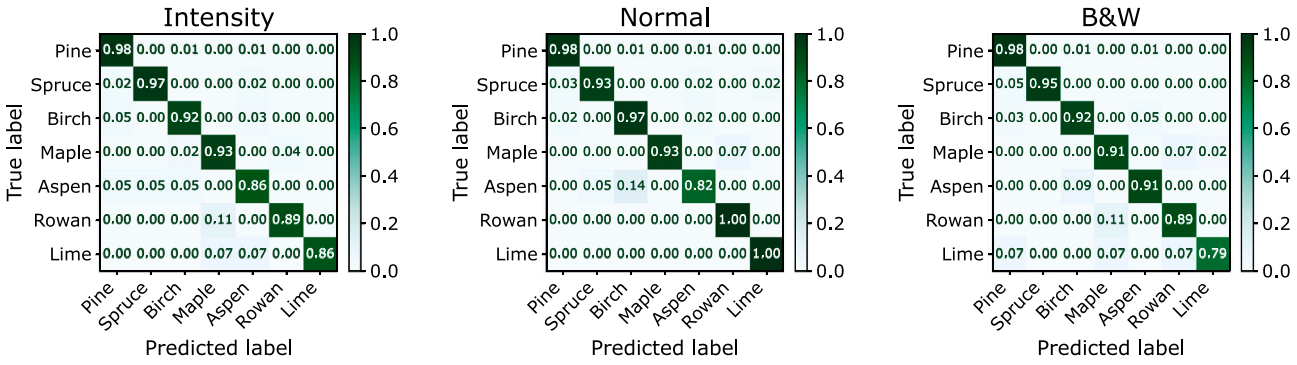


Fig. A.1. Normalised MLS confusion matrices.

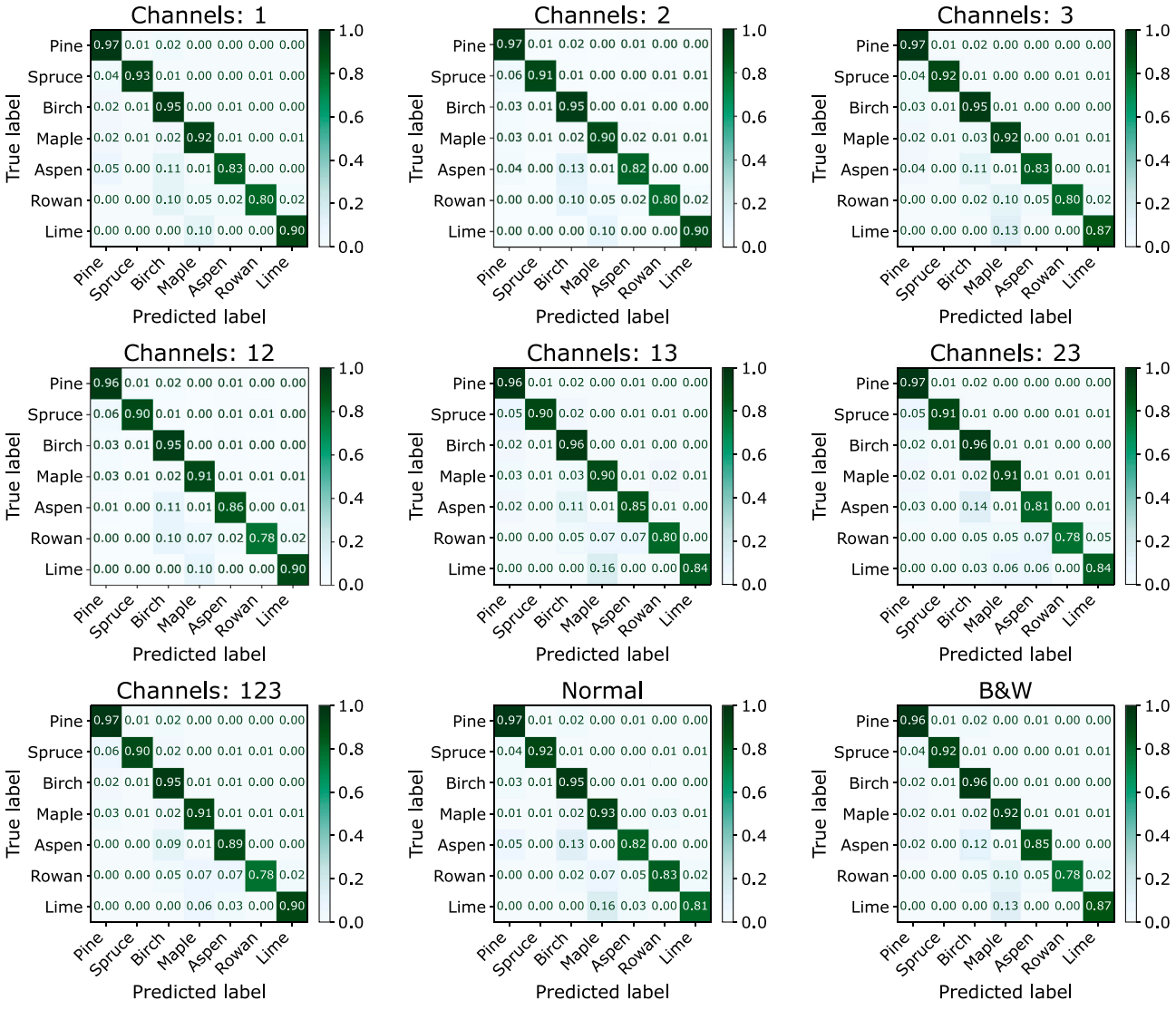


Fig. A.2. Normalised confusion matrices (ALS, 7 species).

Appendix A. Confusion matrices

Figs. A.1 to A.4 present the normalised confusion matrices for the trained models.

Appendix B. Point transformer results with constant point count

Table B.1 present the results of the Point Transformer model on MLS data, when the tree segments are uniformly downsampled to contain 8192 points before being passed to the model.

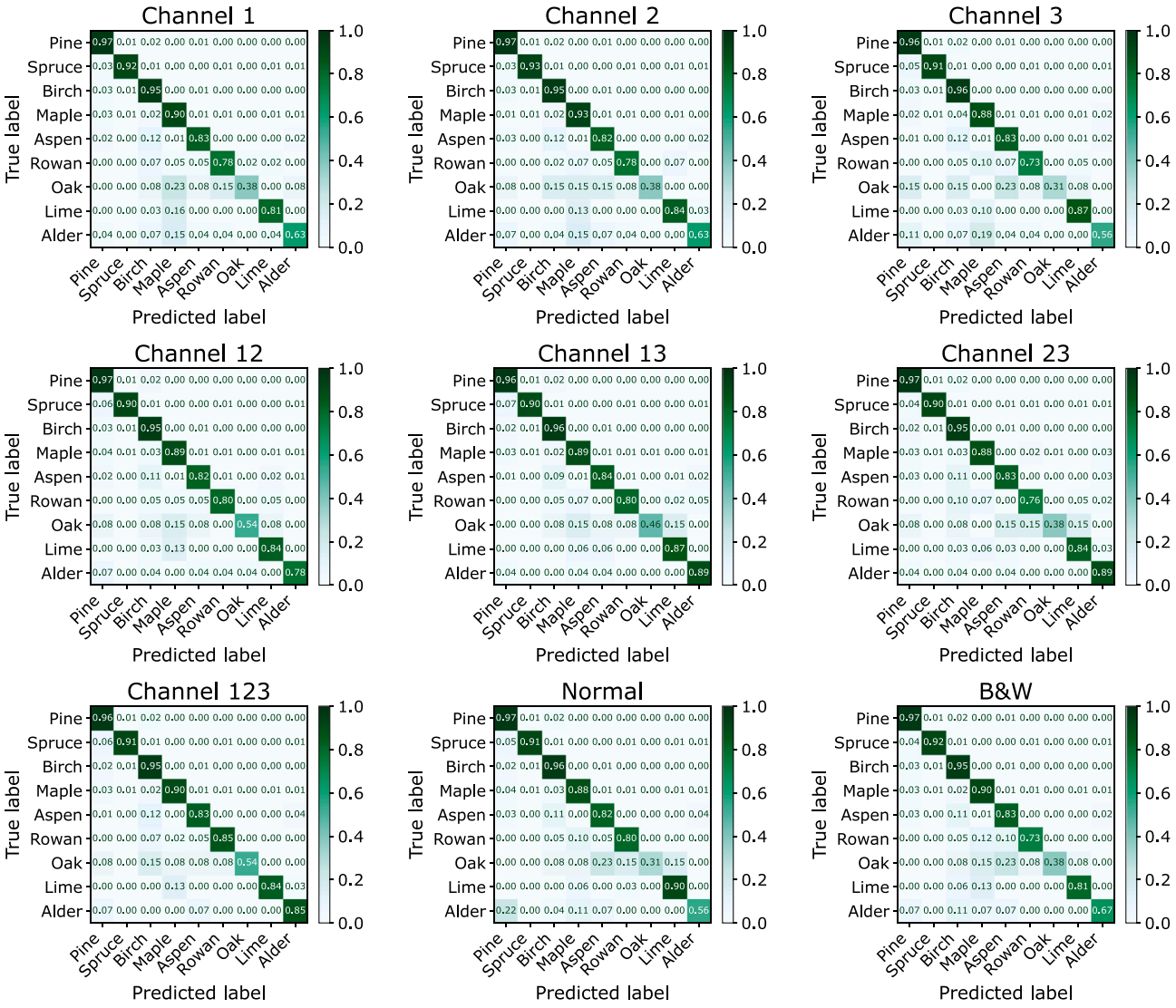


Fig. A.3. Normalised confusion matrices (ALS, 9 species).

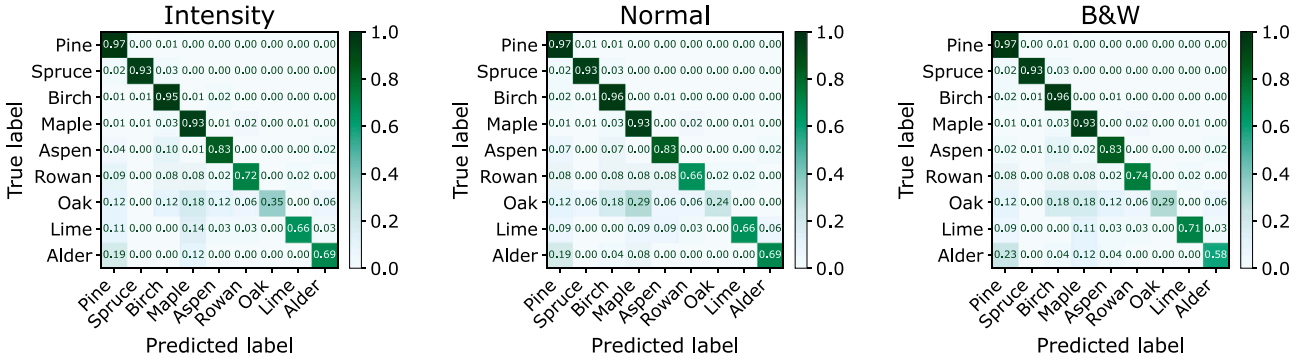


Fig. A.4. Normalised confusion matrices (MLS + ALS, 9 species).

We see that when using a constant point count for the input segments, the Point Transformer performs worse than the version using a constant point density, when all three input features (point coordinates, intensity, normal vectors) or point coordinates and normal vectors are used. With point coordinates and intensity, the performance is rather similar, and with only point coordinates, the one using constant point count achieves higher metrics. Overall, the highest performance metrics by Point Transformer are obtained by downsampling to a constant point density and using all three input features.

Data availability

Data available at: https://zenodo.org/records/21039101.

Table B.1
Performance metrics of Point Transformer using point clouds downsampled to a constant point count on MLS data. XYZ refers to the coordinate values of the points, I to intensity, and N to normal vectors. Bolded values indicate the best performance in each metric, and values in parentheses indicate the difference relative to the NormalView model trained on the full-density MLS dataset (Table 4).

Type	OA (%)	MAA (%)	MAF_1 (%)	K (%)
XYZ	79.6 (−15.9)	72.8 (−22.0)	71.1 (−22.0)	73.8 (−20.4)
XYZ + I	87.7 (−7.8)	77.3 (−17.3)	79.5 (−13.6)	84.1 (−10.1)
XYZ + N	85.3 (−10.2)	72.4 (−22.4)	74.6 (−18.5)	80.9 (−13.3)
XYZ + I + N	**92.2** (−3.3)	**87.5** (−7.3)	**87.6** (−5.5)	**90.0** (−4.2)

References

Allen, M., Grieve, S., Owen, H., Lines, E., 2022. Tree species classification from complex laser scanning data in mediterranean forests using deep learning. Methods Ecol. Evol. 14, 1657–1667. http://dx.doi.org/10.1111/2041-210X.13981.

Angelstam, P., Mikusiński, G., 1994. Woodpecker assemblages in natural and managed boreal and hemiboreal forest - a review. Ann. Zool. Fennici. 31, 157–172.

Arumäe, T., Lang, M., Laarmann, D., 2020. Thinning-and tree-growth-caused changes in canopy cover and stand height and their estimation using low-density bitemporal airborne lidar measurements–a case study in hemi-boreal forests. Eur. J. Remote. Sens. 53, 113–123. http://dx.doi.org/10.1080/22797254.2020.1734969.

Bauwens, S., Bartholomeus, H., Calders, K., Lejeune, P., 2016. Forest inventory with terrestrial lidar: A comparison of static and hand-held mobile laser scanning. Forests 7, 127. http://dx.doi.org/10.3390/f7060127.

Blackburn, R.C., Buscaglia, R., Sánchez Meador, A.J., Moore, M.M., Sankey, T., Sesnie, S.E., 2025. Eigenfeature-enhanced deep learning: advancing tree species classification in mixed conifer forests with lidar. Remote. Sens. Ecol. Conserv. http://dx.doi.org/10.1002/rse2.70014, **Retracted**.

Bradski, G., 2000. The opencv library. Dr. Dobb's J. Softw. Tools.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Chirici, G., McRoberts, R.E., Winter, S., Bertini, R., Brändli, U.B., Asensio, I.A., Bastrup-Birk, A., Rondeux, J., Barsoum, N., Marchetti, M., 2012. National forest inventory contributions to forest biodiversity monitoring. For. Sci. 58, 257–268. http://dx.doi.org/10.5849/forsci.12-003.

City of Espoo, 2024. Espoo WMS and WFS open data resources. retrieved 10.6.2024, URL: https://kartat.espoo.fi/opendata/.

CloudCompare development team, 2024. Cloudcompare, version 2.13.1 [software]. URL: http://www.cloudcompare.org/.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20, 37–46. http://dx.doi.org/10.1177/001316446002000104.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297. http://dx.doi.org/10.1007/BF00994018.

Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V., 2020. Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 3008–3017. http://dx.doi.org/10.48550/arXiv.1909.13719.

Dalponte, M., Bruzzone, L., Gianelle, D., 2012. Tree species classification in the southern alps based on the fusion of very high geometrical resolution multispectral/hyperspectral images and LiDAR data. Remote Sens. Environ. 123, 258–270. http://dx.doi.org/10.1016/j.rse.2012.03.013.

Dalponte, M., Ene, L.T., Gobakken, T., Næ sset, E., Gianelle, D., 2018. Predicting selected forest stand characteristics with multispectral ALS data. Remote. Sens. 10, 586. http://dx.doi.org/10.3390/rs10040586.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. http://dx.doi.org/10.1109/CVPR.2009.5206848.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv Preprint: arXiv.2010.11929 arXiv:2010.11929.

Ferreira, M.P., dos Santos, D.R., Ferrari, F., Coelho, L.C.T., Martins, G.B., Feitosa, R.Q., 2024. Improving urban tree species classification by deep-learning based fusion of digital aerial images and LiDAR. Urban For. & Urban Green. 94, 128240. http://dx.doi.org/10.1016/j.ufug.2024.128240.

Gamfeldt, L., Snäll, T., Bagchi, R., Jonsson, M., Gustafsson, L., Kjellander, P., Ruiz-Jaen, M.C., Fröberg, M., Stendahl, J., Philipson, C.D., Mikusiński, G., Andersson, E., Westerlund, B., Andrén, H., Moberg, F., Moen, J., Bengtsson, J., 2013. Higher levels of multiple ecosystem services are found in forests with more tree species. Nat. Commun. 4, 1340. http://dx.doi.org/10.1038/ncomms2328.

Gazzea, M., Kristensen, L.M., Pirotti, F., Ozguven, E.E., Arghandeh, R., 2022. Tree species classification using high-resolution satellite imagery and weakly supervised learning. IEEE Trans. Geosci. Remote Sens. 60, 1–11. http://dx.doi.org/10.1109/TGRS.2022.3210275.

Gollob, C., Ritter, T., Nothdurft, A., 2020. Forest inventory with long range and high-speed personal laser scanning (PLS) and simultaneous localization and mapping (SLAM) technology. Remote. Sens. 12, 1509. http://dx.doi.org/10.3390/rs12091509.

Goyal, A., Law, H., Liu, B., Newell, A., Deng, J., 2021. Revisiting point cloud shape classification with a simple and effective baseline. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021. ML Research Press. pp. 3809-3820. doi:arXiv:2106.05304v1. publisher Copyright: Copyright ©2021 by the author(s); 38th International Conference on Machine Learning, ICML 2021 ; Conference date: 18-07-2021 Through 24-07-2021.

Graham, B., Engelcke, M., Van Der Maaten, L., 2018. 3D semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9224–9232. http://dx.doi.org/10.48550/arXiv.1711.10275.

Haara, A., Kangas, A., Tuominen, S., 2019. Economic losses caused by tree species proportions and site type errors in forest management planning. Silva Fenn. 53, 10089. http://dx.doi.org/10.14214/sf.10089.

Hakula, A., Ruoppa, L., Lehtomäki, M., Yu, X., Kukko, A., Kaartinen, H., Taher, J., Matikainen, L., Hyyppä, E., Luoma, V., Holopainen, M., Kankare, V., Hyyppä, J., 2023. Individual tree segmentation and species classification using high-density close-range multispectral laser scanning data. ISPRS Open J. Photogramm. Remote. Sens. 9, 100039. http://dx.doi.org/10.1016/j.ophoto.2023.100039.

Hardenbol, A., Kuzmin, A., Korhonen, L., Korpelainen, P., Kumpula, T., Maltamo, M., Kouki, J., 2021. Detection of aspen in conifer-dominated boreal forests with seasonal multispectral drone image point clouds. Silva Fenn. 55, 10515. http://dx.doi.org/10.14214/sf.10515.

He, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21, 1263–1284. http://dx.doi.org/10.1109/TKDE.2008.239.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. ArXiv Preprint: arxiv.1512.03385 arXiv:1512.03385.

Henrich, J., van Delden, J., Seidel, D., Kneib, T., Ecker, A.S., 2024. Treelearn: A deep learning method for segmenting individual trees from ground-based lidar forest point clouds. Ecol. Inform. 84, 102888. http://dx.doi.org/10.1016/j.ecoinf.2024.102888.

Huuskonen, S., Domisch, T., Finér, L., Hantula, J., Hynynen, J., Matala, J., Miina, J., Neuvonen, S., Nevalainen, S., Niemistö, P., Nikula, A., Piri, T., Siitonen, J., Smolander, A., Tonteri, T., Uotila, K., Viiri, H., 2021. What is the potential for replacing monocultures with mixed-species stands to enhance ecosystem services in boreal forests in fennoscandia? Forest Ecol. Manag. 479, 118558. http://dx.doi.org/10.1016/j.foreco.2020.118558.

Hyyppä, J., Inkinen, M., 1999. Detecting and estimating attributes for single trees using laser scanner. Photogramm. J. Finl. 16, 27–42.

Hyyppä, M., Salolahti, K., Hyyppä, E., Yu, X., Taher, J., Matikainen, L., Lehtomäki, M., Litkey, P., Hakala, T., Kaartinen, H., Hyyppä, J., Kukko, A., 2026. Multispectral airborne laser scanning dataset for tree species classification: MS-ALS-SPECIES. ISPRS Open J. Photogramm. Remote. Sens. 21, 100142. http://dx.doi.org/10.1016/j.ophoto.2026.100142.

Hyyppä, M., Salolahti, K., Hyyppä, E., Yu, X., Taher, J., Matikainen, L., Litkey, P., Lehtomäki, M., Hakala, T., Kaartinen, H., Hyyppä, J., Kukko, A., 2025. Multispectral airborne laser scanning for species classification in espoonlahti (MS-ALS-SPECIES) [dataset]. http://dx.doi.org/10.5281/zenodo.17077256, Zenodo.

Hyyppä, E., Yu, X., Kaartinen, H., Hakala, T., Kukko, A., Vastaranta, M., Hyyppä, J., 2020. Comparison of backpack, handheld, under-canopy UAV, and above-canopy UAV laser scanning for field reference data collection in boreal forests. Remote. Sens. 12, 3327. http://dx.doi.org/10.3390/rs12203327.

Hyyppä, J., Yu, X., Ronnholm, P., Kaartinen, H., Hyyppä, H., 2003. Factors affecting laser-derived object-oriented forest height growth estimation. Photogramm. J. Finl. 18, 16–31.

Iandola, F.N., Moskewicz, M.W., Karayev, S., Girshick, R.B., Darrell, T., Keutzer, K., 2014. Densenet: Implementing efficient convnet descriptor pyramids. http://dx.doi.org/10.48550/arXiv.1404.1869, CoRR, arXiv:1404.1869.

Izenman, A.J., 2013. Linear discriminant analysis. In: Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning. Springer, pp. 237–280.

Jarahizadeh, S., Salehi, B., 2026. Tree-net: A novel deep learning tree detection architecture using UAV LiDAR data. Remote Sens. Environ. 332, 115088. http://dx.doi.org/10.1016/j.rse.2025.115088, URL: https://www.sciencedirect.com/science/article/pii/S0034425725004924.

Jocher, G., Qiu, J., 2024. Ultralytics YOLO11. URL: https://github.com/ultralytics/ultralytics.

Kaartinen, H., Hyyppä, J., Yu, X., Vastaranta, M., Hyyppä, H., Kukko, A., Holopainen, M., Heipke, C., Hirschmugl, M., Morsdorf, F., Næsset, E., Pitkänen, J., Popescu, S., Solberg, S., Wolf, B.M., Wu, J.C., 2012. An international comparison of individual tree detection and extraction using airborne laser scanning. Remote. Sens. 4, 950–974. http://dx.doi.org/10.3390/rs4040950.

Kivinen, S., Koivisto, E., Keski-Saari, S., Poikolainen, L., Tanhuanpää, T., Kuzmin, A., Viinikka, A., Heikkinen, R.K., Pykälä, J., Virkkala, R., Vihervaara, P., Kumpula, T., 2020. A keystone species, european aspen (populus tremula l.), in boreal forests: Ecological role, knowledge needs and mapping using remote sensing. Forest Ecol. Manag. 462, 118008. http://dx.doi.org/10.1016/j.foreco.2020.118008.

Koenig, K., Höfle, B., 2016. Full-waveform airborne laser scanning in vegetation studies—a review of point cloud and waveform features for tree species classification. Forests 7, 198. http://dx.doi.org/10.3390/f7090198.

Korkeala, J., 2026. Normalview. URL: https://gitlab.com/fgi_nls/public/normalview.

Korkeala, J., Muhojoki, J., Kukko, A., Hyyppä, J., 2026. Espoonlahti mobile laser scanning tree species classification [dataset]. http://dx.doi.org/10.5281/zenodo.21039101, Zenodo.

Kouki, J., Arnold, K., Martikainen, P., 2004. Long-term persistence of aspen – a key host for many threatened species – is endangered in old-growth conservation areas in finland. J. Nat. Conserv. 12, 41–52. http://dx.doi.org/10.1016/j.jnc.2003.08.002.

Kozniewski, M., Kolendo, Ł., Ksepko, M., Chmur, S., 2022. Tracking individual scots pine (pinus sylvestris l.) height growth using multi-temporal ALS data from North-Eastern Poland. Remote. Sens. 14, 4170. http://dx.doi.org/10.3390/rs14174170.

Landrieu, L., Obozinski, G., 2017. Cut pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs. SIAM J. Imaging Sci. Vol. 10, 1724–1766, HAL-id: https://hal.science/hal-01306779hal-01306779.

Liang, X., Hyyppä, J., Kukko, A., Kaartinen, H., Jaakkola, A., Yu, X., 2014. The use of a mobile laser scanning system for mapping large forest plots. IEEE Geosci. Remote. Sens. Lett. 11, 1504–1508. http://dx.doi.org/10.1109/LGRS.2013.2297418.

Liu, B., Chen, S., Huang, H., Tian, X., 2022a. Tree species classification of backpack laser scanning data using the PointNet++ point cloud deep learning method. Remote. Sens. 14, 3809. http://dx.doi.org/10.3390/rs14153809.

Liu, M., Han, Z., Chen, Y., Liu, Z., Han, Y., 2021. Tree species classification of lidar data based on 3D deep learning. Measurement 177, 109301. http://dx.doi.org/10.1016/j.measurement.2021.109301.

Liu, B., Huang, H., Su, Y., Chen, S., Li, Z., Chen, E., Tian, X., 2022b. Tree species classification using ground-based lidar data by various point cloud deep learning methods. Remote. Sens. 14, 5733. http://dx.doi.org/10.3390/rs14225733.

Liu, B., Huang, H., Tian, X., Ren, M., 2022c. Individual tree species classification using the pointwise MLP-based point cloud deep learning method. Environ. Sci. Proc. 22, 19. http://dx.doi.org/10.3390/IECF2022-13049.

López, V., Fernández, A., García, S., Palade, V., Herrera, F., 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inform. Sci. 250, 113–141. http://dx.doi.org/10.1016/j.ins.2013.07.007.

Loshchilov, I., Hutter, F., 2016. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983.

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Ma, X., Qin, C., You, H., Ran, H., Fu, Y., 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. http://dx.doi.org/10.48550/arXiv.2202.07123, arXiv:2202.07123.

Ma, Q., Su, Y., Tao, S., Guo, Q., 2018. Quantifying individual tree growth and tree competition using bi-temporal airborne laser scanning data: a case study in the sierra nevada mountains, california. Int. J. Digit. Earth 11, 485–503. http://dx.doi.org/10.1080/17538947.2017.1336578.

Marinelli, D., Paris, C., Bruzzone, L., 2022. An approach based on deep learning for tree species classification in lidar data acquired in mixed forest. IEEE Geosci. Remote. Sens. Lett. 19, 1–5. http://dx.doi.org/10.1109/LGRS.2022.3181680.

McTague, J.P., Weiskittel, A., 2021. Evolution, history, and use of stem taper equations: a review of their development, application, and implementation. Can. J. Forest Res. 51, 210–235. http://dx.doi.org/10.1139/cjfr-2020-0326.

Mokroš, M., Mikita, T., Singh, A., Tomaštík, J., Chudá, J., Wężyk, P., Kuželka, K., Surový, P., Klimánek, M., Zięba-Kulawik, K., Bobrowski, R., Liang, X., 2021. Novel low-cost mobile mapping systems for forest inventories as terrestrial laser scanning alternatives. Int. J. Appl. Earth Obs. Geoinf. 104, 102512. http://dx.doi.org/10.1016/j.jag.2021.102512.

Ørka, H.O., Gobakken, T., Næsset, E., Ene, L., Lien, V., 2012. Simultaneously acquired airborne laser scanning and multispectral imagery for individual tree species identification. Can. J. Remote Sens. 38, 125–138. http://dx.doi.org/10.5589/m12-021.

Pauly, M., Keiser, R., Kobbelt, L.P., Gross, M., 2003. Shape modeling with point-sampled geometry. ACM Trans. Graph. 22, 641–650. http://dx.doi.org/10.1145/882262.882319.

Puliti, S., Lines, E., Müllerová, J., Frey, J., Schindler, Z., Straker, A., Allen, M.J., Lukas, W., Rehush, N., Hristova, H., Murray, B., Calders, K., Terryn, L., Coops, N., Höfle, B., Junttila, S., Krucek, M., Krok, G., Král, K., Levick, S.R., Luck, L., Missarov, A., Mokroš, M., Owen, H., Stereńczak, K., Pitkänen, T., Puletti, N., Saarinen, N., Hopkinson, C., Torresan, C., Tomelleri, E., Weiser, H., Astrup, R., 2025a. FOR-species20K [dataset]. http://dx.doi.org/10.5281/zenodo.13255198, Zenodo.

Puliti, S., Lines, E.R., Müllerová, J., Frey, J., Schindler, Z., Straker, A., Allen, M.J., Winiwarter, L., Rehush, N., Hristova, H., Murray, B., Calders, K., Coops, N., Höfle, B., Irwin, L., Junttila, S., Krůček, M., Krok, G., Král, K., Levick, S.R., Luck, L., Missarov, A., Mokroš, M., Owen, H.J.F., Stereńczak, K., Pitkänen, T.P., Puletti, N., Saarinen, N., Hopkinson, C., Terryn, L., Torresan, C., Tomelleri, E., Weiser, H., Astrup, R., 2025b. Benchmarking tree species classification from proximally sensed laser scanning data: Introducing the FOR-species20K dataset. Methods Ecol. Evol. 16, 801–818. http://dx.doi.org/10.1111/2041-210X.14503.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660. http://dx.doi.org/10.1109/CVPR.2017.16.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 5105–5114.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. ArXiv Preprint: Arxiv.1506.02640 arXiv:1506.02640.

Remm, J., Hanski, I.K., Tuominen, S., Selonen, V., 2017. Multilevel landscape utilization of the siberian flying squirrel: Scale effects on species habitat use. Ecol. Evol. 7, 8303–8315. http://dx.doi.org/10.1002/ece3.3359.

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 J. Photogramm. Remote Sens. 228, 694–722. http://dx.doi.org/10.1016/j.isprsjprs.2025.07.038.

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M., 2008. Towards 3D point cloud based object maps for household environments. Robot. Auton. Syst. 56, 927–941.

Salekin, S., Catalán, C.H., Boczniewicz, D., Phiri, D., Morgenroth, J., Meason, D.F., Mason, E.G., 2021. Global Tree Taper Modelling: A Review of Applications, Methods, Functions, and Their Parameters. Forests 12, 913. http://dx.doi.org/10.3390/f12070913.

Schneider, R., Franceschini, T., Fortin, M., Saucier, J.P., 2018. Climate-induced changes in the stem form of 5 north american tree species. Forest Ecol. Manag. 427, 446–455. http://dx.doi.org/10.1016/j.foreco.2017.12.026.

Seidel, D., Annighöfer, P., Thielman, A., Seifert, Q.E., Thauer, J.H., Glatthorn, J., Ehbrecht, M., Kneib, T., Ammer, C., 2021. Predicting tree species from 3d laser scanning point clouds using deep learning. Front. Plant Sci. Volume 12 - 2021, http://dx.doi.org/10.3389/fpls.2021.635440.

Serafin, J., Grisetti, G., 2015. Nicp: Dense normal based point cloud registration. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 742–749. http://dx.doi.org/10.1109/IROS.2015.7353455.

Sha, S., Liu, Q., Zhang, Y., Yun, T., 2026. FA-CTNet: A geometry-aware deep learning approach for tree species classification from LiDAR point clouds. Remote. Sens. 18, 1311, http://dx.doi.org/10.3390/rs18091311.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6, 1–48. http://dx.doi.org/10.1186/s40537-019-0197-0.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. http://dx.doi.org/10.48550/arXiv.1409.1556, arXiv:1409.1556.

Sun, C., Huang, C., Zhang, H., Chen, B., An, F., Wang, L., Yun, T., 2022. Individual tree crown segmentation and crown width extraction from a heightmap derived from aerial laser scanning data using a deep learning framework. Front. Plant Sci. 13, 914974. http://dx.doi.org/10.3389/fpls.2022.914974.

Sun, P., Yuan, X., Li, D., 2023. Classification of individual tree species using uav lidar based on transformer. Forests 14, 484. http://dx.doi.org/10.3390/f14030484.

Taher, J., Hyyppä, E., Hyyppä, M., Salolahti, K., Yu, X., Matikainen, L., Kukko, A., Lehtomäki, M., Kaartinen, H., Thurachen, S., Litkey, P., Luoma, V., Holopainen, M., Kong, G., Fan, H., Rönnholm, P., Vaaja, M., Polvivaara, A., Junttila, S., Vastaranta, M., Puliti, S., Astrup, R., Kostensalo, J., Myllymäki, M., Kulicki, M., Stereńczak, K., de Paula Pires, R., Valbuena, R., Carbonell-Rivera, J.P., Torralba, J., Chen, Y.C., Winiwarter, L., Hollaus, M., Mandlburger, G., Takhtkeshha, N., Remondino, F., Lisiewicz, M., Kraszewski, B., Liang, X., Chen, J., Ahokas, E., Karila, K., Vezeteu, E., Manninen, P., Näsi, R., Hyyti, H., Pyykkönen, S., Hu, P., Hyyppä, J., 2026. Multispectral airborne laser scanning for tree species classification: A benchmark of machine learning and deep learning algorithms. ISPRS J. Photogramm. Remote Sens. 233, 278–309. http://dx.doi.org/10.1016/j.isprsjprs.2026.01.031.

Tan, M., Le, Q.V., 2020. Efficientnet: Rethinking model scaling for convolutional neural networks. ArXiv Preprint: arxiv.1905.11946 arXiv:1905.11946.

Terven, J., Córdova-Esparza, D.M., Romero-González, J.A., 2023. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr. 5, 1680–1716. http://dx.doi.org/10.3390/make5040083.

Tikkanen, O.P., Martikainen, P., Hyvärinen, E., Junninen, K., Kouki, J., 2006. Red-listed boreal forest species of finland: Associations with forest structure, tree species, and decaying wood. Ann. Zool. Fennici. 43.

Tockner, A., Kraßnitzer, R., Gollob, C., Witzmann, S., Ritter, T., Nothdurft, A., 2025. Tree species classification using intensity patterns from individual tree point clouds. Int. J. Appl. Earth Obs. Geoinf. 139, 104502. http://dx.doi.org/10.1016/j.jag.2025.104502.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4489–4497.

Wang, S., Liu, C., Wei, S., Tang, H., 2025. Urban tree species classification using multisource satellite remote sensing data and street view imagery. Geo-Spatial Inf. Sci. 28, 164–184. http://dx.doi.org/10.1080/10095020.2024.2439380.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. 38, 146. http://dx.doi.org/10.1145/3326362.

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 Sens. Environ. 313, 114367. http://dx.doi.org/10.1016/j.rse.2024.114367.

Winberg, O., Pyörälä, J., Yu, X., Kaartinen, H., Kukko, A., Holopainen, M., Holmgren, J., Lehtomäki, M., Hyyppä, J., 2023. Branch information extraction from norway spruce using handheld laser scanning point clouds in nordic forests. ISPRS Open J. Photogramm. Remote. Sens. 9, 100040. http://dx.doi.org/10.1016/j.ophoto.2023.100040.

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, 1469. http://dx.doi.org/10.3390/rs12091469.

Xi, Z., Hopkinson, C., 2022. 3D graph-based individual-tree isolation (treeiso) from terrestrial laser scanning point clouds. Remote. Sens. 14, 6116. http://dx.doi.org/10.3390/rs14236116.

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. http://dx.doi.org/10.1016/j.rse.2024.114078.

Xiang, B., Wielgosz, M., Puliti, S., Král, K., Kr˚ uček, M., Missarov, A., Astrup, R., 2025. ForestFormer3D: A unified framework for end-to-end segmentation of forest LiDAR 3D point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 24717–24727.

Yang, M., Lee, E., 1999. Segmentation of measured point data using a parametric quadric surface approximation. Computer-Aided Des. 31, 449–457. http://dx.doi.org/10.1016/S0010-4485(99)00042-1.

Yu, X., Hyyppä, J., Holopainen, M., Vastaranta, M., 2010. Comparison of area-based and individual tree-based methods for predicting plot-level forest attributes. Remote. Sens. 2, 1481–1495. http://dx.doi.org/10.3390/rs2061481.

Yu, X., Hyyppä, J., Kaartinen, H., Maltamo, M., 2004. Automatic detection of harvested trees and determination of forest growth using airborne laser scanning. Remote Sens. Environ. 90, 451–462. http://dx.doi.org/10.1016/j.rse.2004.02.001.

Yu, X., Hyyppä, J., Kukko, A., Maltamo, M., Kaartinen, H., 2006. Change detection techniques for canopy height growth measurements using airborne laser scanner data. Photogramm. Eng. Remote Sens. 72, 1339–1348. http://dx.doi.org/10.14358/PERS.72.12.1339.

Yu, X., Hyyppä, J., Litkey, P., Kaartinen, H., Vastaranta, M., Holopainen, M., 2017. Single-sensor solution to tree species classification using multispectral airborne laser scanning. Remote. Sens. 9, 108. http://dx.doi.org/10.3390/rs9020108.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne lidar data filtering method based on cloth simulation. Remote. Sens. 8, 501. http://dx.doi.org/10.3390/rs8060501.

Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2021. Point transformer. In: Proceedings - 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021. Institute of Electrical and Electronics Engineers Inc., pp. 16239–16248. http://dx.doi.org/10.1109/ICCV48922.2021.01595.

Zhao, K., Suarez, J.C., Garcia, M., Hu, T., Wang, C., Londo, A., 2018. Utility of multitemporal lidar for forest and carbon monitoring: Tree growth, biomass dynamics, and carbon flux. Remote Sens. Environ. 204, 883–897. http://dx.doi.org/10.1016/j.rse.2017.09.007.