



Article

Transport Infrastructure Management Based on LiDAR Synthetic Data: A Deep Learning Approach with a ROADSENSE Simulator

Lino Comesaña-Cebral , Joaquín Martínez-Sánchez , Antón Nuñez Seoane and Pedro Arias

CINTECX, Applied Geotechnology Group, Universidade de Vigo, 36310 Vigo, Spain

* Correspondence: linojose.comesana@uvigo.gal

Abstract: In the realm of transportation system management, various remote sensing techniques have proven instrumental in enhancing safety, mobility, and overall resilience. Among these techniques, Light Detection and Ranging (LiDAR) has emerged as a prevalent method for object detection, facilitating the comprehensive monitoring of environmental and infrastructure assets in transportation environments. Currently, the application of Artificial Intelligence (AI)-based methods, particularly in the domain of semantic segmentation of 3D LiDAR point clouds by Deep Learning (DL) models, is a powerful method for supporting the management of both infrastructure and vegetation in road environments. In this context, there is a lack of open labeled datasets that are suitable for training Deep Neural Networks (DNNs) in transportation scenarios, so, to fill this gap, we introduce ROADSENSE (Road and Scenic Environment Simulation), an open-access 3D scene simulator that generates synthetic datasets with labeled point clouds. We assess its functionality by adapting and training a state-of-the-art DL-based semantic classifier, PointNet++, with synthetic data generated by both ROADSENSE and the well-known HELIOS++ (HEildelberg LiDAR Operations Simulator). To evaluate the resulting trained models, we apply both DNNs on real point clouds and demonstrate their effectiveness in both roadway and forest environments. While the differences are minor, the best mean intersection over union (MIoU) values for highway and national roads are over 77%, which are obtained with the DNN trained on HELIOS++ point clouds, and the best classification performance in forested areas is over 92%, which is obtained with the model trained on ROADSENSE point clouds. This work contributes information on a valuable tool for advancing DL applications in transportation scenarios, offering insights and solutions for improved road and roadside management.

Keywords: deep learning; forest roads; synthetic LiDAR; 3D scene simulator



Citation: Comesaña-Cebral, L.; Martínez-Sánchez, J.; Seoane, A.N.; Arias, P. Transport Infrastructure Management Based on LiDAR Synthetic Data: A Deep Learning Approach with a ROADSENSE Simulator. *Infrastructures* **2024**, *9*, 58. <https://doi.org/10.3390/infrastructures9030058>

Academic Editors: Giuseppe Cantisani and Giulia Del Serrone

Received: 29 January 2024

Revised: 29 February 2024

Accepted: 8 March 2024

Published: 13 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Light Detection and Ranging (LiDAR) technology has been successfully applied in many fields in recent decades, including architecture and civil engineering [1]. By identifying objects or regions in a point cloud, it is possible to extract their geometric features and infer information about the real world [2]. Conducting a health analysis as well as determining stability and other physical properties that can be inferred from LiDAR point clouds are the main focus areas in many research works [3]. It is important to highlight the applicability of LiDAR, especially for infrastructure inventory and management, based on semantic segmentation [4–6], where it is crucial to overcome the current limitations of technology related to the demanding costs and workforce linked to the huge datasets obtained. A three-dimensional model analysis can also be used to detect road safety issues related to sight distance on sharp vertical curves and horizontal curves. These visibility limitations can create accident-prone areas and may require corrective measures, such as reducing the permitted speed [7].

Although there are several works that have performed LiDAR point cloud segmentation in forest and transportation environments through heuristic and traditional Machine

Learning (ML) methods, DL-based methods have been proven as ideal tools for these specific types of works [8]. DL-based methods for segmenting LiDAR point clouds into road assets have evolved into three main types: projection-based, discretization-based, and point-based methods [9]. Projection-based approaches generate images from original point clouds, apply Convolutional Neural Networks (CNNs) to segment pixels within these images, and then transfer the results back to the original point cloud [10]. However, these methods are limited by the perspective of the subtracted image, hindering a comprehensive study of the complete shape of each object in the point cloud.

Discretization-based methods utilize algorithms, such as voxelization, to discretize the original point cloud, creating image-like data from the grid information stored in each voxel. These images are often classified using CNNs and extended to 3D CNNs [11]. Despite their effectiveness, these methods face challenges in terms of memory and computational resources, especially in larger scenarios where the cubic relationship between memory needs and the size of the target point cloud can be impractical [12].

Point-based methods focus on the spatial coordinates of each point in the point cloud. PointNet [13] is a pioneering DL model in this category, employing multiLayer perceptron (MLP) for point feature extraction. Subsequent solutions based on PointNet include point-wise MLP methods, like PointNet++ [14], which applies PointNet hierarchically in the spatial domain to learn features at different scale levels, and RandLA-Net [12], which was designed for large-scale point clouds. Point convolution methods, such as PCCN [15] and KPConv [16], propose convolution operations for 3D point sets. Additionally, RNN-based methods, as exemplified by Fan et al. [17], and graph-based methods, demonstrated by Shi et al. [18], employ Recurrent Neural Networks (RNNs) or graph networks, respectively. Point-wise methods have been applied to segment point clouds from transport infrastructures, with examples in road segmentation and object detection [19–24].

Given the limited number of studies focused on the classification of real LiDAR point clouds exclusively trained on synthetic data, the simulation of augmented datasets plays a pivotal role in the training process of DL models. These studies typically adopt two distinct perspectives: aerial, represented by ALS, and terrestrial, encompassing TLS-MLS LiDAR point clouds.

To the best of our knowledge, state-of-the-art ALS simulators rely on the physical characteristics of the laser beam, the reflectance properties of the target surface, and the geometric shape of the trajectory through the 3D virtual world. In some works, the laser is considered as an infinitesimal beam with zero divergence in order to minimize the complexity of the ray-tracing procedure [25,26], and in other works, full-waveform laser scanners are considered, and the full waveform of the backscattered signal is simulated [27,28]. Kukko et al. [29] developed a more sophisticated LiDAR simulator with different point of view assumptions apart from ALS as TLS or MLS, and it considers more properties about the physics concerning the laser behavior in the forestry domain. Kim et al. [30] developed a similar model but with radiometric simulations and clear 3D object representations.

TLS simulations are mainly focused on studies with different and specific needs. One case is seen in the work presented by Wang et al. [31], where a simple model with no beam divergence is considered in order to obtain leaf area index inversion. The methodology presented by Hodge et al. in [32] includes a more detailed model of beam divergence and a full backscattered waveform simulation for TLS measurement error quantification.

It is also worth mentioning other recent and more complex ALS simulators, such as Limulator 4.0 [33] or HELIOS [34] and its enhanced version, HELIOS++ [35], where both versions were proven useful in both aerial and terrestrial points of view. HELIOS++ offers a versatile balance between computational efficiency and point cloud quality, accommodating a wide range of virtual laser scanning (VLS) scales. Users can easily configure simulations, combining different scales within a scene, such as modeling intricate tree details alongside a forest represented by voxels. The high usability of this simulation software makes it easy to script and automate workflows, linking seamlessly with various external software, unlike most of the previously mentioned simulators.

As seen in further sections, all of the synthetic scenarios simulated with our methodology do not have any point of view or trajectory for a LiDAR device. Nevertheless, the digital model is generated completely from scratch without considering any of the physical properties present in a real case of LiDAR measurement.

The main objective of this research work consists of designing and developing a synthetic 3D scenario and point cloud data generator that provides cost-effective labeled datasets of roadside environments to be used as an input for DL semantic classifiers. The simulator is assessed by using the obtained datasets to train the well-known CNN Pointnet++ and comparing the classification results with those obtained with the established HELIOS++ simulator. The main contributions to the state of the art consist of helping road managers reduce time-consuming and demanding work to achieve labeled 3D point cloud datasets to train DL classifiers to support the elaboration of an accurate road inventory for improved safety and asset management.

The structure of this paper is as follows: Section 2 provides insights into the architecture, design considerations, and implementation details of ROADSENSE that bear significance for scientific applications; Section 3 illustrates an application example of ROADSENSE; Section 4 showcases and discusses the results; and, lastly, Section 5 concludes this work.

2. Implementation of ROADSENSE

ROADSENSE (Road and Scenic Environment Simulation) is a novel 3D scene simulator that generates synthetic scenarios and data entirely from scratch to serve as input for DL-based semantic classifiers in road environments, including roadside forest areas. Including 3D point clouds of roads in traditional surveying methods helps road managers directly characterize infrastructure from geometric features, permitting them to perform safety assessments, plan road interventions, and simulate the effects of retrofitting activities.

In this section, the principles and interfaces of ROADSENSE are introduced, emphasizing the implementation details that hold significance for scientific utilization and gaining understanding of the produced synthetic data. The key components of ROADSENSE's performance can be categorized into two sub-sections: the geometric design (Section 2.1) and the comprehensive architecture and modules (Section 2.2).

2.1. Geometric Design

ROADSENSE is designed to primarily produce realistic point clouds. This serves to facilitate the training of DL models for semantic segmentation tasks across varied scenarios, encompassing forested areas and road networks.

Furthermore, ROADSENSE is an improved version of the work presented in [36], which consists of three steps: (i) the generation of a digital terrain model (DTM), (ii) the simulation of roadways and their assets according to official road design norms, and, finally, (iii) the positioning of trees along suitable regions of the DTM. Regardless of the type of point cloud, the initial stages involving the creation of the DTM and tree generation remain consistent. Once these initial steps are completed, the sole differing aspect between the two scenarios is the road and its main assets.

Innovatively, our simulator enhances realism by incorporating road cross section definitions, addressing a limitation observed in a prior work [36]. The previous model placed all road stretch points at the same height, hindering DL model learning due to difficulties in distinguishing equivalent areas. To overcome this, we introduced a vertical pumping feature in the height profile of road points (Figure 1), ensuring improved accuracy and realism in simulated scenarios.

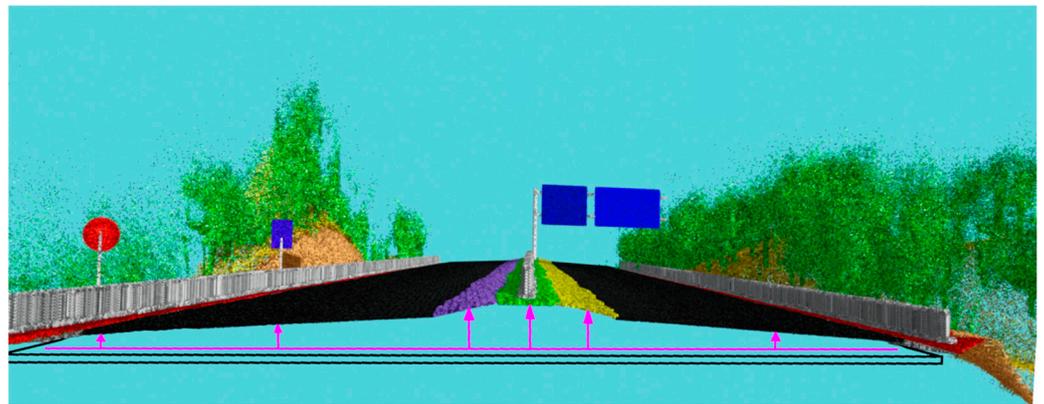


Figure 1. A cross section example. The magenta arrows show the vertical pumping for each point on the road.

Synthetic point clouds may differ from real ones in certain shapes like scanner occlusions. To describe this issue, Figure 2 shows a sample of a real MLS point cloud, where the green line represents the scanner trajectory, and the red rectangles with the same size provide information about the point density in different parts of the point cloud referred to the MLS trajectory. Neural networks are expected to generalize and overcome these errors unless overtrained on specific features; thus, we try to include these features by introducing selective downsampling, a random approach used to simulate occlusions to enhance analytical results.



Figure 2. An MLS point cloud sample of an actual road being surveyed. Rectangles 1, 2, and 3 contain, respectively, 25,000, a few tens, and approximately 500 points.

2.2. Architecture and Modules

All changes within the 3D scenarios generated by ROADSENSE are monitored by the configuration file, from now on referred to as config file, which is the central element and is located inside the config folder. The config file contains all of the parameters required to simulate the point clouds, considering all different scenarios and settings, and it is written in a .txt format, which can be easily modified using a text editor. All parameters defined in the config file are summarized as follows:

- **seed** (index 0): This is a random seed used in intermediate randomizers. Options: "None" or "integer number". If the None option is selected, the simulator will consider the `time.time()` value. Specifying an int number will make all clouds and their features equal.
- **road** (index 1): This indicates whether the resulting point clouds will have a road or not. Options: "True" or "False".
- **spectral_mode** (index 2): This parameter is used in the case of performing a spectral simulation on the resulting point clouds. Options: "True" or "False". If True is set, the

simulator will take longer to generate each point cloud since there are several intermediate calculations, like the normals estimation or the virtual trajectory simulation.

- **road_type** (index 3): This indicates the type of road to simulate. Options: “highway”, “national”, “local”, or “mixed”. If the “mixed” option is selected, the resulting point cloud dataset will consist of a mix of different types of roads.
- **tree_path** (index 4): This represents the absolute path to the directory where all of the tree segments are stored.
- **number_of_clouds** (index 5): This represents the total number of point clouds that will be simulated. It must be an integer number greater than zero.
- **scale** (index 6): This is a geometric scale of the resulting point cloud. Bigger scales will result in bigger sized point clouds but with less point densities. It must be an integer number greater than 0.
- **number_of_trees** (index 7): This represents the total number of trees per cloud. It must be an integer number greater than 0.
- **number_of_transformations** (index 8): This is the number of Euclidean transformations that will suffer each tree segment. It must be an int number. If the total number of tree segments specified in the “tree_path” is less than the “number_of_trees”, then the simulator will randomly repeat all segments until completion.
- **X_buffer** (indexes 9–11): This represents the width of the “X” element in meters. “X” can be “road”, “shoulder”, or “berm”. It must be a float number greater than 0.
- **slope_buffer** (index 12): This represents the width of the slope in meters. Options: “float number greater than 0” or “random”. If “random” is set, then there will be slopes with different widths in the final point clouds.
- **noise_X** (indexes 13–18): This represents the noise threshold in each XYZ direction per point of the “X” element. “X” can be “DTM”, “road”, “shoulder”, “slope”, “berm”, or “refugee_island”. It must be in the “(x,y,z)” bracket notation, where x, y, and z must be float numbers.
- **number_of_trees_refugee_island** (index 19): This represents the number of trees in the median strip. It must be an integer number.
- **number_points_DTM** (index 20): This represents the number of points in the edge of the DTM grid. It should be ~10 times the scale. It must be an integer number greater than 0.
- **vertical_pumping** (index 21): This indicates whether the simulated road will have vertical pumping or not, i.e., if the points closer to the axis road will have different heights than the ones that are further. Options: “True” or “False”.

Also, the different functionalities that ROADSENSE requires to perform each intermediate operation, such as the generation of every road element or the vertical pumping of the road, are defined under the utils folder. The main modules and functions can be summarized as follows:

- **DTM_road_wizard.DTM_road_generator**: This function generates all road-related geometries from scratch.
- **cross_section.X_vertical_pumping**: This function generates some sort of ground elevation in the road-related parts of the “X” road type, i.e., the points closer to the axis road will have a different height than the ones that are further. “X” can be highway, national, or mixed roads.
- **reading_trees.read_segments**: This function reads and stores external tree segments.
- **road_generator.Road_Generator**: This function generates all ground components of the road (except traffic signals and barriers).
- **signal_generator.create_X_signal**: This function generates signals regarding its “X” type. “X” can be “elevated” for elevated big traffic signals in highway and mixed roads or “triangular”, “circle”, or “square” for shaped vertical signals.
- **tree_wizard.tree_generator**: This function performs data augmentation with the previously read tree data and generates new tree segments.

- **trajectory_simulation.compute_spectral_intensity:** This function draws a trajectory over the generated 3D scene and performs a spectral simulation.

3. Case Study

The full workflow of this procedure is illustrated in Figure 3. As a concise overview of the subsequent subsections, the methodology is structured in the following manner: First, LiDAR point clouds of real scenarios are acquired with different scanners in the study area and segmented in order to obtain a reference validation dataset to evaluate the performance of the proposed method. Then, a bunch of datasets comprising 3D scenes are generated using the simulator software under different conditions. Finally, a state-of-the-art DL model is trained with the previously obtained synthetic data and used to perform semantic classifications of the real point clouds acquired in the study area. This procedure is also followed using HELIOS++, which simulates a different input dataset for the DL model training stage in order to compare the results obtained in our work with a well-established simulator.

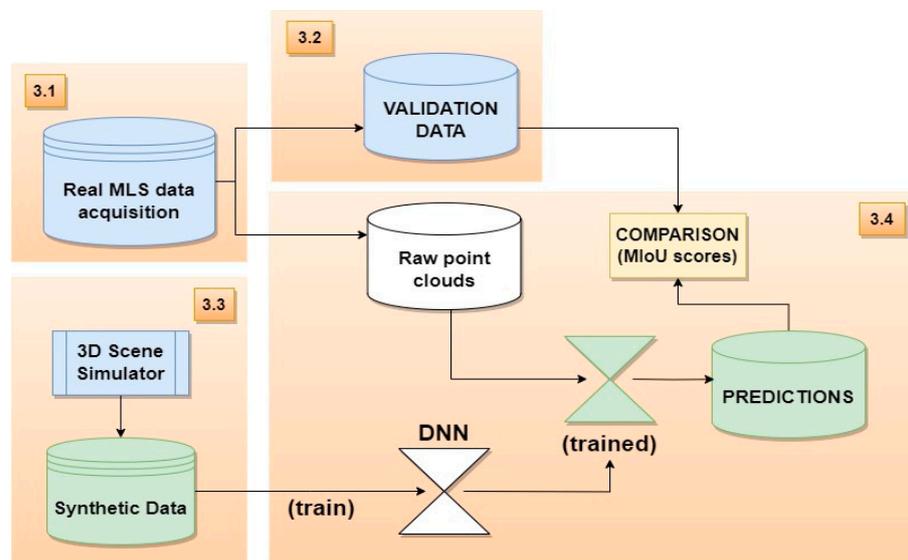


Figure 3. The workflow of the method.

3.1. Real MLS Point Cloud Acquisition

To assess the suitability of 3D scene simulation for specific applications, two distinct scenarios were chosen: dense forests, comprising a mixture of various tree species, shrubs, and undergrowth, which were scanned using Terrestrial Laser Scanning (TLS) devices, and highway roads, which were surveyed with a Mobile Laser Scanning (MLS) system.

The TLS forest data collection sites are situated within the O Xurés region of Galicia, Spain, as depicted in Figure 4.

The TLS data acquisition process employed two distinct sensors, namely Riegl VUX-1UAV and Riegl miniVUX-1DL, referred to hereafter as VUX and miniVUX, respectively. Detailed technical specifications for each sensor are provided in Table 1.

Conversely, the road case study involved a labeled dataset comprising point clouds and images that were collected along a 5 km stretch of highway in Santarem, Portugal. These MLS road data were collected using an Optech Lynx Mobile Mapper M1 system equipped with two LiDAR sensors capable of rotating at a rapid rate of up to 200 Hz, enabling a remarkable number of 200 scanning cycles per second [39]. Each LiDAR sensor within this system boasts a laser measurement rate of up to 500 kHz, achieving a range measuring precision of 8 mm. The navigation system utilized for this MLS data collection process was provided by Applanix [40], featuring an Inertial Navigation System (INS), two Global Navigation Satellite System (GNSS) antennas, and an odometer.

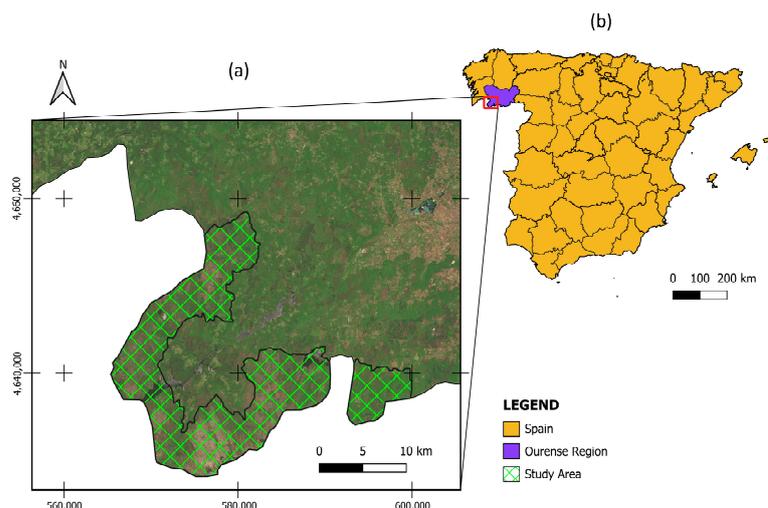


Figure 4. Location of study area at different scales: (a) region of O Xurés (Galicia, Spain), (b) location of study area in Spain.

Table 1. Technical characteristics of laser scanning systems [37,38].

	miniVUX-1DL	VUX-1UAV
Field of view	46° ± 23° off-nadir	330°
Scanning pattern	Circular	Linear
Pulse repetition frequency	100 kHz	550 kHz
Wavelength	905 nm (NIR)	1550 nm
Beam divergence	1.6 × 0.5 mrad	0.5 mrad
Footprint size at 100 m	160 × 50 mm	50 mm
Accuracy	15 mm at 50 m	10 mm at 150 m
Precision	10 mm at 50 m	5 mm at 150 m

For the purposes of this study, the scan frequency was set to its maximum of 200 Hz, and the laser measurement rate was maintained at 250 kHz for both sensors. A single revolution of one sensor generated a scan comprising 1250 points, resulting in an angular resolution of 0.288 degrees. In practical terms, this equates to a point separation of approximately 13 mm within the same scanning line at a range of 2.5 m.

3.2. Validation Data of Forest Scenarios

The accuracy of ground truth, particularly in the context of LiDAR data within forest environments, is often uncertain and is typically assumed rather than precisely known [41]. Specifically, the TLS forest data of this study consist of complex plots with diverse vegetation elements at various heights, which are challenging even for human classifiers. Figure 5 illustrates this complexity, where the close proximity of multiple trees and shrubs makes it difficult to discern individual points.

To address this challenge, we employ the algorithm introduced in [42], which identifies clusters representing individual and multiple trees. While existing methods in the state of the art address the semantic segmentation of individual trees within TLS forest data, the selected algorithm allows for the collection of vegetation segments irrespective of their type, whether represented by trees or shrubs. Also, for the task of retrieving a validation dataset in this section, only semantic information regarding the “vegetation” class is required instead of instance information of each individual element, so all resulting segments are merged in one single collection, reducing classification errors while enhancing the reliability of the validation data. However, this approach is only applicable to forested areas, so road point clouds, which are easier to segment manually, must follow a different process.

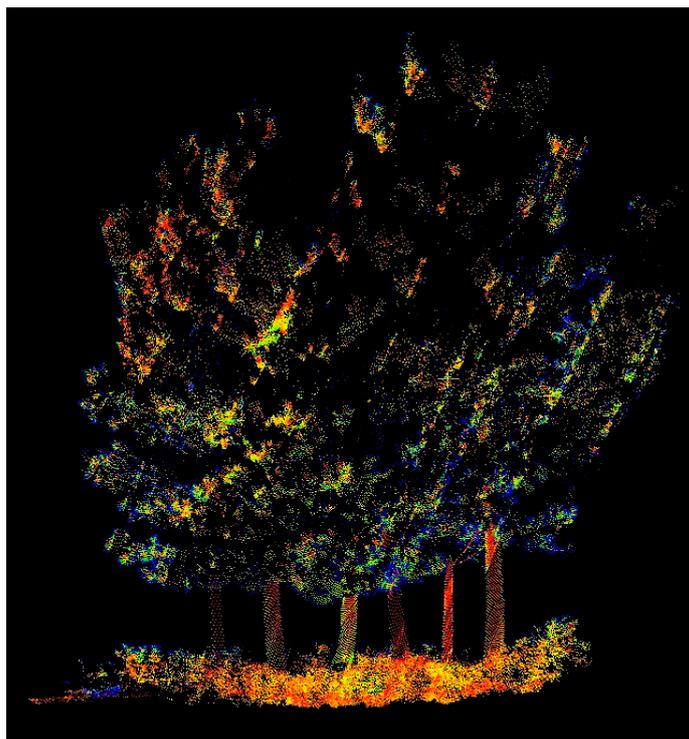


Figure 5. Side view of different trees colored by spectral reflectance at $\lambda = 905$ nm. Proximity of tree crowns and undergrowth makes it challenging to distinguish points belonging different elements.

Now that the process of obtaining a single validation point cloud has been explained, the relevance of DL-based classifiers is notable in these tasks, as they have been proven to be better in terms of time cost and overall result goodness than other manual or ML-based methods.

3.3. Generation of Synthetic Datasets

As mentioned previously, DL models need large and labeled datasets for their training stages, which is a requirement that can be difficult to fulfill depending on the case. To overcome this situation, simulating artificial data can be useful if there is enough similarity between synthetic and real data.

Some examples of point clouds generated with the version 1.0 of the ROADSENSE software are freely available in sharepoint [43].

Since the point clouds used for this study can be grouped as highways and national roads and forestry infrastructures, the generation of synthetic data is carried out within these environments. The differences between the types of roads considered rely on the presence or absence of certain elements, such as the number of platforms, lanes, or median strips. A comparative view of these types of roads can be found in Figure 6.

On the other hand, the well-established virtual laser scanning (VLS) simulator HELIOS++ is used to generate an alternative dataset to train the PointNet++ network. This software takes a 3D scene as an input and performs a full-wave LiDAR simulation, which outputs a 3D point cloud that is hypothetically seen by the specified scanner.

To use this software, it is required to define a survey XML file which contains general VLS information, and it is also necessary to define other XML files, like platform, scene, and scanner files. The definition of the scene XML file is the most important and difficult among all, since all semantic elements must be manually specified one by one with their coordinates and spatial transformations, if applied. This proves the fact that each discrete element must be previously defined from scratch, as carried out with ROADSENSE, or subtracted from an external data source. One way to carry this out is by using the 0.3.1 version of the OSM2World software [44], which creates 3D models of the world from

OpenStreetMap (OSM) [45] data. However, these data lack terrain elevation, and the study cases of this work are underrepresented since only few semantic classes can be retrieved from the OSM data. Elements like barriers, traffic signals, tree varieties, and elevations of the terrain should be added separately to the OSM2World output, because they are conditions that hinder the fluency of the proposed methodology.

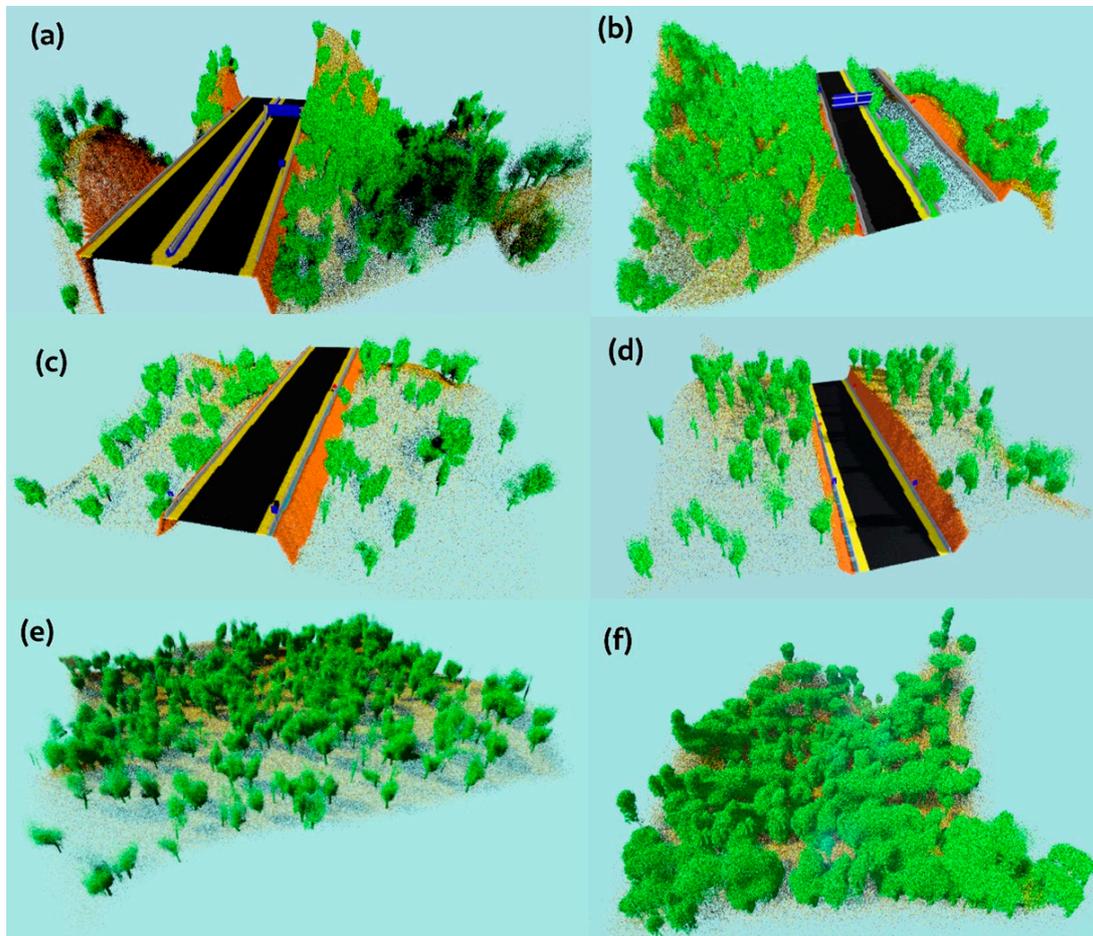


Figure 6. Synthetic samples from each studied environment generated by ROADSENSE: (a,b) highway roads with and without selective downsampling, (c,d) national roads, and (e,f) forests. Colors were applied to help with visualization.

By setting the RIEGL VUX-1UAV as the main scanner device in the survey, whose physical characteristics are reflected in Table 1, onboard of a terrestrial vehicle at different speeds, synthetic forest and road datasets are generated. Some examples are shown in Figure 7, and Table 2 offers a comparative view of the requirements and main characteristics of both simulators.

Table 2. Requirements and main characteristics of HELIOS++ and ROADSENSE simulators.

Features	HELIOS++	ROADSENSE
Trajectory independent	No	Yes
Performs full-wave simulation	Yes	No
Scene creation flexibility	No	Yes
Performs multispectral simulation	Yes	Yes
Not dependent on laser scanner parameter setup	No	Yes
Provides semantic information	Yes	Yes
Not dependent on prior labeling procedures	No	Yes

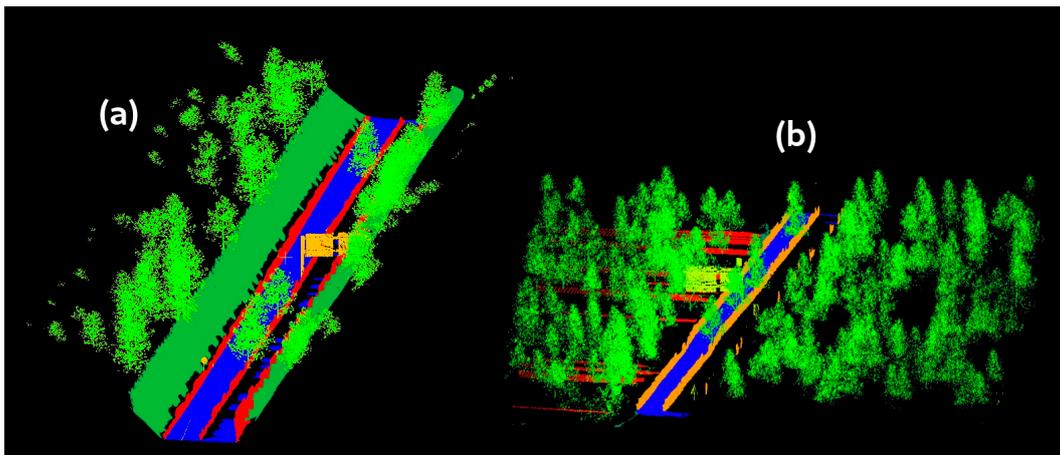


Figure 7. Synthetic samples from each studied environment generated by HELIOS++: (a) highway roads and (b) national roads surrounded by forested areas. Colors refer to each semantic class.

3.4. Semantic Segmentation with DNN Model

As shown in the related work, there are several architectures that perform semantic classification on point clouds, but only a few of them work directly with the point cloud itself and not with discretized representations or images. Among all point-wise methods, PointNet is considered as the pioneer, and its enhanced version, PointNet++, was tested in many cases, providing state-of-the-art results for object detection and semantic classification [46]. Accordingly, PointNet++ is the model selected to perform semantic classification in real scenarios from synthetic data learning.

The original PointNet applies a function, f , that creates a vector of an unordered set of points:

$$f(x_1, x_2, \dots, x_n) = \gamma(\text{Max}_{i=1}^n \{h(x)\}), \quad (1)$$

where γ and h represent MLP (multilayer perceptron). The architecture of PointNet++ includes several key layers related to sampling, feature extraction, grouping, and segmentation, which are based on PointNet. The sampling layer is focused on efficiency; thus, it selects a subset of points from the input. This small subset of points is used to extract features related to the fine detail of the objects. The following layer is a grouping that constructs local region sets using the Farthest Point Sampling algorithm to obtain higher-level features. The final layer based on PointNet performs the classification based on feature aggregation. This DNN was first used for classifying the Scannet dataset [47], and, because of this, the internal architecture of the model was designed to fit that purpose. This dataset is mainly made of 3D scanned objects with small point densities, and this is the reason why the authors of PointNet++ considered a total of 8192 entries for their model. However, this is not enough in scenes where there are high point densities spread over big spaces like the synthetic and real environments studied in this work. In this work, four tests were conducted using modifications of the original PointNet++ architecture to see how sensitive the DNN is to the input data in each scenario. Tables 3 and 4 include a definition of the default and modified architectures of PointNet++.

As the number of road assets considered in this work exceeds the number of tree species, the variability of the synthetic road point clouds is higher than that of the nearby forest point clouds. Thus, a different planification of the training stages must be developed for each scenario. So, the synthetic training datasets that were simulated were made by a total of 400 road point clouds, with 200 of each scenario, while just 50 samples were needed to train the DNN in forest environments. This difference is due to the number of classes that the DNN needs to learn, which consists of 7–8 classes and just 2 classes in road and forest environments, respectively.

Table 3. The key properties of the default architecture of PointNet++. Each layer receives 32 sample points and their associated tensors, which contain the features to be used for the classification of the input cloud. The N_Points, Radius, and N_samples columns show the different sizes of PointNet characteristics for each type (ID) of layer.

	Layer ID	Point Features	N_Points	Radius
Downsampling	0	[7,32,32,64]	1024	0.1
	1	[67,64,64,128]	256	0.2
	2	[131,128,128,256]	64	0.4
	3	[259,256,256,512]	16	0.8
Upsampling	4	[1024,512,512,256]	16	0.1
	5	[256,256,256,256]	64	0.2
	6	[64,128,256,128]	256	0.4
	7	[16,128,128,128]	1024	0.8

Table 4. The modifications of PointNet++ basic architectures. As an extension of Table 2, the first 3 columns represent whether each modified version of PointNet++ incorporates the specified layers. The symbol ✓ indicates the presence of the layer in question in the modified version.

4-Layer Version	5-Layer Version	6-Layer Version	Layer ID	Point Features	N_Points	Radius
✓	✓	✓	0	[7,32,32,64]	65,536	0.1
✓	✓	✓	1	[67,64,64,128]	8192	0.2
✓	✓	✓	2	[131,128,128,256]	2048	0.4
✓	✓	✓	3	[259,256,256,512]	1024	0.5
	✓	✓	4	[512,512,512,1024]	512	0.7
		✓	5	[1024,1024,1024,2048]	256	0.8
✓	✓	✓	6	[256,4,128,128]	256	0.1
✓	✓	✓	7	[1024,64,128,128]	512	0.2
✓	✓	✓	8	[2048,64,128,128]	1024	0.4
✓	✓	✓	9	[4096,128,256,128]	2048	0.5
	✓	✓	10	[8192,256,256,256]	8192	0.7
		✓	11	[65536,512,512,256]	65,536	0.8

Finally, the GPU NVIDIA A1000 and 40 GB of HBM2 of the FinisTerra III (FTIII) were used during this stage, and it took 1 h per architecture to train over 100 epochs from scratch.

4. Results and Discussion

4.1. Segmentation Metrics

In order to evaluate the performance of the neural network for semantic classification during the training validation and testing stages, we obtain the following well-established quantitative parameters that are frequently used for both image and 3D scene segmentation [48]: overall accuracy (OA), mean accuracy (MA), mean loss (ML), intersection over union (IoU) and mean intersection over union (MIoU).

It is worth mentioning that the mean values, i.e., MA and MIoU, take into account their values per class in the dataset and compute their averages, complementing the overall accuracy and the intersection over union. This improves the confidence for the classification of unbalanced data, and the highest MIoU value is used to estimate when the DNN achieved its best performance.

4.2. Architectures Training

Based on [49,50], over 100 epochs, the DNN model is trained with synthetic data in its four different architecture modifications, and to analyze its training performance, the metrics of the previous subsection are computed after each epoch. Also, to test the model, a total of five labeled point clouds of real scenarios are considered in each case.

The four different DNN configurations are tested at four different scenarios, and the metrics are calculated from random samples, as shown in Tables 4–6. Since smooth-

ing is a method that involves finding a sequence of values that exhibits the trend of a given set of data [51], the obtained graphs are smoothed to overcome the variability, as depicted in Figures 8–10. These figures present the results for the training and validation of PointNet++ in different environments using synthetic point clouds generated by ROADSENSE as input data. The figure includes the results using (a) the default architecture of PointNet, (b) four PointNet-like modified layers, (c) five PointNet-like layers, and (d) six PointNet-like layers. The overall accuracy (OA) is depicted in blue, whereas the mean accuracy (MA) is depicted in yellow. Other metrics, such as the mean loss (ML) and mean intersection over union (MIoU), are presented in red and green, respectively. The context of the results is as follows: Figure 8 is focused on a highway environment, Figure 9 is focused on national roads, and Figure 10 is focused on a forest environment.

Table 5. Summary of results in highway environments with ROADSENSE point clouds used as input data. Class labels are as follows: (0) trees, (1) DTM, (2) circulation points, (3) barriers, (4) signals, (5) refuge island, (6) slope, and (7) berm. Bold numbers represent the highest MIoU score.

		Default	4 Layers	5 Layers	6 Layers
OA (%)		98.113	97.768	97.440	98.051
MA (%)		92.463	96.417	93.832	96.925
IoU per class (%)	0	10.074	75.910	76.022	10.074
	1	2.075	96.290	95.014	2.075
	2	98.072	97.932	96.420	98.072
	3	1.002	95.010	95.227	1.002
	4	64.213	80.009	98.092	64.213
	5	30.026	22.069	31.440	30.026
	6	0.000	0.000	0.000	0.000
	7	75.363	72.073	68.962	75.363
MIoU (%)		35.103	67.412	70.147	65.547

Table 6. Summary of results in national road environments with ROADSENSE point clouds used as input data. Class labels are as follows: (0) trees, (1) DTM, (2) circulation points, (3) barriers, (4) signals, (6) slope, and (7) berm. Bold numbers represent the highest MIoU score.

		Default	4 Layers	5 Layers	6 Layers
OA (%)		87.019	88.581	97.533	96.870
MA (%)		80.839	82.336	94.892	96.347
IoU per class (%)	0	21.031	65.915	85.072	82.007
	1	0.000	0.000	0.000	0.000
	2	92.762	91.057	93.996	95.044
	3	71.091	75.002	70.671	69.200
	4	31.059	66.931	82.552	91.404
	6	11.12	62.448	83.683	83.726
	7	23.331	78.798	79.385	77.481
	MIoU (%)		35.771	62.879	70.766

Additionally, the same experiments were conducted with the point clouds generated by HELIOS++, and the computed metrics are reflected in Figures 11–13 and Tables 7–9. The figures show the resulting metrics with the same color code as that in the previous ROADSENSE case, focusing also on highway (Figure 11), national road (Figure 12), and forest environments (Figure 13).

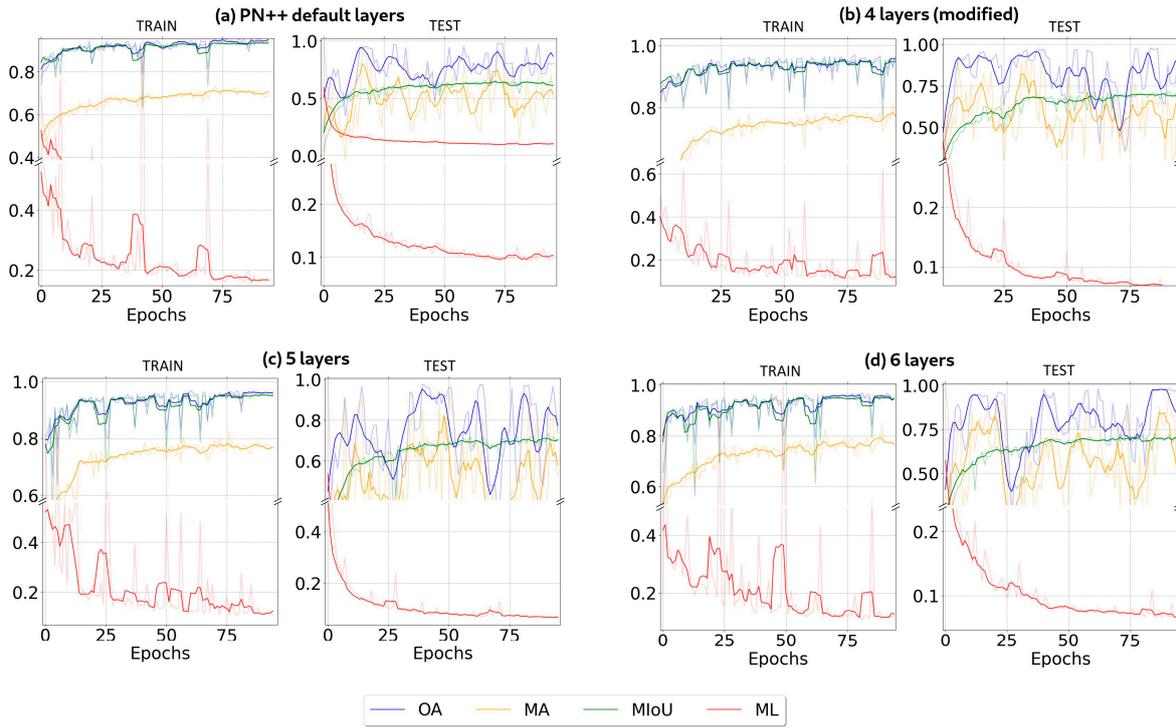


Figure 8. Results for training and validation of PointNet++ in highway environments.

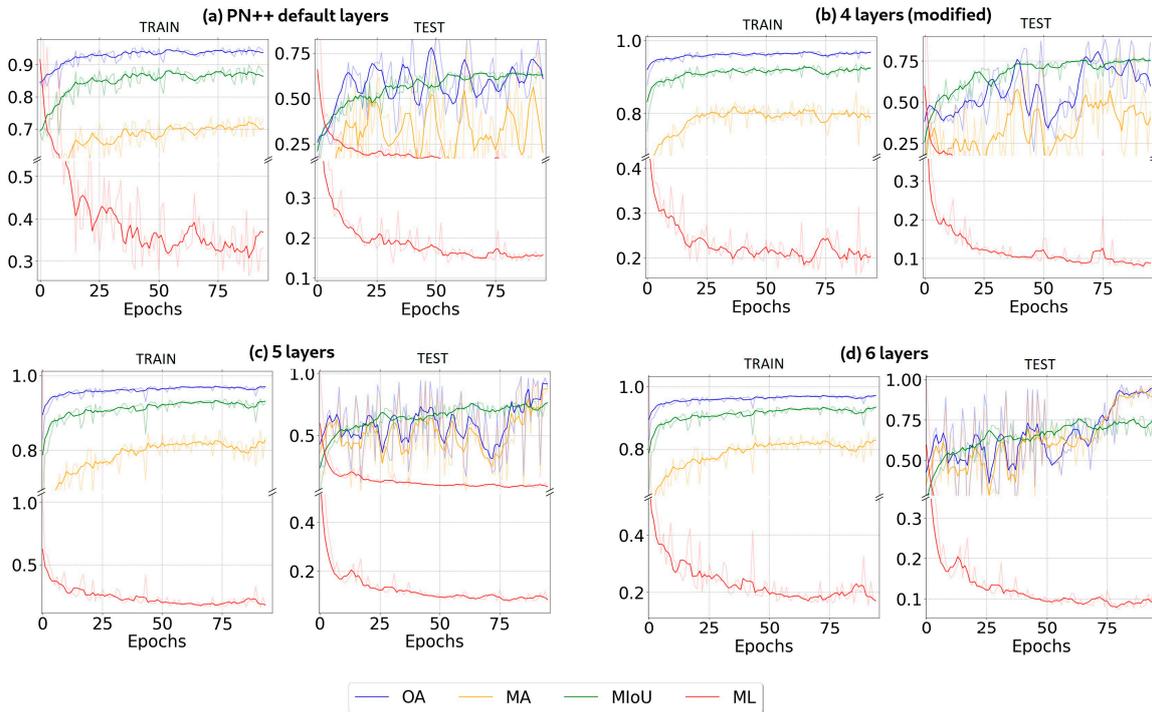


Figure 9. Training and validation of PointNet++ in national road environments with ROADSENSE point clouds used as input data.

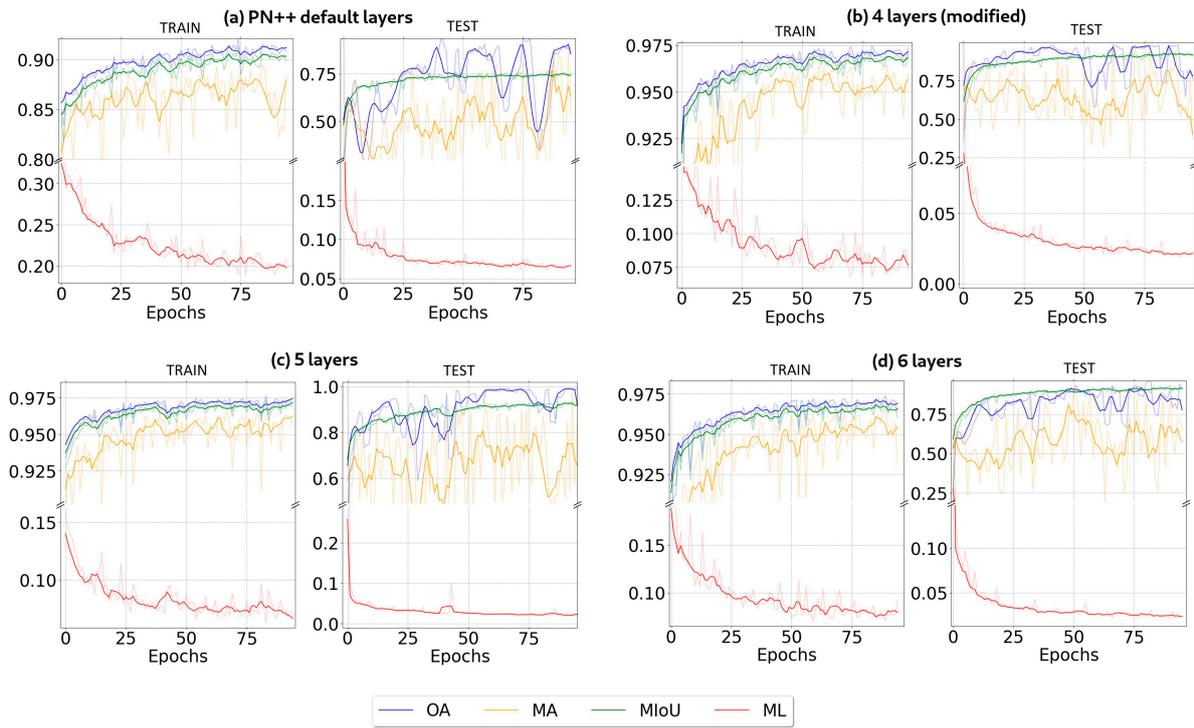


Figure 10. Training and validation of PointNet++ in forest environments with ROADSENSE point clouds used as input data.

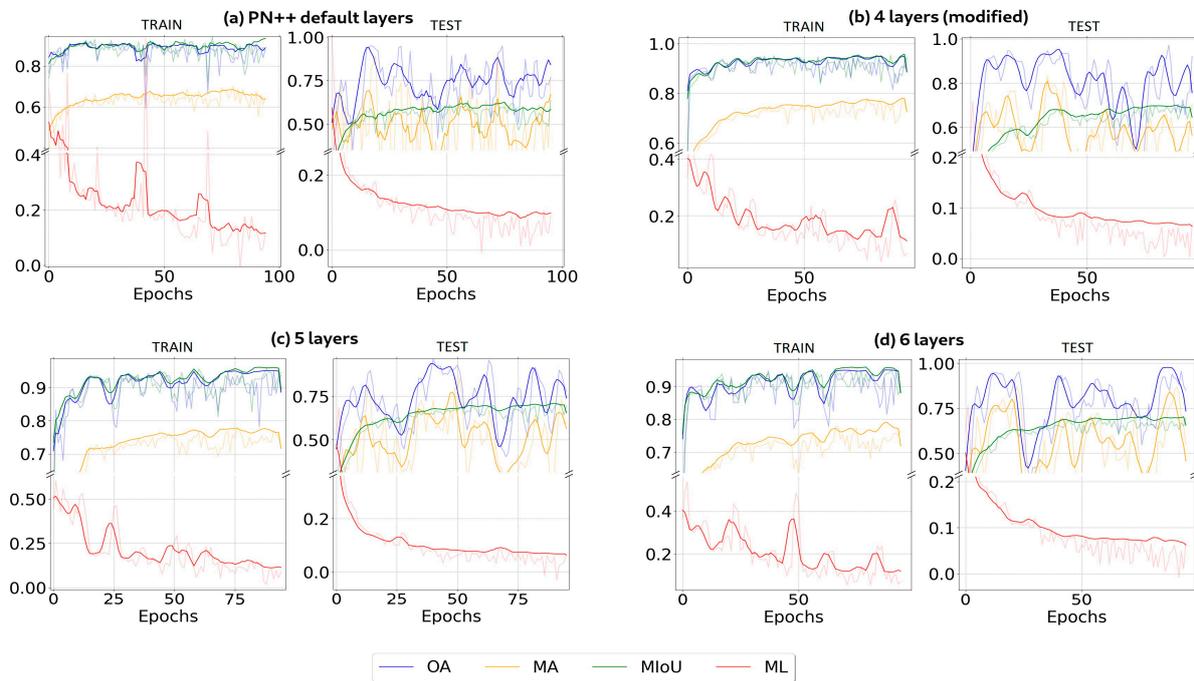


Figure 11. Training and validation of PointNet++ in highway environments with HELIOS++ point clouds used as input data.

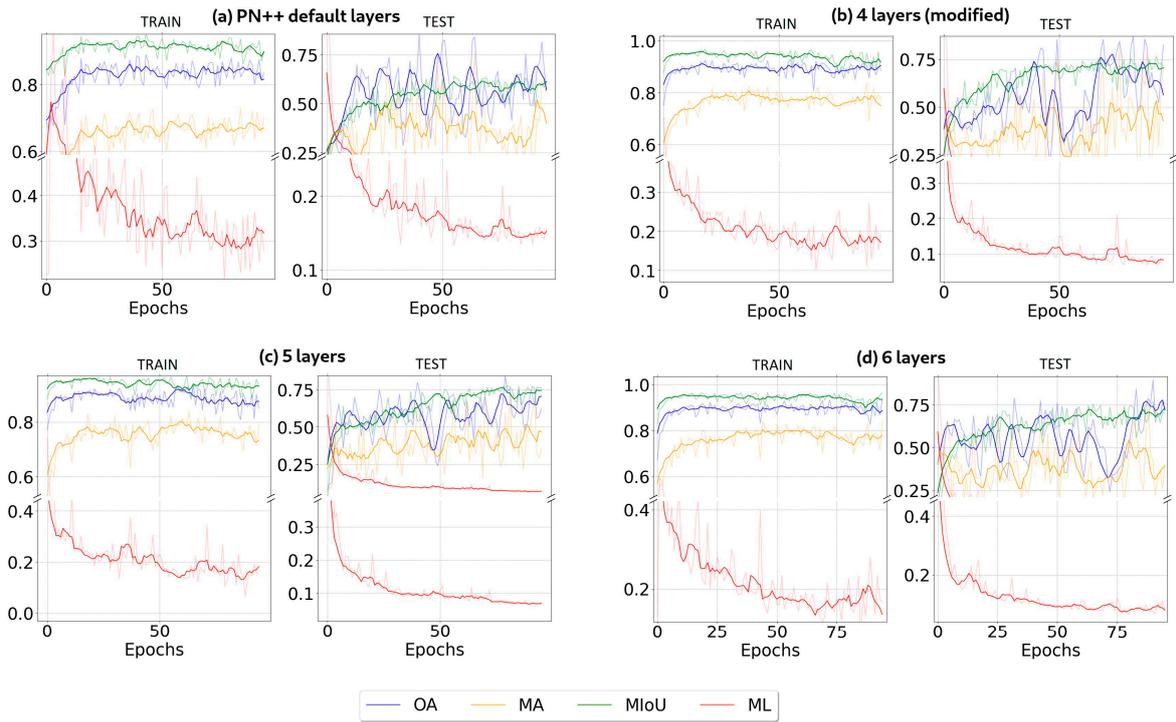


Figure 12. Training and validation of PointNet++ in national road environments with HELIOS++ point clouds used as input data.

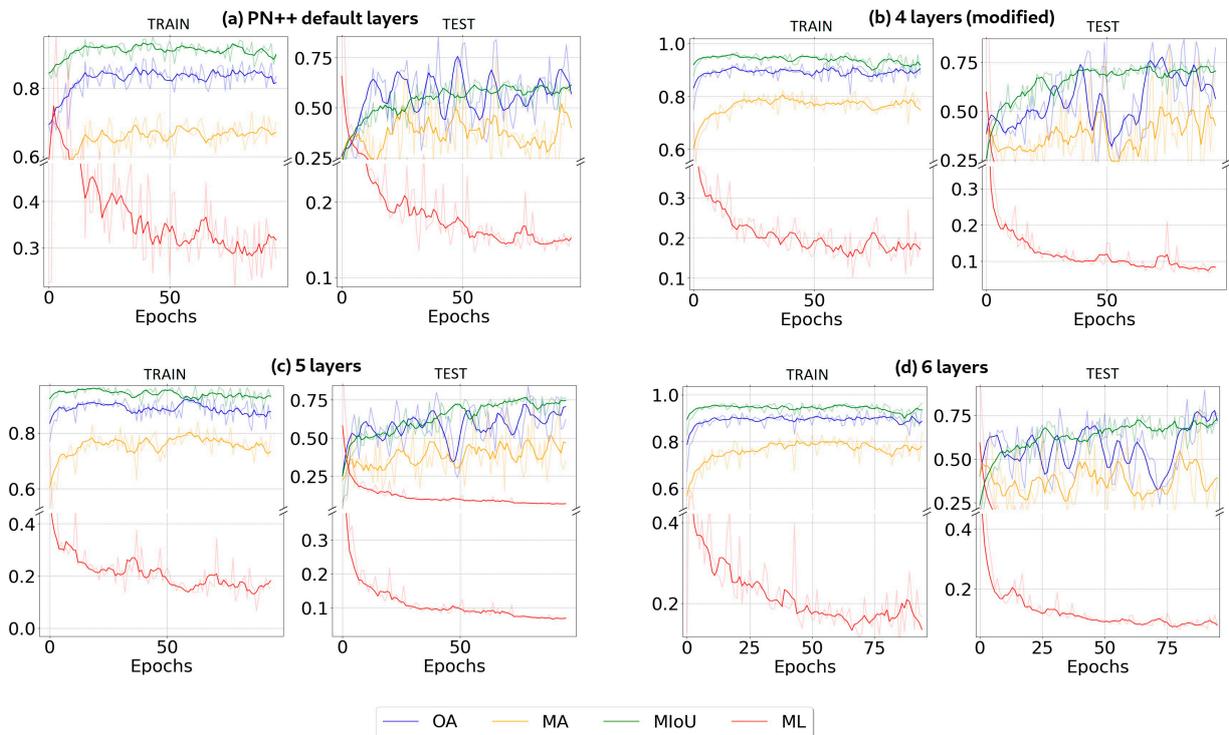


Figure 13. Training and validation of PointNet++ in forest environments with HELIOS++ point clouds used as input data.

Table 7. Summary of results in forest environments with ROADSENSE point clouds used as input data. Class labels are as follows: (0) trees and (1) DTM. Bold numbers represent the highest MIoU score.

		Default	4 Layers	5 Layers	6 Layers
OA (%)		94.385	98.662	99.283	95.979
MA (%)		87.976	96.370	96.594	89.102
IoU per class (%)	0	75.092	95.281	92.001	91.085
	1	80.270	90.310	92.312	89.271
MIoU (%)		77.681	92.796	92.157	90.178

Table 8. Summary of results in highway environments with HELIOS++ point clouds used as input data. Class labels are as follows: (0) trees, (1) DTM, (2) circulation points, (3) barriers, (4) signals, (5) refuge island, (6) slope, and (7) berm. Bold numbers represent the highest MIoU score.

		Default	4 Layers	5 Layers	6 Layers
OA (%)		90.841	86.804	79.798	95.119
MA (%)		61.948	63.124	65.798	82.106
IoU per class (%)	0	30.527	75.842	75.432	76.028
	1	91.280	96.016	94.942	95.796
	2	97.248	97.019	96.022	97.751
	3	55.161	94.043	94.498	93.742
	4	64.142	79.554	98.059	64.119
	5	59.247	21.800	20.796	29.129
	6	0.000	0.000	0.000	0.000
	7	88.014	71.180	68.030	75.067
MIoU (%)		65.183	69.959	70.911	71.289

Table 9. Summary of results in national road environments with HELIOS++ point clouds used as input data. Class labels are as follows: (0) trees, (1) DTM, (2) circulation points, (3) barriers, (4) signals, (6) slope, and (7) berm. Bold numbers represent the highest MIoU score.

		Default	4 Layers	5 Layers	6 Layers
OA (%)		66.383	74.731	73.503	71.715
MA (%)		51.205	53.563	50.655	52.865
IoU per class (%)	0	50.889	65.035	64.734	71.432
	1	0.000	0.000	0.000	0.000
	2	92.742	91.031	93.826	84.510
	3	90.898	74.449	70.571	68.253
	4	50.607	66.465	82.194	85.830
	6	30.522	62.012	75.475	73.352
	7	52.992	78.766	78.880	70.679
	MIoU (%)		57.224	72.321	77.081

4.3. DNN Performance: Inferences on Real LiDAR Point Clouds

The outcomes of the training phases exhibit remarkable similarity, with the primary distinction being the convergence time. This is particularly notable as an increase in the number of network layers enhances the results.

As a common fact, in all three scenarios, the minimum IoU scores were achieved with the default architecture of PointNet++, and this is because the DNN increased its size as it used more layers and more feature vectors to learn.

The difference between the MIoUs achieved by the default architecture and the variants presented in this section is in the range of 15–36% of improvement in the classifications, which is a phenomenon that supports the idea of using alternative versions and configurations of the hidden layers of the DNN model selected, if withstood by the hardware.

Some views of each real scenario classified by the best models are shown in Figures 14–16. Figures 14 and 15 show, respectively, the segmentation results for highways and national roads, with a similar color scheme, as follows: black for circulation points, light grey for barriers, blue for signals, brown for DTM, orange for berms, dark grey for refuge islands, and green for vegetation. For forest environments, the color scheme consists of green and brown colors to represent vegetation and DTM points, respectively.

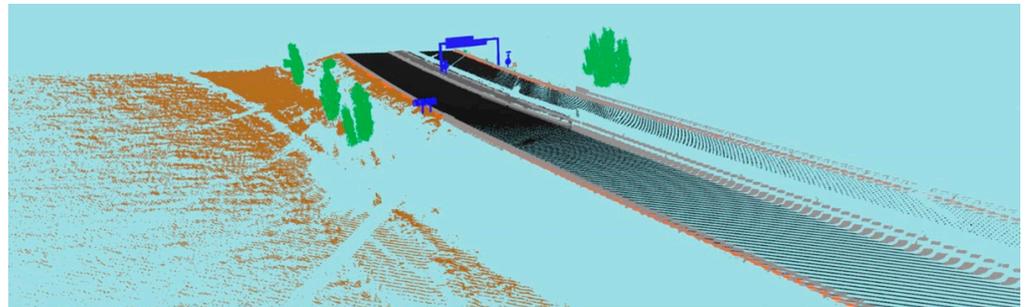


Figure 14. A sample of a real highway road segmented by an amplified version of PointNet++, which was trained with the ROADSENSE output.

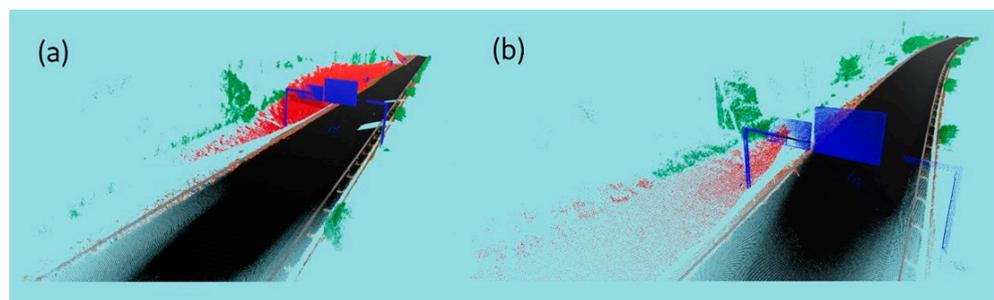


Figure 15. (a,b) Samples of real national roads segmented by an amplified version of PointNet++, which was trained with the ROADSENSE output.

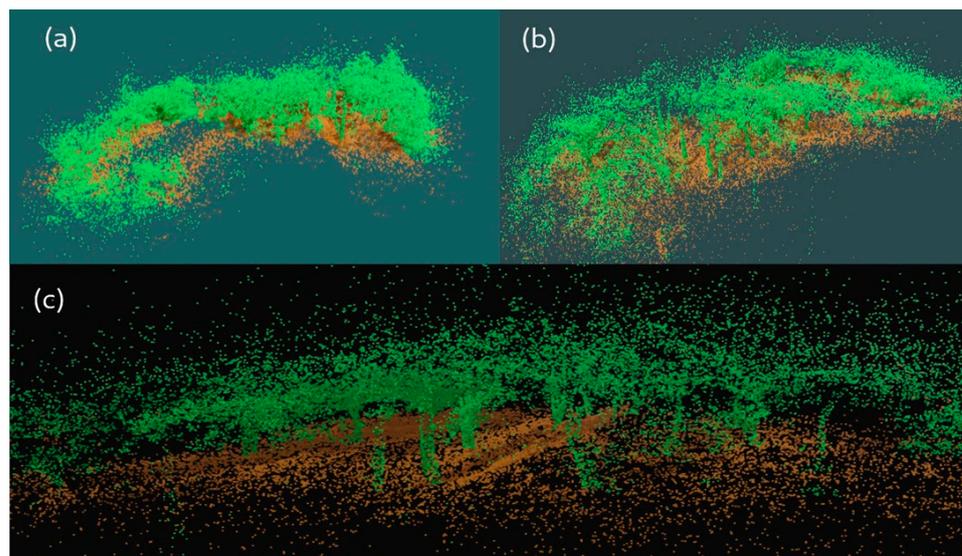


Figure 16. (a–c) Samples of real forest TLS point clouds segmented by an amplified version of PointNet++, which was trained with the ROADSENSE output.

The trained models used in these classifications were those that achieved the highest MIOU values during their training stages.

When comparing the performances of this methodology applied on the ROADSENSE and HELIOS++ point clouds and those of other related works, as shown in Table 10, it can be seen that the MIOU scores achieved by the models proposed in this paper, which are in the range of 71–93%, outperform the results of other related works in the literature where other simulators and real data sources were used.

Table 10. Summary of results in forest environments with HELIOS++ point clouds used as input data. Class labels are as follows: (0) trees and (1) DTM. Bold numbers represent the highest MIOU score.

		Default	4 Layers	5 Layers	6 Layers
	OA (%)	94.956	91.418	95.169	88.608
	MA (%)	68.160	74.094	77.870	65.868
IoU per classes (%)	0	70.210	85.550	86.605	85.145
	1	79.909	90.519	91.824	89.471
	MIOU (%)	74.903	87.999	89.176	87.281

The MIOU metrics from other state-of-the-art works shown in Table 11 vary from ~40% to ~63%, which are significantly lower than the 70–95% range of the results from the experiments carried out using the methodology in this paper. The main difference between this work and most of the previous ones is the use of synthetic data to train a DL model instead of a previous labeling process with real data, which can be highly time-consuming, even if it is carried out manually by the user or by using a traditional unsupervised ML method. It is worth mentioning the cases of references [19,24], where the DL models were trained by mixed datasets composed of both synthetic and real data but in urban scenes. In these works, there were some classes like “cyclist” or “fence” where the DL model achieved very low results (~3% and ~21% for IoU, respectively), showing that the observed MIOU scores were under 45%. Instead, if all classes whose IoU scores were below 25% were ignored, these works would reach MIOU scores close to ~71%, which is on the same range as the results offered by PointNet++, which was previously trained with ROADSENSE.

In contrast, in this paper, PointNet++ was used, which is a point-wise DL model like the SEGCloud adaptation of [24] or the SqueezeSegV2 network presented in [19]. However, it can outperform many other point-wise classifiers in terms of accuracy due to improved training strategies and increased model sizes [52], such as the four, five-, and six-layer adaptations of PointNet++ used in this work. Since the classification metrics on real environments made by these PointNet++ variants are good enough, it can be assumed that it is preferable to avoid performing on-site measurements of several point clouds in real scenarios that can be digitally simulated, either using full geometry generators or VLS frameworks, like ROADSENSE and HELIOS++, respectively.

These results show that ROADSENSE, along with HELIOS++, can help improve road management. An application use case of the simulator is related to road safety. Previous works have shown that the use of unmanned aerial vehicles (UAVs) for data acquisition on road sections that could present road safety problems is straightforward [7]. Even though UAVs make it easier for highway managers to collect data soon after identifying safety issues, national- and European-level regulations restrict flight operations over roads. ROADSENSE can generate 3D scenarios to test analytic procedures that are suitable for assessing sight distance on a road section. The resulting methods could be applied to actual 3D models of a road derived from the data captured by the UAV platform.

Table 11. A segmented comparison between the performance of the proposed methodology applied on ROADSENSE point clouds and that of other state-of-the-art works. The technology and source columns refer to the points of view of the collected or simulated training data.

Work	Technology	Source	Task	Model	Max. MIoU (%)
[23]	MLS	Real	Urban and transport scene segmentation	SEGCloud	61.30
[20]	MLS	Real	Road marking segmentation	PointNet++	63.35
[22]	MLS	Real	Urban and transport scene segmentation	ARandLA-Net	54.40
[21]	ALS	Real	Urban scene segmentation	PointNet++	44.28
[19]	MLS	Synthetic	Urban scene segmentation	SqueezeSegV2	44.90
[24]	MLS	Synthetic	Urban scene segmentation	SEGCloud based	39.97
(Ours)	MLS	HELIOS++	Transport scene segmentation	PointNet++ based	77.08
			Forest scene segmentation		89.18
	Synthetic	ROADSENSE	Transport scene segmentation		71.27
			Forest scene segmentation		92.80

5. Conclusions

This study introduces a new open-source 3D scene simulation framework, ROADSENSE. This simulator is designed to address the scarcity of public 3D datasets in forest and roadway environments by generating point clouds with semantic information. Its specific focus lies in facilitating the training of DL-based models that necessitate extensive datasets.

To assess the reliability of the simulator, experiments were conducted. The state-of-the-art DL model, PointNet++, was trained with synthetic data generated by ROADSENSE and demonstrated its effectiveness through inferences on real point clouds, yielding favorable results. Comparable experiments were also conducted using point clouds generated by the state-of-the-art LiDAR simulator HELIOS++, which, unlike ROADSENSE, requires the specification of non-trivial inputs such as a predefined 3D scene and physical characteristics of the simulated LiDAR device alongside a trajectory. Although configuring training datasets with HELIOS++ demands a notably longer time compared to ROADSENSE, the observed similarity in the results encourage digital simulations of scenarios with ROADSENSE to be prioritized over on-site measurements of multiple point clouds in real environments. This paper addressed how ROADSENSE can benefit road safety assessments by generating data that are suitable for road managers to evaluate the visibility distance on a road section or the vegetation condition nearby roads. These generated data allow for the labeling of assets at a negligible cost compared to other sources and, most importantly, will permit safety-related methodologies to be applied to actual datasets. As a result, ROADSENSE is a valuable tool for monitoring green and grey infrastructures within transportation systems and their surrounding forest environments, whose interaction is critical for anticipating potential risks concerning roadway health.

Author Contributions: Conceptualization, L.C.-C., J.M.-S. and P.A.; methodology, L.C.-C., J.M.-S. and P.A.; software, L.C.-C.; validation, L.C.-C., J.M.-S., A.N.S. and P.A.; formal analysis, L.C.-C., J.M.-S. and A.N.S.; investigation, L.C.-C. and J.M.-S.; resources, L.C.-C. and A.N.S.; data curation, L.C.-C. and A.N.S.; writing—original draft preparation, L.C.-C. and J.M.-S.; writing—review and editing, L.C.-C., J.M.-S., A.N.S. and P.A.; visualization, L.C.-C. and A.N.S.; supervision, J.M.-S. and P.A.; project administration, J.M.-S. and P.A.; funding acquisition, L.C.-C., J.M.-S., A.N.S. and P.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Spanish government through the PREP2022-000304 grant and the 4Map4Health project, selected in the ERA-Net CHIST-ERA IV call (2019) and founded by the State Research Agency of Spain (reference PCI2020-120705-2/AEI/10.13039/501100011033) and InfraROB (Maintaining integrity, performance and safety of the road infrastructure through autonomous robotized solutions and modularization), which has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement no. 955337. This document only reflects the authors’ views, and the agency is not responsible for any use that may be made of the information it contains. The statements made herein are solely the responsibility of the authors.

Data Availability Statement: The code is available at <https://github.com/GeoTechUVigo/ROADSENSE>. Some examples of point clouds generated with the ROADSENSE software are freely available at the following link: [geotech_uvigo_sharepoint_link](#).

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

References

1. Zimmer, W.; Ercelik, E.; Zhou, X.; Ortiz, X.J.D.; Knoll, A. A Survey of Robust 3D Object Detection Methods in Point Clouds. *arXiv* **2022**, arXiv:2204.00106. [[CrossRef](#)]
2. Velizhev, A.; Shapovalov, R.; Schindler, K. Implicit Shape Models for Object Detection in 3D Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *1–3*, 179–184. [[CrossRef](#)]
3. Kaartinen, E.; Dunphy, K.; Sadhu, A. LiDAR-Based Structural Health Monitoring: Applications in Civil Infrastructure Systems. *Sensors* **2022**, *22*, 4610. [[CrossRef](#)] [[PubMed](#)]
4. Buján, S.; Guerra-Hernández, J.; González-Ferreiro, E.; Miranda, D. Forest Road Detection Using LiDAR Data and Hybrid Classification. *Remote Sens.* **2021**, *13*, 393. [[CrossRef](#)]
5. Ma, H.; Ma, H.; Zhang, L.; Liu, K.; Luo, W. Extracting Urban Road Footprints from Airborne LiDAR Point Clouds with PointNet++ and Two-Step Post-Processing. *Remote Sens.* **2022**, *14*, 789. [[CrossRef](#)]
6. Xu, D.; Wang, H.; Xu, W.; Luan, Z.; Xu, X. LiDAR Applications to Estimate Forest Biomass at Individual Tree Scale: Opportunities, Challenges and Future Perspectives. *Forests* **2021**, *12*, 550. [[CrossRef](#)]
7. Iglesias, L.; De Santos-Berbel, C.; Pascual, V.; Castro, M. Using Small Unmanned Aerial Vehicle in 3D Modeling of Highways with Tree-Covered Roadsides to Estimate Sight Distance. *Remote Sens.* **2019**, *11*, 2625. [[CrossRef](#)]
8. Chen, J.; Su, Q.; Niu, Y.; Zhang, Z.; Liu, J. A Handheld LiDAR-Based Semantic Automatic Segmentation Method for Complex Railroad Line Model Reconstruction. *Remote Sens.* **2023**, *15*, 4504. [[CrossRef](#)]
9. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [[CrossRef](#)]
10. Lawin, F.J.; Danelljan, M.; Tosteberg, P.; Bhat, G.; Khan, F.S.; Felsberg, M. *Deep Projective 3D Semantic Segmentation*; Springer: Cham, Switzerland, 2017; pp. 95–107.
11. Lu, H.; Wang, H.; Zhang, Q.; Yoon, S.W.; Won, D. A 3D Convolutional Neural Network for Volumetric Image Semantic Segmentation. *Procedia Manuf.* **2019**, *39*, 422–428. [[CrossRef](#)]
12. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
13. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
14. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv* **2017**, arXiv:1706.02413.
15. Zhang, Y.; Chen, X.; Guo, D.; Song, M.; Teng, Y.; Wang, X. PCCN: Parallel Cross Convolutional Neural Network for Abnormal Network Traffic Flows Detection in Multi-Class Imbalanced Network Traffic Flows. *IEEE Access* **2019**, *7*, 119904–119916. [[CrossRef](#)]
16. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. *arXiv* **2019**, arXiv:1904.08889.
17. Fan, H.; Yang, Y. PointRNN: Point Recurrent Neural Network for Moving Point Cloud Processing. *arXiv* **2019**, arXiv:1910.08287.
18. Shi, W.; Raganathan, R. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
19. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 4376–4382.
20. Yao, L.; Qin, C.; Chen, Q.; Wu, H. Automatic Road Marking Extraction and Vectorization from Vehicle-Borne Laser Scanning Data. *Remote Sens.* **2021**, *13*, 2612. [[CrossRef](#)]
21. Jing, Z.; Guan, H.; Zhao, P.; Li, D.; Yu, Y.; Zang, Y.; Wang, H.; Li, J. Multispectral LiDAR Point Cloud Classification Using SE-PointNet++. *Remote Sens.* **2021**, *13*, 2516. [[CrossRef](#)]
22. Zou, Y.; Weinacker, H.; Koch, B. Towards Urban Scene Semantic Segmentation with Deep Learning from LiDAR Point Clouds: A Case Study in Baden-Württemberg, Germany. *Remote Sens.* **2021**, *13*, 3220. [[CrossRef](#)]
23. Tchapmi, L.P.; Choy, C.B.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017.
24. Wang, F.; Zhuang, Y.; Gu, H.; Hu, H. Automatic Generation of Synthetic LiDAR Point Clouds for 3-D Data Analysis. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 2671–2673. [[CrossRef](#)]
25. Lohani, B.; Mishra, R.K. Generating LiDAR Data in Laboratory: LiDAR Simulator. In Proceedings of the ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Espoo, Finland, 12–14 September 2007; pp. 264–269.

26. Lovell, J.L.; Jupp, D.L.B.; Newnham, G.J.; Coops, N.C.; Culvenor, D.S. Simulation Study for Finding Optimal Lidar Acquisition Parameters for Forest Height Retrieval. *For. Ecol. Manag.* **2005**, *214*, 398–412. [CrossRef]
27. Sun, G.; Ranson, K.J. Modeling Lidar Returns from Forest Canopies. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 2617–2626. [CrossRef]
28. Morsdorf, F.; Frey, O.; Koetz, B.; Meier, E. Ray Tracing for Modeling of Small Footprint Airborne Laser Scanning Returns. In Proceedings of the ISPRS Workshop ‘Laser Scanning 2007 and SilviLaser 2007’, Espoo, Finland, 12–14 September 2007; ISPRS: Espoo, Finland, 2007; pp. 294–299.
29. Kukko, A.; Hyyppä, J. Small-Footprint Laser Scanning Simulator for System Validation, Error Assessment, and Algorithm Development. *Photogramm. Eng. Remote Sens.* **2009**, *75*, 1177–1189. [CrossRef]
30. Kim, S.; Min, S.; Kim, G.; Lee, I.; Jun, C. *Data Simulation of an Airborne Lidar System*; Turner, M.D., Kamerman, G.W., Eds.; SPIE: Bellingham, WA, USA, 2009; p. 73230C.
31. Wang, Y.; Xie, D.; Yan, G.; Zhang, W.; Mu, X. Analysis on the Inversion Accuracy of LAI Based on Simulated Point Clouds of Terrestrial LiDAR of Tree by Ray Tracing Algorithm. In Proceedings of the 2013 IEEE International Geoscience and Remote Sensing Symposium—IGARSS, Melbourne, Australia, 21–26 July 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 532–535.
32. Hodge, R.A. Using Simulated Terrestrial Laser Scanning to Analyse Errors in High-Resolution Scan Data of Irregular Surfaces. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 227–240. [CrossRef]
33. Dayal, S.; Goel, S.; Lohani, B.; Mittal, N.; Mishra, R.K. Comprehensive Airborne Laser Scanning (ALS) Simulation. *J. Indian. Soc. Remote Sens.* **2021**, *49*, 1603–1622. [CrossRef]
34. Bechtold, S.; Höfle, B. Helios: A Multi-Purpose Lidar Simulation Framework for Research, Planning and Training of Laser Scanning Operations with Airborne, Ground-Based Mobile and Stationary Platforms. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *III-3*, 161–168. [CrossRef]
35. Winiwarter, L.; Esmoris Pena, A.M.; Weiser, H.; Anders, K.; Martínez Sánchez, J.; Searle, M.; Höfle, B. Virtual Laser Scanning with HELIOS++: A Novel Take on Ray Tracing-Based Simulation of Topographic Full-Waveform 3D Laser Scanning. *Remote Sens. Env.* **2022**, *269*, 112772. [CrossRef]
36. Comesaña Cebral, L.J.; Martínez Sánchez, J.; Rúa Fernández, E.; Arias Sánchez, P. Heuristic Generation of Multispectral Labeled Point Cloud Datasets for Deep Learning Models. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *XLIII-B2-2022*, 571–576. [CrossRef]
37. RIEGL MiniVUX-1DL Data Sheet. Available online: [https://www.gtbi.net/wp-content/uploads/2021/06/riegl-minivux-1dl_folleto\(EN\).pdf](https://www.gtbi.net/wp-content/uploads/2021/06/riegl-minivux-1dl_folleto(EN).pdf) (accessed on 17 July 2023).
38. RIEGL VUX-1UAV Data Sheet. Available online: <http://www.riegl.com/products/unmanned-scanning/riegl-vux-1uav22/> (accessed on 4 January 2024).
39. Teledyne Optech Co. Teledyne Optech. 2024. Available online: <https://www.teledyneoptech.com/en/home/> (accessed on 4 January 2024).
40. Applanix Corp. Homepage. Available online: <https://www.applanix.com/> (accessed on 4 January 2024).
41. Carlotto, M.J. Effect of Errors in Ground Truth on Classification Accuracy. *Int. J. Remote Sens.* **2009**, *30*, 4831–4849. [CrossRef]
42. Comesaña-Cebral, L.; Martínez-Sánchez, J.; Lorenzo, H.; Arias, P. Individual Tree Segmentation Method Based on Mobile Backpack LiDAR Point Clouds. *Sensors* **2021**, *21*, 6007. [CrossRef]
43. Applied Geotechnologies Research Group ROADSENSE Dataset. Available online: https://universidavegigo-my.sharepoint.com/personal/geotech_uvigo_gal/_layouts/15/onedrive.aspx?id=/personal/geotech_uvigo_gal/Documents/PUBLIC%20DATA/DataSets/synthetic&ga=1 (accessed on 31 January 2024).
44. OSM2World Home Page. 2024. Available online: <https://osm2world.org/> (accessed on 4 January 2024).
45. OpenStreetMap Official Webpage. 2024. Available online: <https://www.openstreetmap.org/> (accessed on 4 January 2024).
46. Kumar, A.; Anders, K.; Winiwarter, L.; Höfle, B. Feature Relevance Analysis for 3D Point Cloud Classification Using Deep Learning. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *IV-2/W5*, 373–380. [CrossRef]
47. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
48. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
49. Smith, L.N. A Disciplined Approach to Neural Network Hyper-Parameters: Part 1—Learning Rate, Batch Size, Momentum, and Weight Decay. *arXiv* **2018**, arXiv:1803.09820.
50. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.-A. Deep Learning for Time Series Classification: A Review. *Data Min. Knowl. Discov.* **2018**, *33*, 917–963. [CrossRef]
51. Castillo, R.C.J.; Mendoza, R. On Smoothing of Data Using Sobolev Polynomials. *AIMS Math.* **2022**, *7*, 19202–19220. [CrossRef]
52. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.A.A.K.; Elhoseiny, M.; Ghanem, B. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23192–23204.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.