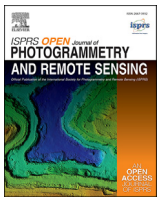
ISPRS Open Journal of Photogrammetry and Remote Sensing 3 (2022) 100011

Contents lists available at ScienceDirect

ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: www.editorialmanager.com/OPHOTO



# Semantic segmentation of point cloud data using raw laser scanner measurements and deep neural networks

Risto Kaijaluoto [a,b,*], Antero Kukko [b,c], Aimad El Issaoui [b], Juha Hyyppä [b], Harri Kaartinen [b,d]

[a] Independent researcher, Helsinki, Finland
[b] Finnish Geospatial Research Institute, 02430, Masala, Finland
[c] Department of Built Environment, Aalto University, Espoo, Finland
[d] Department of Geography, Turku University, Turku, Finland

ARTICLE INFO

*Keywords:*
Deep learning
Convolutional neural networks
Semantic segmentation
Point cloud
Laser scanner
Forest

ABSTRACT

Deep learning methods based on convolutional neural networks have shown to give excellent results in semantic segmentation of images, but the inherent irregularity of point cloud data complicates their usage in semantically segmenting 3D laser scanning data. To overcome this problem, point cloud networks particularly specialized for the purpose have been implemented since 2017 but finding the most appropriate way to semantically segment point clouds is still an open research question. In this study we attempted semantic segmentation of point cloud data with convolutional neural networks by using only the raw measurements provided by a multiple echo detection capable profiling laser scanner. We formatted the measurements to a series of 2D rasters, where each raster contains the measurements (range, reflectance, echo deviation) of a single scanner mirror rotation to be able to use the rich research done on semantic segmentation of 2D images with convolutional neural networks. Similar approach for profiling laser scanner in forest context has never been proposed before. A boreal forest in Evo region near Hämeenlinna in Finland was used as experimental study area. The data was collected with FGI Akhka-R3 backpack laser scanning system, georeferenced and then manually labelled to ground, understorey, tree trunk and foliage classes for training and evaluation purposes. The labelled points were then transformed back to 2D rasters and used for training three different neural network architectures. Further, the same georeferenced data in point cloud format was used for training the state-of-the-art point cloud semantic segmentation network RandLA-Net and the results were compared with those of our method. Our best semantic segmentation network reached the mean Intersection-over-Union value of 80.1% and it is comparable to the 80.6% reached by the point cloud -based RandLA-Net. The numerical results and visual analysis of the resulting point clouds show that our method is a valid way of doing semantic segmentation of point clouds at least in the forest context. The labelled datasets were also released to the research community.

## 1. Introduction

Laser scanning is a measurement technique to determine shape, and possibly the appearance, of real-world objects and environments in the form of a point cloud. The development of point cloud generation optoelectronics has been fast, the first Airborne Laser Scanners (ALS) were built in the early 1990s; the first Mobile Laser Scanners (MLS) were developed in the early 2000s. Today, MLS point coulds can be collected with multiple techniques, for example using hand-held, backpack, and mini-unmanned aerial vehicle (UAV) laser scanning. Lidar-based vision system prototypes targeted and tested at autonomous driving context today use similar technologies to MLS, permitting autonomous perception.

Modern MLS systems can cover large areas and measure huge quantities of data quickly. Processing and getting useful information from large point clouds manually is time consuming and automatic methods are required. Semantic segmentation of the data to useful classes is an important step in utilizing 3D data as it enables users to concentrate on parts of the point clouds they are interested in. Deep learning is one the fastest-growing technologies in analyzing measurement and big data, characterized by deep neural networks (DNN) involving more than two hidden layers. Deep learning has been applied in several image analysis tasks, including semantic segmentation and object detection Kattenborn et al. (2021). Common convolutional architectures require highly regular

* Corresponding author. Finnish Geospatial Research Institute, 02430, Masala, Finland.
*E-mail address:* risto.s.kaijaluoto@gmail.com (R. Kaijaluoto).

https://doi.org/10.1016/j.ophoto.2021.100011
Received 29 September 2021; Received in revised form 2 December 2021; Accepted 7 December 2021
Available online 16 December 2021

input data formats, such as 2D rasters or 3D voxels, to carry out, e.g., weight sharing and other kernel optimizations, but many approaches to utilize them with irregular point cloud data have also been explored Guo et al. (2020).

Point clouds are an important type of geometric data structure in many applications related to mapping and perception/detection. The kinematically obtained range information is converted into point clouds typically using direct georeferencing, i.e., with position and orientation data provided by Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU), by Simultaneous Localization and Mapping (SLAM) methods, or as with our use case, a combination of both. While point clouds are irregular, it is quite common that data before georeferencing is regular, i.e., in sequential order based on the time of measurement.

In this research we investigated whether non-georeferenced, raw mobile/kinematic laser scanner measurements in forest context contain enough information to classify the points for the purpose of modeling and analyzing forest structures. Currently the best-performing Convolutional Neural Networks (CNN) provide the state-of-the-art results in pixel-wise semantic segmentation of images Kattenborn et al. (2021). Since raw laser scanner measurements can easily be formatted into a data structure mimicking raster image, similar well documented CNN based approaches can be applied to classify laser scanner data. By having a classification pipeline, which uses just the raw data provided by the laser scanner, the anticipated errors in the subsequent georeferencing do not affect the classification result. It is also possible to use the pipeline as a preprocessing step to, for example, select or downsample points to speed up further processing by involving only points relevant for an application such as SLAM or tree parameter estimation.

To test this novel idea for profiling laser scanner in forest context, we manually classified two georeferenced 3D point clouds from two boreal forest plots and used this data to train neural networks to do point-wise classification. Point clouds were acquired with FGI's Akhka-R3 backpack MLS system. For feeding the data to the neural networks, the classified data was arranged as scans, where each scan was structured as a 3-dimensional array containing measurements taken during one rotation of the scanner mirror. The first dimension was the sequential pulse number, the second the received echo number pertaining to the pulse and the third dimension had the actual measurements (range, reflectance, and echo deviation) of the echo.

The key aim of this paper was to test whether it is possible to do semantic segmentation of 3D point cloud data by just using raw 2D laser scanner measurements fed into a convolutional neural network. In addition, we investigated whether it can be done in real time, how important are the multi-echo capabilities of the laser scanner we use and how our semantically segmented data could be used in forestry. Finally, we released the manually labelled point clouds to the research community to give people the possibility to advance the research on 3D data in forest context. The dataset is available at https://doi.org/10.23729/8d2d3765-b5a0-4998-82c1-13a6f8bc9de3, please cite this article if you use it in your research.

## 2. Related work

A review and meta-analysis of different deep learning methods used in processing remote sensing data can be found in Ma et al. (2019). In the following we go through some relevant studies done in the field of forestry, the applied neural network architectures, and related research on semantic segmentation of point clouds and images with convolutional neural networks. The latter is included as our method is closely related to semantic segmentation of images.

### 2.1. Semantic segmentation of images with convolutional neural networks

Long et al. (2015) showed that it is possible to adapt deep neural networks utilized for image classification to do pixel-wise semantic segmentation of an image by replacing fully connected layers with convolutional layers. In their architecture, instead of aggregating the information extracted by the convolutional layers over the whole image with the fully connected layers, the hierarchy of features encoded by the convolutional layers on each pixel is used to do pixel-wise classification. To overcome the problem of degrading learning ability of deeper neural networks, He et al. (2016) proposed to modify network architecture by adding shortcut connections to every few convolutional layers. This way the convolutional layers are allowed to learn residual mapping, which is easier for the optimizer and enables the use of deeper networks, which in turn provides improved classification capability. A residual block containing residual mapping can be seen in Fig. 5b. They named their network as ResNet. Wu et al. (2016) applied residual networks to semantic segmentation and similarly reported improved results. Inspired by the fully convolutional network of Long et al. (2015), Ronneberger et al. (2015) introduced the U-net architecture where they increased the amount of both the skip connections between the encoder and decoder sides and the number of convolutional filters on the decoder side. These changes enable the network to propagate the higher semantic level information which has lower resolution to the higher resolution layers near the output side of the network.

Further research on semantic segmentation with convolutional neural networks can be found in Badrinarayanan et al. (2017); Chen et al. (2018); Zoph et al. (2020) and Zhao et al. (2017).

### 2.2. Semantic segmentation of point clouds

Traditionally semantic segmentation of point clouds has relied on a combination of hand-crafted features and a machine learning based classifier. Examples can be found in the works of Hackel et al. (2016) and Munoz et al. (2008).

In principle, if the point cloud is made to conform to a grid like format, for example by voxelizing it, convolutional neural network methods used for semantic segmentation of images could be extended to the 3rd dimension and used in a similar fashion. Unfortunately, in practice it is not feasible, because of the curse of dimensionality and incurring exponential growth of memory requirements combined with the inherent sparsity of the point cloud. Graham et al. (2018) took advantage of the sparsity by proposing the use of submanifold sparse convolutional networks. They use a hash table to increase the performance and decrease the memory usage by only applying convolutional calculations to locations where there are data points.

To get around hand-craft features Qi et al. (2017a) proposed PointNet architecture, which works directly on unordered points and learns a set of functions that describe local information about the input points. The functions can then be maximum pooled to create a global descriptor about the point cloud. A point-wise semantic classification neural network can be trained by concatenating the global descriptor with individual local descriptors. Pointnet++ Qi et al. (2017b) improved the capacity of the Pointnet to capture hierarchy of local structures.

Riegler et al. (2017) employed a grid of shallow octrees as their data format and implemented accompanying functions required to create a CNN, which can take advantage of the data formatting. With grid-octrees they were able to have much higher resolution than with a normal dense CNN while keeping the memory and processing requirements manageable.

Landrieu and Simonovsky (2018) proposed a new architecture called superpoint graph where groups of homogenous points are partitioned together and these superpoints are then used as building blocks of a supergraph. Graph convolutions are utilized to enable fast processing of large point clouds. Hu et al. (2020) proposed the RandLA-Net neural architecture for semantic segmentation of very large point clouds. Their method employs random sampling combined with a novel local feature aggregation module, which encodes local geometry. By stacking these local feature aggregation modules, intervened with random sampling, they can encode information for each point with a large receptive field

with relatively small memory footprint and fast processing time to enable the processing of large point clouds.

3D laser scanners with multiple beams produce point clouds which can easily be transformed to range images, if done in the increments of a single scanner rotation. Regular 2D image segmentation methods can then be employed to do the semantic segmentation. Wu et al. (2018) used spherical projection to transform the point clouds to range images with range, intensity and cartesian coordinate information for each pixel which is then segmented with a SqueezeNet like network combined with conditional random field. Biasutti et al. (2019a) adapted a similar approach but only used range and intensity information with U-Net architecture in their RIU-Net architecture, reaching similar accuracy. In Biasutti et al. (2019b) the authors further improved RIU-Net performance by adding a 3D feature extraction module which learns a descriptor of the local geometry for each range value as the first step of the segmentation pipeline.

Other approaches can be found in Thomas et al. (2019), Zhang et al. (2019b), Lu et al. (2019), Wang et al. (2021a), Boulch et al. (2020), Xu et al. (2020), and Li et al. (2020). A survey of different deep learning approaches on 3D data can be found in Guo et al. (2020).

### 2.3. Semantic segmentation in forestry

Semantic classification and deep learning have been applied already quite extensively for forest applications, namely for detecting forest fires (Zhang et al. (2016); Peng and Wang (2019)), for tree species classification (Hafemann et al. (2014); Guan et al. (2015); Zou et al. (2017); Liu et al. (2019); Xi et al. (2020); Seidel et al. (2021); Hamraz et al. (2019); Dechesne et al. (2017)), for biomass and volume estimation (Zhang et al. (2019a); Narine et al. (2019); Ayrey and Hayes (2018); Liu et al. (2019)), for forest damage assessment (Hamdi et al. (2019)), for detection of stems (Windrim and Bryson (2020)), for individual tree isolation (Wang et al. (2019); Chen et al. (2021)), for forest area or deforestation area determination (Ye et al. (2019); Dong et al. (2019); Sothe et al. (2020); Rizaldy et al. (2018)), and for ground point filtering of ALS data of forested areas (Jin et al. (2020)).

Digumarti et al. (2019) explore variety of convolutional neural network architectures to semantically segment RGB-D images of trees to trunk, branch, twig and leaf classes. Guan et al. (2015) classified tree species from mobile laser scanning data. The processing steps included removal of ground points, tree segmentation using Euclidean distance clustering and voxel-based normalized cut segmentation, and use of waveform representation to model geometric structures of trees. Ten (10) tree species classes were classified with an overall accuracy of 86.1%. Wang (2020) semantically segmented 3D terrestrial laser scanning (TLS) point cloud data into leaf and wood classes. Morel et al. (2020) semantically segmented TLS point clouds of single trees into leaf and wood classes by first enhancing the point cloud by creating local descriptors which encode local geometry and then using PointNet++ inspired neural network model to do classification on the enhanced point cloud reaching mIoU values of 85.59 to 97.07. Krisanski et al. (2021) use PointNet++ inspired neural network model to semantically segment point clouds to terrain, vegetation, coarse woody debris and stem classes. By manually labelling 7 extensive forest point cloud datasets and using them for training of their model they reached excellent result of 95.4% overall accuracy.

## 3. Experiment materials and methods

### 3.1. Applied mobile laser scanner system

The data for this study was collected with the Finnish Geospatial Research Institute (FGI) Akhka-R3 backpack laser scanning system (Fig. 1). The system consists of Riegl VUX-1HA laser scanner, NovAtel Flexpak6 GNSS receiver and Pinwheel 703GGG antenna to observe Global Positioning System (GPS) and GLONASS constellation satellites


**Fig. 1.** Applied Akhka-R3 backpack laser scanner instrument scans 360-degree cross track profiles while GNSS-IMU tracks the platform dynamics during the kinematic mapping.

for positioning complemented with a fibre optical gyroscope and microelectromechanical accelerometer data from NovAtel UIMU-LCI inertial measurement unit for 200Hz trajectory output. The system receiver also serves the lidar unit with PPS time pulses, National Marine Electronics Association (NMEA) messaging and Inertial Navigation System Position Velocity Acceleration (INSPVA) data for real-time trajectory display.

In addition to range, the Riegl VUX-1HA laser scanner provides reflectance and echo deviation information for each received echo and this information was used to help classify the points. The echo deviation value tells how much the received echo shape deviates from the original pulse shape with small values representing small change, typically corresponding to hard surfaces, e.g. tree trunks or building walls. This information can reveal something about the material or angle in which the pulse is reflected. Large values can be due to, for example, the target being slanted or reflections from multiple targets at close range resulting in one, widening echo, if the scanner is unable to differentiate the different targets. Both values are provided by the Riegl data. The scanner can receive multiple echoes for each emitted laser pulse.

For scanning the forest structure we set the scanner mirror to rotate at 100Hz and set the pulse repetition rate to 500 kHz, which works out to around 5000 pulses per full revolution of the mirror. Though not limited, in practice we never detected more than 10 echoes per pulse in our datasets. The scanning geometry can be seen in Fig. 2.

### 3.2. Test site

A boreal forest in Evo region near Hämeenlinna in Finland (61.19 N, 25.11 E) was used as experimental study area. The laser scanning measurements were conducted on three test sites (A, B and C) of size 32 m 32 m. Data from A and B sites were labelled and used for neural network training, validation and testing while C site was only used for additional verification of the method. All sites consisted mainly of pines with some spruce and birches. Descriptive statistics of the sites are provided in Table 1. All point clouds cover a considerably wider area than the test site because of the long range of the laser scanner.

### 3.3. MLS data processing

The GNSS-IMU data from the MLS system was post-processed using Waypoint Inertial Explorer software to incorporate differential GNSS correction using Virtual Reference Station (VRS) base station data (Trimnet) and precise ephemeris and satellite clock data in a multi-pass process with three forward and reverse solutions combined and smoothed in tightly coupled processing to generate the initial trajectory. The lidar data was then calibrated for bore-sight alignment and computed into point clouds based on the trajectory by using Riegl RiProcess software with SDC, MTA and RiWorld modules.

The trajectory was then refined by graph SLAM method where we formulate the trajectory as a graph, detect tree stems in the point cloud and then use the detections of the same tree at different timestamps as additional constraints in the graph. If there are errors in the initial trajectory, the detections of the same tree stem at different timestamps don't align spatially and this information is used by graph optimization to do corrections to the trajectory. Then the new optimized trajectory is used for generating new point cloud. In-depth explanation on the trajectory correction pipeline can be found in Kukko et al. (2017). The tree-based trajectory optimization leaves some errors to the height component of the trajectory. These errors were manually corrected to remove heightwise discrepancies in the point cloud to facilitate manual classification process.

The points in the clouds were then labelled manually to ground, understorey, tree trunk and foliage classes to enable supervised learning. The low vegetation - ground cut was done by extracting the ground points with the Terrascan (Terrasolid, Finland) function. Terrascan has a region growing algorithm that starts growing from the lowest point of a 1-m-by-1-m area and grows to neighboring low points if the angle between them is under a threshold value and then adds all points closer than 2.5 cm to it. Rest of the labelling process was done by drawing a closed polygon of points in different classes on 2D views of the point cloud with CloudCompare software (Girardeau-Montaut (2016)).

The labelling process was extremely time consuming because it is often hard even for a human to discern to which class some group of points belong, especially in areas where the point density is low, or points are very irregularly distributed. The cut between foliage and tree trunk classes and between ground and low foliage classes is also difficult to do. It is often practically impossible to say, where the tree trunk stops and foliage class starts, as there really isn't a clear-cut difference between tree trunks, larger branches, smaller branches and needles or leaves, in particular that holds for deciduous trees. Similarly, with the ground and low vegetation classes, the location of the actual ground level is rather ambiguous in forest with dense low undergrowth vegetation. That is why some level of misclassification between these classes is to be expected. Points further than 70 m from the scanner were not labelled but were included in the dataset to give context to the points with labels. There

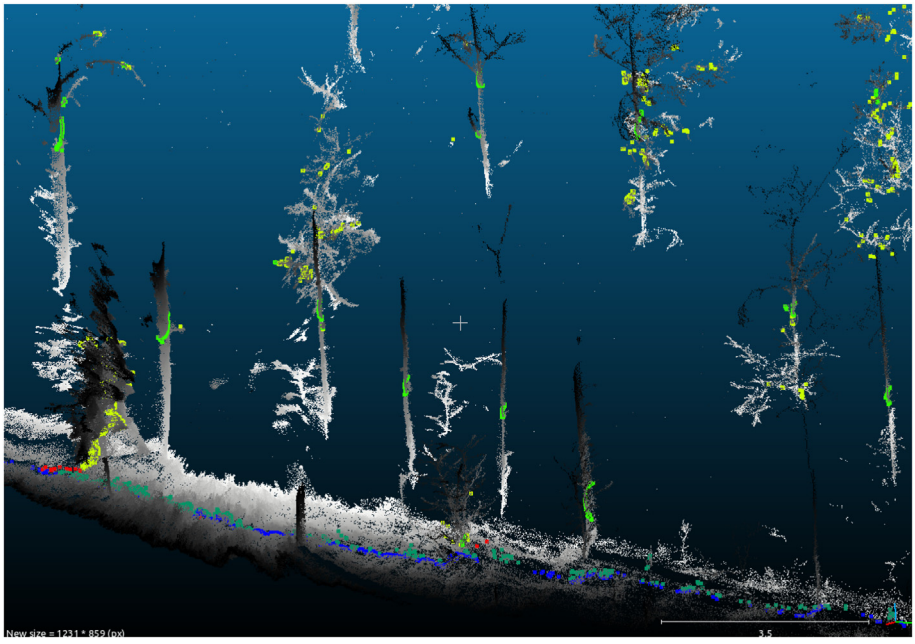


**Fig. 2.** Slice of a point cloud with single scan (mirror rotation) highlighted with colors to show the scanning geometry. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

**Table 1**
Stand descriptive statistics of the test site locations.

| Stand | Basal area (m2/ha) | Volume (m3/ha) | Mean diameter (mm) | Mean height (m) | Height dominant layer (m) | Stems (/ha) | Total biomass (tons/ha) |
|---|---|---|---|---|---|---|---|
| A | 37.49 | 503.65 | 227 | 20.12 | 32.94 | 654 | 213.11 |
| B | 24.82 | 223.25 | 173 | 16.43 | 21.08 | 898 | 120.91 |
| C | 21.48 | 205.26 | 227 | 18.73 | 21.02 | 488 | 102.33 |

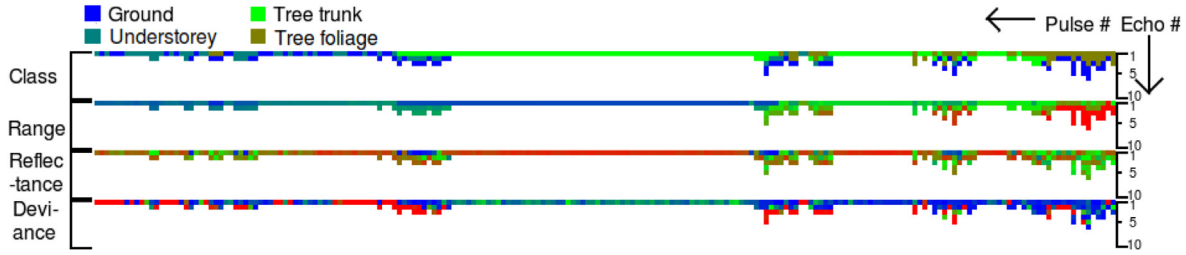



**Fig. 3.** Visualization of the format of the data fed to the network. The class array is the target of the supervised learning, which the neural networks try to learn to predict based on the range, reflectance and deviations arrays. Most pulses get only a few echoes, thus the array is mostly filled with zeros. Only part of a full scan containing 5120 pulses is shown here.

were also humans, some buildings and some points labelled as noise in the point clouds. These points were included in the training but ignored during testing as there was only a small number of them and they were not well spread between the training, validation, and test sets. A total of 202 million points were labelled out of 203 million.

The laser scanner provides timestamp, mirror angle $\varphi$ and echo number value for each point of the point cloud. To facilitate moving between point cloud (Fig. 2) and 2D scan format (Fig. 3), timestamp and mirror angle values are used to calculate scan index and pulse index values for each point of the point cloud according to Algorithm 1.

**Algorithm 1**. Generate scan index and pulse index for each point in a point cloud where points are sorted according to their timestamps

The point cloud from plot A contains 16 725 scans and plot B contains 9889 scans for a total of 26 614 scans. For deep learning purposes the dataset is then be formatted as two arrays with dimensions of 26614x10x5120x3 for the input data and 26614x10x5120x1 for the supervised learning target data. The first dimension is the scan index, second is the echo number, third is the pulse index and the last contains the actual data fields (range, reflectance and echo deviation for input and class labels for target data). A part of single scan is shown in Fig. 3. We can move between the data formats easily as the location of the data in the array corresponds with the indice fields in the point cloud.

The first 2601 scans out of 16 725 scans of plot A were used as test set and the next 2600 scans were used for validation. The remaining 11 524

**Algorithm 1** Generate scan index and pulse index for each point in a point cloud where points are sorted according to their timestamps

**Require:** $0.0 \leq point.\phi < 360.0$

$\phi_\Delta \leftarrow$ angle difference between consecutive laser pulses
$\phi_{preceding} \leftarrow 0.0$
$scan\_index \leftarrow 0$
**for all** $point$ in point cloud **do**
  **if** $point.\phi < \phi_{preceding}$ **then**
    $scan\_index \leftarrow scan\_index + 1$ {New mirror rotation has started}
  **end if**
  $\phi_{preceding} \leftarrow point.\phi$
  $point.scan\_index \leftarrow scan\_index$
  $point.pulse\_index \leftarrow p.\phi/\phi_\Delta$
**end for**

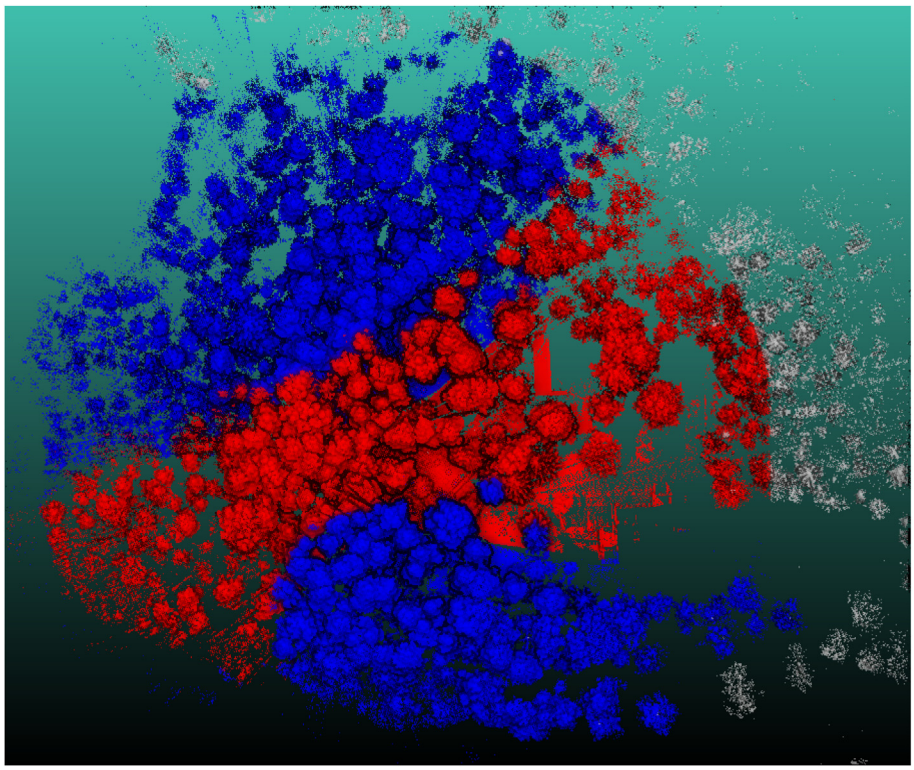


**Fig. 4.** Training-validation-test data split from plot A as viewed from above. Red areas correspond to test data, blue to training and validation (they overlap spatially) and gray areas are unlabeled points. The apparent overlap between test and other data is due to the MLS system not scanning vertically but with an angle. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

scans and all 9889 scans of plot B were used for training. The walking trajectory during data collection formed circles, hence the same area was contained in scans taken at different times. Points in the spatial region of the test dataset were removed from scans in training and validation datasets. 63 million labelled points were eliminated. It is likely that including those points would have improved our training results, but we did this to ensure that absolutely no information was leaked from training to the final semantic segmentation of the test data. This was not done to separate validation and training datasets to conserve the amount of training data. Spatial distribution of the split on plot can be seen in Fig. 4, as plot B is fully used for training it is not shown. The validation dataset was only used for hyperparameter optimization and to decide when to stop training, so even if the network would inadvertently learn from the validation data set this would only give false confidence to the network and likely result in lower performance with the test set.

### 3.4. Deep learning methods used

Keras with Tensorflow 2.3.1 was utilized as the deep learning framework. The general neural network architectures selected for this study were inspired by research on image classification and semantic segmentation of images. We call our segmentation network Laser Scan Segmentation Network or LSSegNet. We tested three different approaches named LSSegNet1 - LSSegNet3 in this paper.

LSSegNet1 follows the architecture of Fully Convolutional Neural Network by Long et al. (2015), but with the plain convolutional layers replaced with ResNet (He et al. (2016)) like residual blocks as it enables deeper networks. The network architecture and an example of a residual block can be found in Fig. 5. LSSegNet2 and LSSegNet3 are based on U-Net (Ronneberger et al. (2015)) architecture. LSSegNet2 6 is similar to the original U-Net but with a different number of convolutional filters. In LSSegNet3 the regular convolutional blocks on the encoder side are replaced by residual blocks.

In the LSSegNet2 and 3 networks (Fig. 6) the left (contracting) side functions as an encoder, where the convolutional layers encode progressively higher-level contextual information, while the pooling layers reduce the resolution to keep the memory requirements manageable. The decoder side combines the information contained in the encoded filters while increasing the resolution to the original resolution to perform the point-wise classification. In the LSSegNet1 network (Fig. 5a) the consecutive residual blocks work as an encoder while the Conv2DTranspose layers and the final convolutional layers carry out the decoding. Some basic details of the networks can be found in Table 2.

In all networks we replaced Dropout (Srivastava et al. (2014)) layers with SpatialDropout2D (Tompson et al. (2015)) layers. Both layer types regularize the network weights and help avoid overfitting, but SpatialDropout2D is more geared towards use with fully convolutional networks.

Values for the number of filters and strides, dropout percentage and the number of residual blocks for each base architecture were selected based on standard hyperparameter optimization with grid search, with target being maximum mean Intersection over Union (IoU) value on validation data set. In addition to the IoU values, processing speed with each hyperparameter combination was also considered. If two networks gave similar results the one with lower number of parameters was selected.

Adam optimizer was utilized as an optimizing backend and was used to minimize the categorical cross-entropy loss. The learning rate was set with a cyclical learning rate scheduler built according to Smith (2017). Their triangular2 scheduling policy was employed and the minimum and maximum learning rates were set with the learning rate range test introduced in the same paper. In triangular2 scheduling policy the learning rate oscillates between the set maximum and minimum values with the maximum value being halved each time it has been reached. The data was augmented by randomly mirroring the scans on the longest dimension. The networks were trained until the loss on validation set stopped decreasing and the network with the lowest validation loss was used to process the test set for final results. Nvidia GeForce GTX 1070 GPU was used to do the computations.

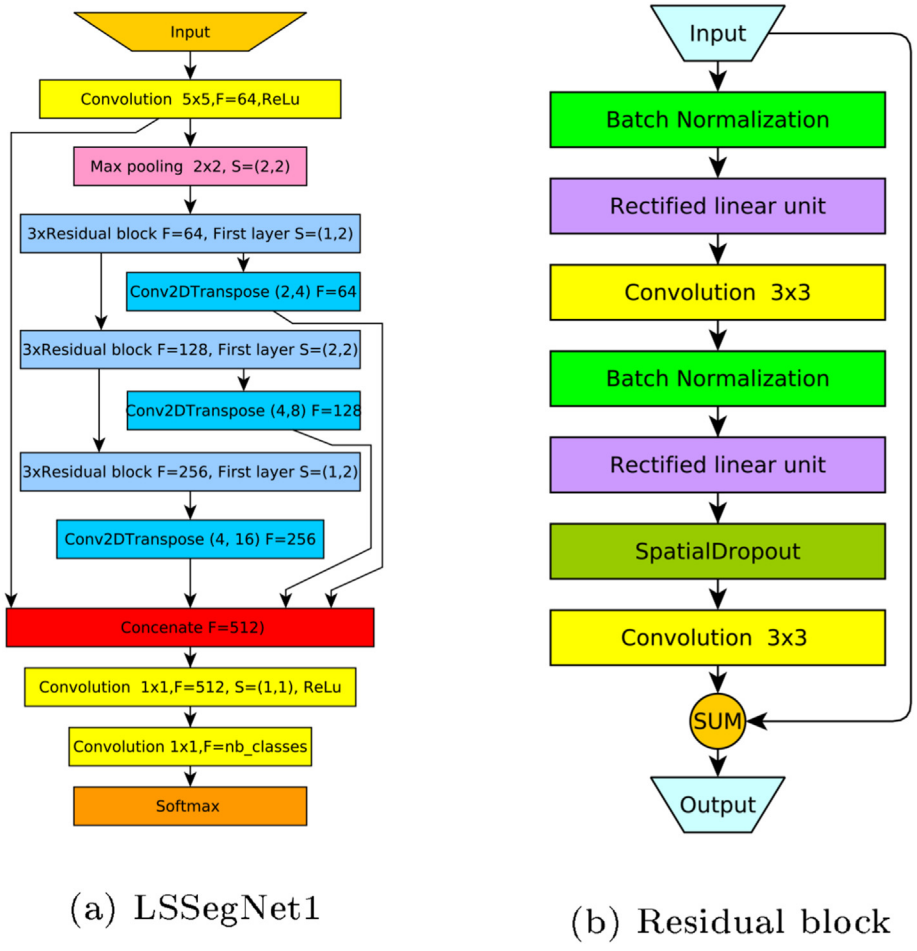



**Fig. 5.** LSSegNet1 with the used parameters showing and one residual block. F represents the number of filters in the layer, S in max pooling is the stride used.

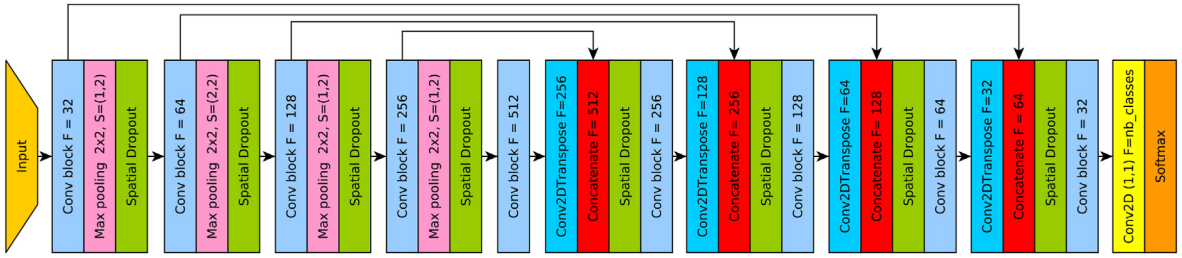



**Fig. 6.** LSSegNet2 architecture used in this research. Each convolution block consists of two consecutive Conv2D - BatchNorm - Relu layer combinations. Arrows on the top are the skip connections. In the selected LSSegNet3 architecture the number of filters is multiplied by 1.5 and each convolutional block on the encoder side is replaced with 3 sequential residual blocks.

**Table 2**
Networks used in the study. Conv layers is the total number of convolutional layers in the network, conv filters is the number of convolutional filters in the first layer. Parameters are the total number of parameters in the network. Inference speed is the number of scans the network can process in a second.

| Network | Conv layers | Conv filters | Parameters | Inference speed (Hz) |
|---|---|---|---|---|
| LSSegNet1 | 24 | 64 | 5 794 310 | 24.51 |
| LSSegNet2 | 23 | 32 | 7 440 295 | 53.48 |
| LSSegNet3 | 43 | 48 | 45 432 355 | 18.02 |

The RandLA-Net neural network architecture by Hu et al. (2020) was used to compare our method with the current state of the art of semantic segmentation of point clouds. RandLA-Net was chosen as it is fast and works well with very large point clouds, such as ours, and has shown to give excellent results in SemanticKitti, Semantic3D and S3DIS semantic segmentation benchmarks, Hu et al. (2020). The same test data was used for our method and RandLA-Net. As RandLA-Net uses point cloud as input, the training - validation data division was done by cutting a spatially continuous segment from the training and validation data point cloud with a similar number of points as in the 2600 scans used for validation in our method (around 13,7 million points). RandLA-Net was run with the RGB fields replaced by range, reflectance and echo deviation fields provided by the scanner, and with smaller batch sizes, to account for our GPU having a lower amount of memory. Otherwise, the default parameters used for S3DIS dataset were used.

To gauge the ability of segmentation networks to extract tree trunks for the purpose of stem curvature and or tree height estimation we manually measured the heights of 15 trees from the bottom of the stem to the topmost point of foliage and compared with the corresponding measured height determined by the points classified as representing tree trunk. The trees in this evaluation were selected from the C plot with varying lateral distance up to 25 m from the scanner.

**Table 3**
Classwise IoU, mean IoU and overall accuracy of the test set processed with different networks. LSSegNet2_RRD uses all of the input fields (Range, reflectance and echo deviation) R is using only range, RR has Range and Reflectanse and RD has Range and echo deviation fields. LSSegNet3_67% contains the results from LSSegNet3 network with 33% of the lowest probability points in each class removed.

| Network name | Ground | Understorey | Tree trunk | Foliage | mIoU | OA |
|---|---|---|---|---|---|---|
| LSSegNet1 | 58.4 | 83.5 | 80.3 | 94.3 | 79.1 | 92.9 |
| LSSegNet2_RRD | 58.4 | 83.0 | 79.4 | 94.0 | 78.7 | 92.6 |
| LSSegNet2_R | 56.2 | 77.2 | 73.7 | 91.7 | 74.7 | 90.3 |
| LSSegNet2_RR | 51.4 | 80.9 | 76.7 | 93.3 | 75.6 | 91.6 |
| LSSegNet2_RD | 59.4 | 83.0 | 76.3 | 90.0 | 78.0 | 92.2 |
| LSSegNet3 | **60.2** | **83.8** | 81.6 | 94.6 | 80.1 | **93.1** |
| RandLA-Net | 59.8 | 79.9 | **87.2** | **95.7** | **80.6** | 92.9 |
| LSSegNet3_67 | 82.6 | 95.2 | 96.5 | 99.3 | 93.4 | 98.3 |

## 4. Results and discussions

RandLA-Net gave slightly better results than our best network LSSegNet3 (Table 3). The results are, however, encouraging as RandLA-Net is one of the highest performing point cloud semantic segmentation methods, and our way of doing segmentation is much more information starved. Still, inspection of the resulting point cloud shows that in practice RandLA-Net results are even better than could be inferred from the small differences in IoU values. With both methods most of the classification errors occur between the ground and understorey classes, understorey and foliage classes and tree trunk and foliage classes, as can be seen in the confusion matrices in Tables 4 and 5. The spatial location of these misclassifications are also found mostly in the boundary areas where the class changes from one to another, as could be expected as there are imperfections in the labelling of the data. The largest difference is that our method labels some of the branches into the tree trunk class instead of the foliage class. Having branches as their own class in the training data could remedy this problem and detecting them would also be useful to estimate forest and tree parameters. Another problem with our method is due to the fact that scans were classified independently. The misclassifications in each scan are not correlated with the misclassifications in the other scans, and as such have spatially more random-like locations after the scans have been georeferenced, which can be seen in Fig. 7. With RandLA-net there is more local context for each classification, which causes the misclassifications to be more spatially aligned (blobs of misclassifications as opposed to random "salt and pepper" like misclassifications with our method). The amount of misclassifications increases with distance from the scanner, as lower point density gives less contextual information. This was seen for the RandLA-Net results as well. Results with LSSegNet1 and 2 networks are similar to LSSegNet3, as can be seen in Figs. 7 and 8, just with more misclassifications as shown in Table 3.

The resulting spatially random misclassifications with our method complicates the use of the resulting point cloud, but this can be remedied by dropping points when the classifier is unsure about the classification. The final softmax layer of the neural network outputs a probability distribution of a point belonging to different classes and we can select only the points where the network is confident in the classification. Table 3 and the confusion matrix in Table 6 shows the results from the LSSegNet3 network after 33% of points with the lowest classification probability in each class were removed. Probability thresholds corresponding to removing 33% of the points were 0.650 (ground), 0.858 (understorey), 0.955 (tree trunks) and 0.949 (foliage). The distribution of the confidence values for tree trunk, foliage and understorey classes were skewed extremely to high confidence values with a long tail for understorey. For ground class the model was much less sure about it predictions with a close to even distribution of confidence values between 0.95 and 0.50 with very few values outside that range.

While many points with the right classification are removed, more importantly we get rid of almost all of the randomly distributed wrong classifications and misclassified branches, which results in point clouds that look much cleaner. We also tried to process the scans in batches of 3 consecutive scans to give more context to semantically segmenting the central scan. We employed 3D convolutions for this but we could not get any improvement on the segmentation results. The approach could be explored more as only few tests were done. There are also other ways to solve this problem such as conditional random fields (CRF) as done by Wu et al. (2018) or k-nearest-neighbors (kNN) search-based consensus voting scheme as done by Milioto et al. (2019) and we will explore them in the future. One downside of our approach is that as it uses raw laser scanner data, the trained model might not work if applied on other laser scanners. Reflectance values will be different on laser scanners employing different wavelength and echo deviation values are also likely laser

**Table 4**
Confusion matrix of the test set processed with LSSegNet3 architecture.

| | | Predicted | | | |
|---|---|---|---|---|---|
| | | Ground | Understorey | Tree trunk | Foliage |
| Actual | Ground | 942 285 | 334 803 | 1 485 | 1 024 |
| | Understorey | 266 162 | 4 742 588 | 36 118 | 115 067 |
| | Tree trunk | 15 609 | 68 666 | 1 963 740 | 187 043 |
| | Foliage | 3 615 | 94 786 | 133 748 | 9 438 231 |

**Table 5**
Confusion matrix of the test set processed with RandLA-Net architecture.

| | | Predicted | | | |
|---|---|---|---|---|---|
| | | Ground | Understorey | Tree trunk | Foliage |
| Actual | Ground | 1 170 626 | 107 677 | 1 191 | 103 |
| | Understorey | 642 285 | 4 274 976 | 16 215 | 226 459 |
| | Tree trunk | 35 846 | 66 075 | 2 061 414 | 7 172 |
| | Foliage | 3 832 | 14 500 | 111 024 | 9 541 024 |

**Table 6**
Confusion matrix of the test set processed with LSSegNet3 architecture with 33 of the lowest probability points in each class removed.

| | | Predicted | | | |
|---|---|---|---|---|---|
| | | Ground | Understorey | Tree trunk | Foliage |
| Actual | Ground | 700 731 | 26 106 | 10 | 21 |
| | Understorey | 114 241 | 3 456 441 | 481 | 2 243 |
| | Tree trunk | 5 965 | 17 501 | 1 424 590 | 19 899 |
| | Foliage | 1 772 | 14 120 | 7 811 | 6 507 299 |

scanner model dependant. On the other hand, segmentation models trained only on range values could work, although different angular resolution can be a challenge. The segmentation model might also not generalize if the scanning geometry is changed. With Akhka-R3 MLS system the scanner is oriented to scan at an angle which undulates several degrees around 26° angle from the vertical when walking as seen in Figs. 1 and 2. If the scanner would be oriented to scan on an angle closer to horizontal the performance of the segmentation model could be degraded.

Tests were done using different pieces of input information removed in order to gauge the importance of an expensive laser scanner, having multi echo capabilities, and more than just range information received per echo. LSSegNet2 gave the highest mIoU results on validation data set and was selected as the network for these tests. Results can be seen in Tables 3 and 7 and in Figs. A.1 and A.2 in the Appendix. It should be noted that the numbers in the particular tables are not comparable as the single echo runs only use a subset of the points. Results indicate that multi echo capability and all three fields give the best results but reveals also that the segmentation can be done using just range information and single echo. Therefore, the method should also be replicable with other laser scanners, albeit with presumably higher error rates. Echo deviation seems to be a more important predictor than the more commonly available reflectance value. This would suggest that having more information about the return, for example the full waveform of the echo, could improve semantic segmentation results. In the single echo case, the reflectance value didn't bring any advantages and surprisingly, the best results were acquired without it. It is possible that the reflectance value is correlated to other reflectance values of the echoes of one laser pulse and is less useful without that information.

In addition to the data with ground truth labels, altogether 16 033 scans from plot C with no ground truth were also processed with the proposed classifier and the resulting semantic segmentation results were

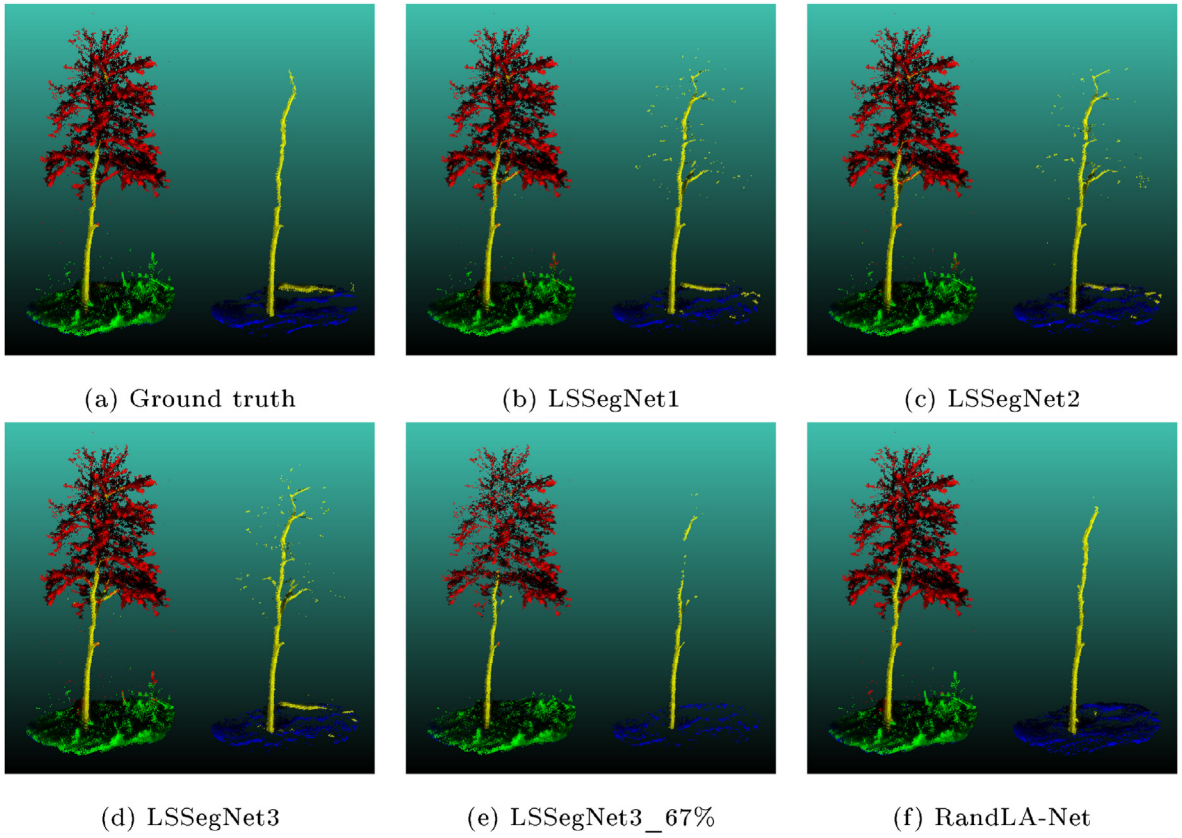



**Fig. 7.** Single tree of the results on the test set including a fallen tree. For each image, the right-hand side has foliage and understorey points removed with only ground and tree trunk classes present. Red: Foliage, Yellow: Tree trunk, Green: Understorey, Blue: Ground. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

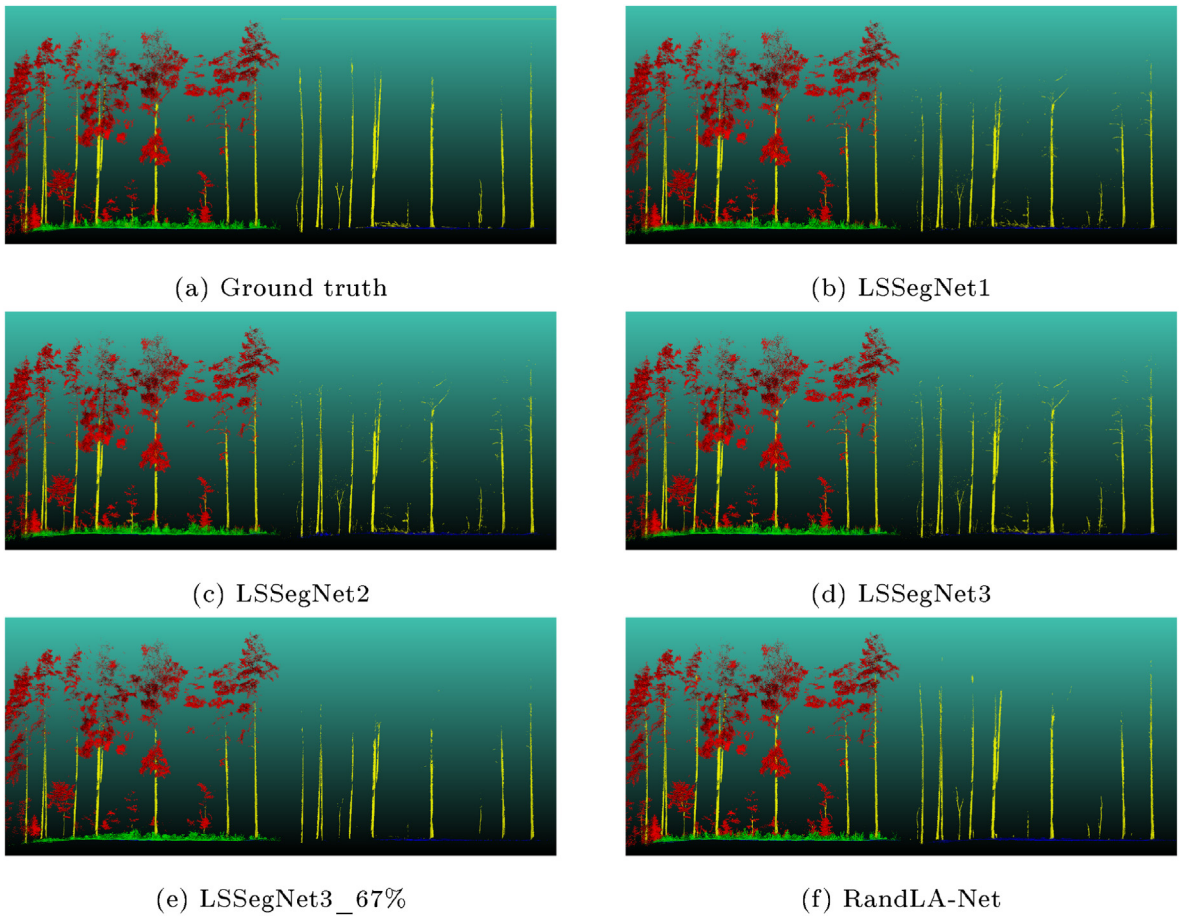



**Fig. 8.** Slice of the point cloud of the results on the test set. Rows from the top ground truth, RandLA-Net, LSSegNet3, LSSegNet3 with 33% of points with the lowest probability in each class removed. On the right-hand side only ground and tree trunk classes are left.

visually compared with the results on the test part of the training dataset. The results look similar (see Fig. A.3 in Appendix) to the ones from the test and we can be confident that the neural network generalizes and can process data from other forest plots. The figure also shows that a point cloud with georeferencing errors causes problems for methods such as RandLA-Net, which utilize the whole point cloud. Trajectory errors in the height component cause multiple levels of ground. RandLA-Net shows a tendency to misclassify that to foliage class. This shows one advantage of doing the semantic segmentation on raw laser scanner data as the result is not dependent on the success of the georeferencing. Splitting the point cloud according to measurement time to slices and processing the slices independently with point cloud based segmentation method could also overcome these problems caused by errors in georeferencing.

Our network can semantically segment laser scans at 53.48Hz–18.03Hz, depending on the network (Table 2) and, as the scanner recorded them at 100Hz, we cannot process scans as fast as they are measured. However, the Geforce 1070 GPU used in the computation is aimed at gaming instead of deep learning and is over 5 years old as of now. Newer generation GPU models easily have enough performance to classify points in real time as they are measured by the scanner, at least with the LSSegNet2 network and likely with the others, too.

**Table 8**
Portion of the height of the tree trunk extracted by the semantic segmentation network as compared to the full height of the tree. 15 trees with varying distance from the scanner were used from the plot C.

| | Tree height (m) | Detected portion | |
|---|---|---|---|
| | | LSSegNet3 | LSSegNet3 67% |
| Mean | 20.0 | 72.6% | 60.4% |
| Std | 1.6 | 2.9% | 5.2% |
| Min | 16.6 | 66.4% | 48.2% |
| Max | 23.0 | 75.9% | 68.0% |

**Table 7**
Classwise IoU, mean IoU and overall accuracy of the test set processed with different networks using only single echo per pulse. LSSegNet2_1E_RRD uses all of the input fields (Range, reflectance and echo deviation) R is using only range, RR has Range and Reflectanse and RD has Range and echo deviation fields. LSSegNet2_1E_Res3_67 contains the results from LSSegNet2_1E_RRD network with 33% of the lowest probability points in each class removed. LSSegNet2_ME_RRD row has the results on the same points extracted from results processed with multiple echoes for comparison.

| Network name | Ground | Understorey | Tree trunk | Foliage | mIoU | OA |
|---|---|---|---|---|---|---|
| LSSegNet2_1E_RRD | 60.5 | 82.5 | 78.5 | 92.3 | 78.5 | 90.5 |
| LSSegNet2_1E_R | 57.4 | 79.5 | 81.2 | **93.5** | 77.9 | 89.8 |
| LSSegNet2_1E_RR | 56.8 | 79.3 | **82.0** | **93.5** | 77.9 | 89.8 |
| LSSegNet2_1E_RD | **61.0** | 82.7 | 80.1 | 92.5 | 79.1 | 90.7 |
| LSSegNet2_ME_RRD | 58.4 | **83.9** | 81.5 | 93.1 | **79.2** | **91.1** |
| LSSegNet2_1E_RRD 67 | 78.4 | 91.9 | 94.0 | 98.7 | 90.8 | 96.5 |

Wang (2020) partly focuses on the semantic segmentation of 3D point cloud data into forestry components. According to Wang, validation using simulated data resulted in an overall accuracy of 87.7% for classification to leaf and wood classes. Krisanski et al. (2021) segmented forest point clouds to terrain, vegetation, coarse woody debris and stem classes with corresponding IoU values of 89.1%, 93.6%, 40.7%, 91.3%. As the dataset and classification targets are different, we shouldn't put too much weight on the specific numbers but it is encouraging that our results are in the same ballpark.

Separation between ground, understorey, tree trunk and foliage classes would be useful as a pre-processing method for all forestry application development. Especially in the field of mobile laser scanning, such a technique would be highly useful as there are multiple challenges in processing under-canopy MLS data. For example, pre-classification of point cloud data on-the-fly would be highly useful for improving and assisting in SLAM approaches (Lehtola et al. (2019); Kukko et al. (2017)). When doing SLAM with mobile laser scanner data, removing points corresponding to understorey and foliage can be advantageous as matching measurements acquired of those areas at different times can be prone to mismatch errors. They have extremely complex shapes, occlusions, and even light wind will move them, which results in extreme difficulty to find and match the exact same locations and objects observed at different times. Thus, having the data classified in real time can be useful, in addition to SLAM also in reducing saved or wirelessly relayed data volume. Data volumes recorded by modern laser scanners can be large and generally not all of it is necessary. For example, in case we are calculating forest parameters (e.g., number of trees, stem diameter breast heights (DBH) and volumes), it is much simpler if we only have the measurements corresponding to trees (tree trunk and foliage class) as input for such a process.

Results for trying to extract tree trunks for stem curvature and tree height estimation from the segmented point cloud are shown in Table 8. Generally, the lower percentages achieved were from trees furthest from the scanner with increased traverse of the laser light through foliage and, thus, reduced visibility. Also, trees close to the scanner are measured from below, while trees further away are covered with pulses at more slant angles and with reduced spatial density (effect of angular resolution, also turning reduces the spatial density of the point). Stems could be extracted up to 76% of relative height.

In comparison to earlier works (e.g., Hyyppä et al. (2020); Wang et al. (2021b)), the detection power for tree stem points with the proposed method seems to give remarkably good results. In Hyyppä et al. (2020), the focus was on finding good-quality arcs determining the stem curvature. Typically, the quality of arcs dropped at a relative height of 40% (the ratio between the extracted stem curvature maximum height and the tree height). In Wang et al. (2021b), the corresponding relative height was 64% in a very sparse forest. Corresponding relative mean heights of 60% and 73% (Table 8) were obtained in this study. The corresponding densities of the forests were as follows: 200 stems per hectare in Wang et al. (2021b), 410–420 in Hyyppä et al. (2020), and 488 in this study. Stem curvature is the most important quality related parameter needed in harvesting in deciding the optimal cutting of a trunk. An additional piece of useful information is the amount of branches, especially living branches, surrounding each height layer of the tree. The proposed method also seems to provide (Fig. 8) a valuable contribution to this information. The abundance of dead wood is considered to be an indicator of forest biodiversity since many threatened species are dependent on decaying wood as a habitat. Our algorithm is able to provide prior information when looking for dead trees either laying on the forest floor or standing up. This research was done with data collected in forests, but a similar approach could also be applied on built environments on objects such as utility poles and portals, traffic signs, road objects and building features, etc.

## 5. Conclusions

In this paper we showed that it is possible to semantically segment 3D data measured by a mobile laser scanner with deep neural networks by just using raw (non-georeferenced) 2D laser scanner measurements in 2D raster format. We obtained 0.5 %-unit lower mean Intersection over Union value when classifying forest point cloud data to ground, understorey, tree trunk and foliage classes with our method (80.1%) as compared with the state-of-the-art point cloud based RandLA-Net (80.6%). The results are promising, considering that our raw laser scanner measurements based method has much less contextual information for classifying each point. Our method tended to classify some branches to the tree trunk class instead of foliage class, since large branches appear pretty similar in surface texture, reflectivity, and echo properties. Dropping unsure points was also found to be a good way to reduce misclassification. This approach is acceptable in mobile laser scanning applications as the distances from the scanner are relatively short and the point density is often very high, providing redundant data.

By doing the semantic segmentation in increments of single scans (mirror rotations) on raw measurements our method also avoids problems caused by possible errors in the trajectory of the MLS system. Errors in trajectory cause spatial discrepancies such as duplicates of objects and blurring of the geometry in the point cloud which can hamper automatic interpretation of it.

In comparison to earlier works, the detection power to extract high stem points with the proposed classification method seems to give remarkably good results. This has significance in pursuit for automated timber volume and stem quality estimates. 3D laser mapping systems enhanced with semantic segmentation pipeline such as ours will also help in understanding the complexity of terrain and forest structures with applications in, for example research on carbon sequestration by forests or other forest ecosystems research.

This paper shows that classification of raw laser scanning data is feasible, fast (real-time) and provides potential to speed up, e.g., SLAM process to correct the data for geometric errors by reducing the search space and adding semantics to the process. Classification also reduces the processing power needed by permitting only classes of interest to be selected for data processing, map compilation, and other interpretation and modeling tasks. Further, an increase in performance would contribute to implementation of edge computation solutions for time critical applications such as robotics and UAV scene.

## Author contributions

R.K acted as the sole first author and developed the semantic segmentation processing chain, created all the algorithms, processed the data to obtain the results, and wrote the majority of the article. A.K. and H.K. made the measurements. A.K. was the major supervisor during the whole process in years 2016–2021. A.E.I helped R.K. in the labelling process. A.K., J.H, and H.K. all participated in supervising the work and writing of the article. H.K. and J.H. provided funding for the research.

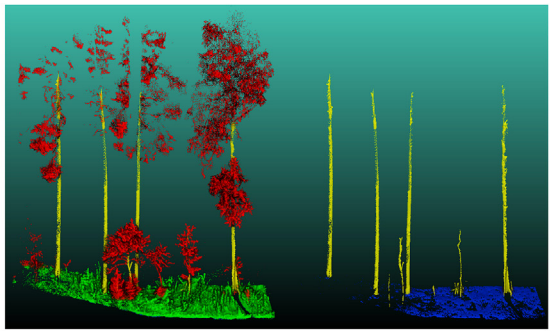
## Declaration of competing interest

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

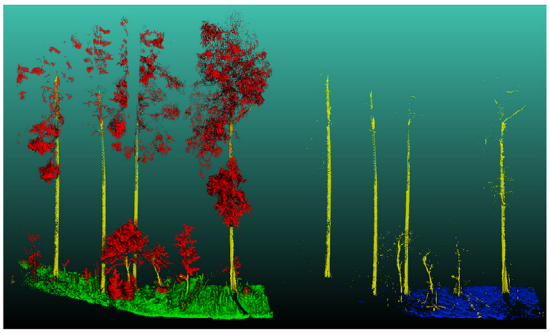
## Acknowledgements

This work was supported in part by the Strategic Research Council at the Academy of Finland project "Competence Based Growth Through Integrated Disruptive Technologies of 3D Digitalization, Robotics, Geospatial Information and Image Processing/Computing - Point Cloud Ecosystem" (293 389, 314 312), and Academy of Finland projects "Estimating Forest Resources and Quality related Attributes Using Automated Methods and Technologies" (334 830, 334 829), "Monitoring and understanding forest ecosystem cycles" (334 060), "Lidar-based energy efficient ICT solutions" (319 011). Publication is also part of Finnish Research Flagships Forest-Human-Machine Interplay (UNITE)

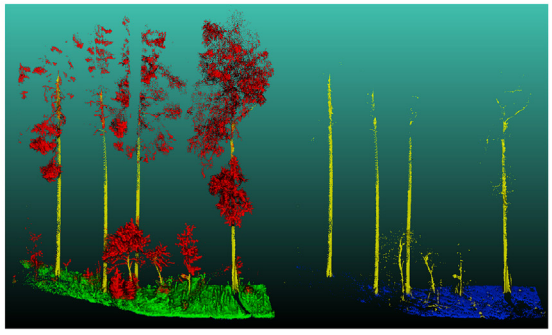
## Appendix



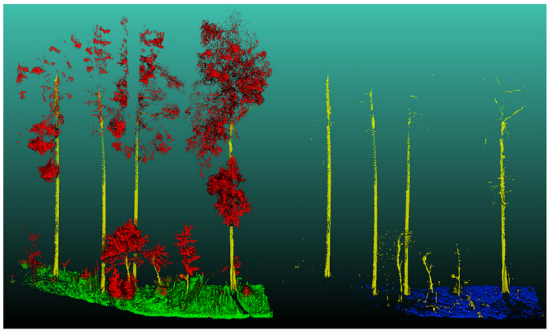
(a) Ground truth



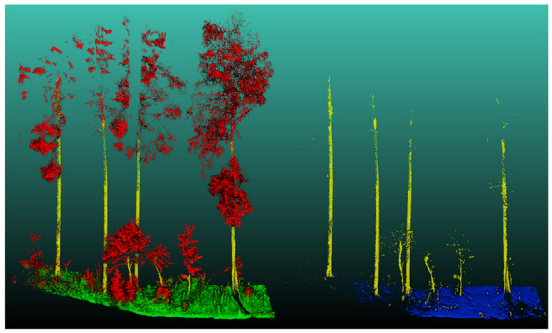
(b) LSSegNet1



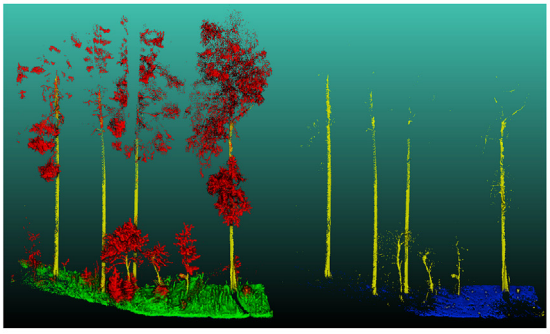
(c) LSSegNet2



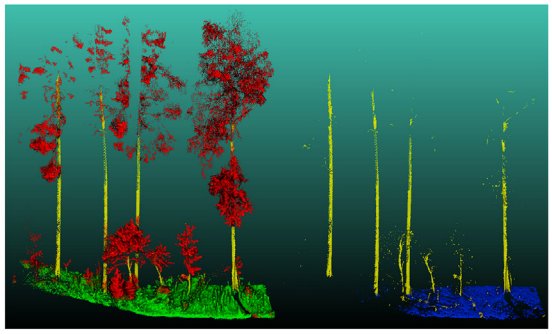
(d) LSSegNet3



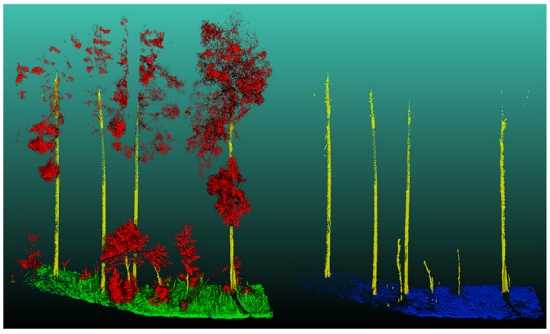
(e) LSSegNet2 with only range information



(f) LSSegNet2 with only range and reflectance information

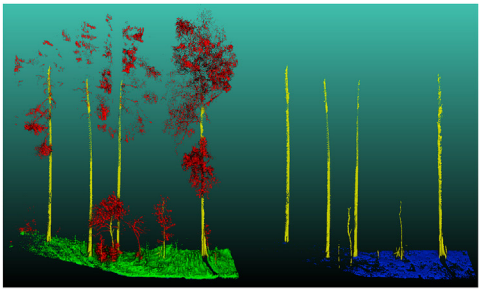


(g) SSegNet2 with only range and echo deviation information

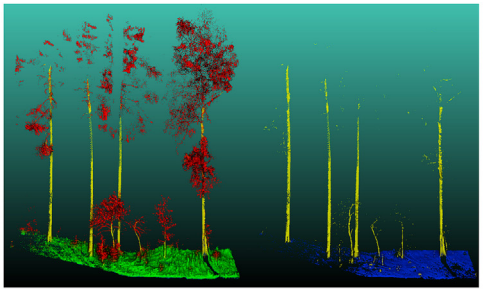


(h) RandLA-Net

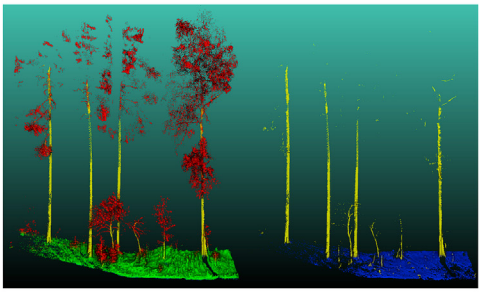
**Fig. A.1.** Semantic segmentation results of the A plot test set.



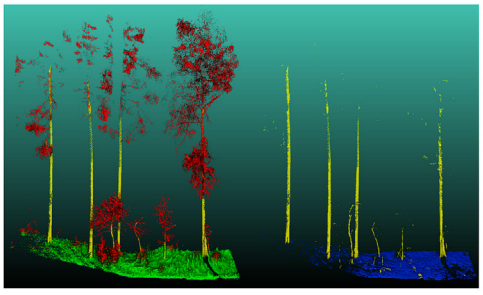
(a) Ground truth



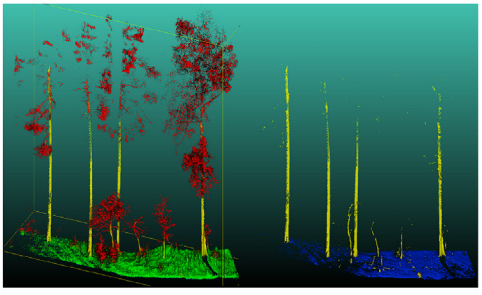
(b) LSSegNet2 with only range information



(c) LSSegNet2 with only range and reflectance information

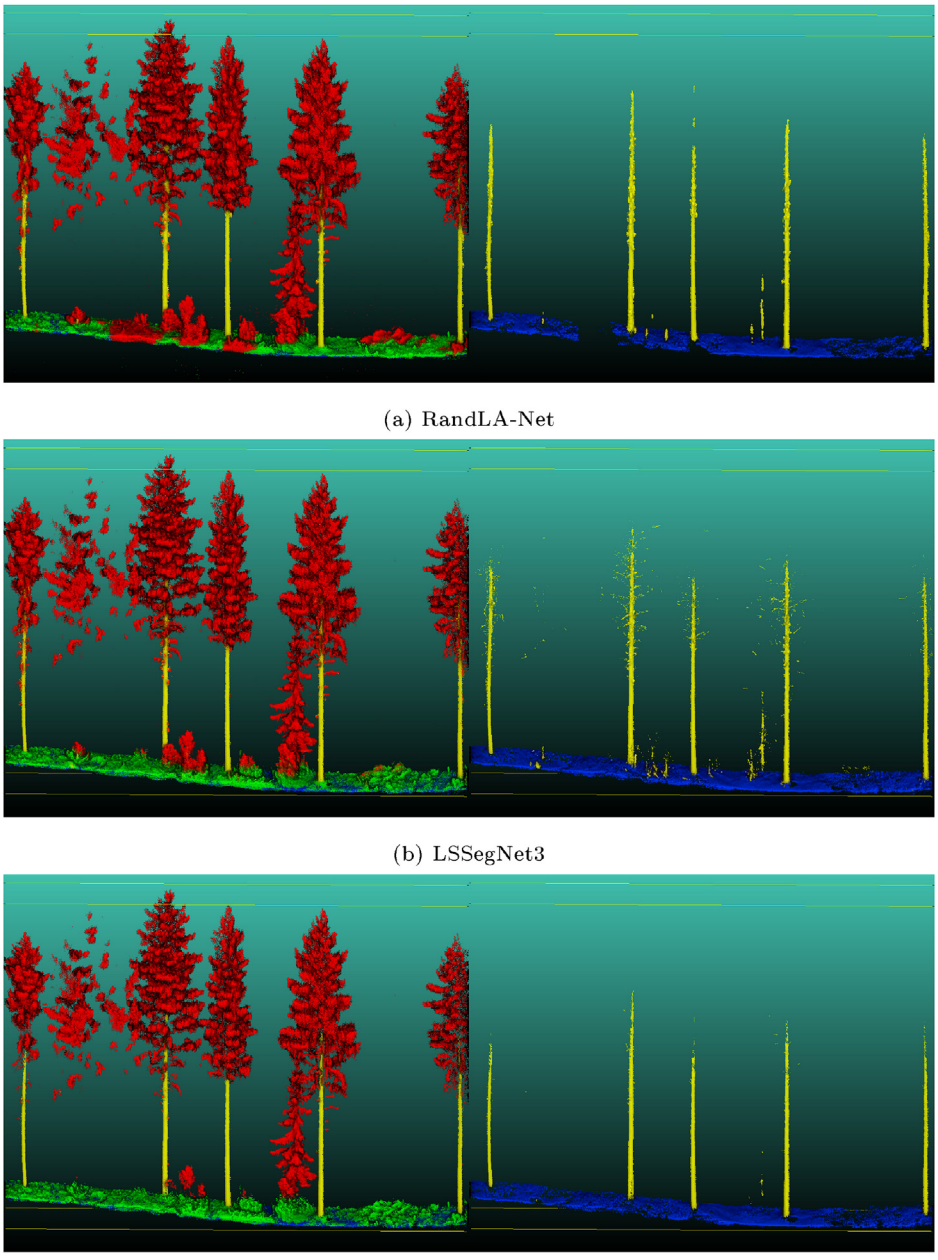


(d) LSSegNet2 with only range and echo deviation information



(e) LSSegNet2

**Fig. A.2.** Semantic segmentation results of the A plot test set with only single echo used



(a) RandLA-Net

(b) LSSegNet3

(c) LSSegNet3_67%

**Fig. A.3.** Semantic segmentation results of the C forest plot shown as approximately 5 m wide cross section taken from the classified point cloud; the second stem from the left was out of this slice, but some foliage remains visible.

## References


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

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2481–2495.

Biasutti, P., Bugeau, A., Aujol, J.F., Brédif, M., 2019a. Riu-net: Embarrassingly Simple Semantic Segmentation of 3d Lidar Point Cloud arXiv preprint arXiv:1905.08748.

Biasutti, P., Lepetit, V., Aujol, J.F., Brédif, M., Bugeau, A., 2019b. Lu-net: an efficient network for 3d lidar point cloud semantic segmentation based on end-to-end-learned 3d features and u-net. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 0–0.

Boulch, A., Puy, G., Marlet, R., 2020. Lightconvpoint: Convolution for Points arXiv preprint arXiv:2004.04462.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 801–818.

Chen, X., Jiang, K., Zhu, Y., Wang, X., Yun, T., 2021. Individual tree crown segmentation directly from uav-borne lidar data using the pointnet of deep learning. Forests 12, 131.

Dechesne, C., Mallet, C., Le Bris, A., Gouet-Brunet, V., 2017. Semantic segmentation of forest stands of pure species combining airborne lidar data and very high resolution multispectral imagery. ISPRS J. Photogrammetry Remote Sens. 126, 129–145.

Digumarti, S.T., Schmid, L.M., Rizzi, G.M., Nieto, J., Siegwart, R., Beardsley, P., Cadena, C., 2019. An approach for semantic segmentation of tree-like vegetation. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE, pp. 1801–1807.

Dong, L., Du, H., Mao, F., Han, N., Li, X., Zhou, G., Zheng, J., Zhang, M., Xing, L., Liu, T., et al., 2019. Very high resolution remote sensing imagery classification using a fusion of random forest and deep learning technique—subtropical area for example. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 13, 113–128.

Girardeau-Montaut, D., 2016. Cloudcompare. France: EDF R&D Telecom ParisTech.

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.

Guan, H., Yu, Y., Ji, Z., Li, J., Zhang, Q., 2015. Deep learning-based tree classification using mobile lidar data. Remote Sensing Letters 6, 864–873.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep Learning for 3d Point Clouds: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Hackel, T., Wegner, J.D., Schindler, K., 2016. Fast semantic segmentation of 3d point clouds with strongly varying density. ISPRS.Ann. Photogram.remote sensing.Spatial Inf. Sci. 3, 177–184.

Hafemann, L.G., Oliveira, L.S., Cavalin, P., 2014. Forest species recognition using deep convolutional neural networks. In: 2014 22Nd International Conference on Pattern Recognition. IEEE, pp. 1103–1107.

Hamdi, Z.M., Brandmeier, M., Straub, C., 2019. Forest damage assessment using deep learning on high resolution remote sensing data. Rem. Sens. 11, 1976.

Hamraz, H., Jacobs, N.B., Contreras, M.A., Clark, C.H., 2019. Deep learning for conifer/deciduous classification of airborne lidar 3d point clouds representing individual trees. ISPRS J. Photogrammetry Remote Sens. 158, 219–230.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. Randla-net: efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11108–11117.

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. Rem. Sens. 12, 3327.

Jin, S., Su, Y., Zhao, X., Hu, T., Guo, Q., 2020. A point-based fully convolutional neural network for airborne lidar ground point filtering in forested environments. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 13, 3958–3974.

Kattenborn, T., Leitloff, J., Schiefer, F., Hinz, S., 2021. Review on convolutional neural networks (cnn) in vegetation remote sensing. ISPRS J. Photogrammetry Remote Sens. 173, 24–49.

Krisanski, S., Taskhiri, M.S., Gonzalez Aracil, S., Herries, D., Turner, P., 2021. Sensor agnostic semantic segmentation of structurally diverse and complex forest point clouds using deep learning. Rem. Sens. 13, 1413.

Kukko, A., Kaijaluoto, R., Kaartinen, H., Lehtola, V.V., Jaakkola, A., Hyyppä, J., 2017. Graph slam correction for single scanner mls forest data under boreal forest canopy. ISPRS J. Photogrammetry Remote Sens. 132, 199–209.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4558–4567.

Lehtola, V.V., Lehtomäki, M., Hyyti, H., Kaijaluoto, R., Kukko, A., Kaartinen, H., Hyyppä, J., 2019. Preregistration classification of mobile lidar data using spatial correlations. IEEE Trans. Geosci. Rem. Sens. 57, 6900–6915.

Li, W., Wang, F.D., Xia, G.S., 2020. A geometry-attentional network for als point cloud classification. ISPRS J. Photogrammetry Remote Sens. 164, 26–40.

Liu, J., Wang, X., Wang, T., 2019. Classification of tree species and stock volume estimation in ground forest images using deep learning. Comput. Electron. Agric. 166, 105012.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

Lu, J., Liu, T., Luo, M., Cheng, H., Zhang, K., 2019. Pfcn: a fully convolutional network for point cloud semantic segmentation. Electron. Lett. 55, 1088–1090.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: a meta-analysis and review. ISPRS J. Photogrammetry Remote Sens. 152, 166–177.

Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019. Rangenet++: fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 4213–4220.

Morel, J., Bac, A., Kanai, T., 2020. Segmentation of unbalanced and in-homogeneous point clouds and its application to 3d scanned trees. Vis. Comput. 36, 2419–2431.

Munoz, D., Vandapel, N., Hebert, M., 2008. Directional Associative Markov Network for 3-d Point Cloud Classification.

Narine, L.L., Popescu, S.C., Malambo, L., 2019. Synergy of icesat-2 and landsat for mapping forest aboveground biomass with deep learning. Rem. Sens. 11, 1503.

Peng, Y., Wang, Y., 2019. Real-time forest smoke detection using hand-designed features and deep learning. Comput. Electron. Agric. 167, 105029.

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.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space arXiv preprint arXiv:1706.02413.

Riegler, G., Osman Ulusoy, A., Geiger, A., 2017. Octnet: learning deep 3d representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3577–3586.

Rizaldy, A., Persello, C., Gevaert, C., Oude Elberink, S., Vosselman, G., 2018. Ground and multi-class classification of airborne laser scanner point clouds using fully convolutional networks. Rem. Sens. 10, 1723.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

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. 12, 141.

Smith, L.N., 2017. Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp. 464–472.

Sothe, C., De Almeida, C., Schimalski, M., Liesenberg, V., La Rosa, L., Castro, J., Feitosa, R., 2020. A comparison of machine and deep-learning algorithms applied to multisource data for a subtropical forest area classification. Int. J. Rem. Sens. 41, 1943–1969.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. Kpconv: flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6411–6420.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C., 2015. Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–656.

Wang, D., 2020. Unsupervised semantic and instance segmentation of forest point clouds. ISPRS J. Photogrammetry Remote Sens. 165, 86–97.

Wang, G., Xu, G., Wu, Q., Wu, X., 2021a. Two-stage point cloud super resolution with local interpolation and readjustment via outer-product neural network. J. Syst. Sci. Complex. 34, 68–82.

Wang, J., Chen, X., Cao, L., An, F., Chen, B., Xue, L., Yun, T., 2019. Individual rubber tree segmentation based on ground-based lidar data and faster r-cnn of deep learning. Forests 10, 793.

Wang, Y., Kukko, A., Hyyppä, E., Hakala, T., Pyörälä, J., Lehtomäki, M., El Issaoui, A., Yu, X., Kaartinen, H., Liang, X., et al., 2021b. Seamless integration of above-and under-canopy unmanned aerial vehicle laser scanning for forest investigation. Forest Ecosystems 8, 1–15.

Windrim, L., Bryson, M., 2020. Detection, segmentation, and model fitting of individual tree stems from airborne laser scanning of forests using deep learning. Rem. Sens. 12, 1469.

Wu, B., Wan, A., Yue, X., Keutzer, K., 2018. Squeezeseg: convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 1887–1893.

Wu, Z., Shen, C., Hengel, A.v.d., 2016. High-performance Semantic Segmentation Using Very Deep Fully Convolutional Networks arXiv preprint arXiv:1604.04339.

Xi, Z., Hopkinson, C., Rood, S.B., Peddle, D.R., 2020. See the forest and the trees: effective machine and deep learning algorithms for wood filtering and tree species classification from terrestrial laser scanning. ISPRS J. Photogrammetry Remote Sens. 168, 1–16.

Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M., 2020. Squeezesegv3: spatially-adaptive convolution for efficient point-cloud segmentation. In: European Conference on Computer Vision. Springer, pp. 1–19.

Ye, L., Gao, L., Marcos-Martinez, R., Mallants, D., Bryan, B.A., 2019. Projecting Australia's forest cover dynamics and exploring influential factors using deep learning. Environ. Model. Software 119, 407–417.

Zhang, L., Shao, Z., Liu, J., Cheng, Q., 2019a. Deep learning based retrieval of forest aboveground biomass from combined lidar and landsat 8 data. Rem. Sens. 11, 1459.

Zhang, Q., Xu, J., Xu, L., Guo, H., 2016. Deep convolutional neural networks for forest fire detection. In: 2016 International Forum on Management, Education and Information Technology Application. Atlantis Press.
Zhang, Z., Hua, B.S., Yeung, S.K., 2019b. Shellnet: efficient point cloud convolutional neural networks using concentric shells statistics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1607–1616.
Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890.
Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.V., 2020. Rethinking Pre-training and Self-Training arXiv preprint arXiv:2006.06882.
Zou, X., Cheng, M., Wang, C., Xia, Y., Li, J., 2017. Tree classification in complex forest point clouds based on deep learning. Geosci. Rem. Sens. Lett. IEEE 14, 2360–2364.