

## Article

# Deep Ordinal Classification in Forest Areas Using Light Detection and Ranging Point Clouds

Alejandro Morales-Martín <sup>1,\*</sup>, Francisco-Javier Mesas-Carrascosa <sup>2</sup>, Pedro Antonio Gutiérrez <sup>1</sup>, Fernando-Juan Pérez-Porras <sup>2</sup>, Víctor Manuel Vargas <sup>1</sup> and César Hervás-Martínez <sup>1</sup>

<sup>1</sup> Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, 14071 Córdoba, Spain; pagutierrez@uco.es (P.A.G.); vvargas@uco.es (V.M.V.); chervas@uco.es (C.H.-M.)

<sup>2</sup> Department of Graphic Engineering and Geomatics, University of Córdoba, Campus de Rabanales, 14071 Córdoba, Spain; ig2meca@uco.es (F.-J.M.-C.); o12pepof@uco.es (F.-J.P.-P.)

\* Correspondence: g52momaa@uco.es

**Abstract:** Recent advances in Deep Learning and aerial Light Detection And Ranging (LiDAR) have offered the possibility of refining the classification and segmentation of 3D point clouds to contribute to the monitoring of complex environments. In this context, the present study focuses on developing an ordinal classification model in forest areas where LiDAR point clouds can be classified into four distinct ordinal classes: ground, low vegetation, medium vegetation, and high vegetation. To do so, an effective soft labeling technique based on a novel proposed generalized exponential function (CE-GE) is applied to the PointNet network architecture. Statistical analyses based on Kolmogorov-Smirnov and Student's *t*-test reveal that the CE-GE method achieves the best results for all the evaluation metrics compared to other methodologies. Regarding the confusion matrices of the best alternative conceived and the standard categorical cross-entropy method, the smoothed ordinal classification obtains a more consistent classification compared to the nominal approach. Thus, the proposed methodology significantly improves the point-by-point classification of PointNet, reducing the errors in distinguishing between the middle classes (low vegetation and medium vegetation).



**Citation:** Morales-Martín, A.; Mesas-Carrascosa, F.-J.; Gutiérrez, P.A.; Pérez-Porras, F.-J.; Vargas, V.M.; Hervás-Martínez, C. Deep Ordinal Classification in Forest Areas Using Light Detection and Ranging Point Clouds. *Sensors* **2024**, *24*, 2168. <https://doi.org/10.3390/s24072168>

Academic Editor: Nicolas Baghdadi

Received: 21 February 2024

Revised: 20 March 2024

Accepted: 26 March 2024

Published: 28 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/>).

**Keywords:** LiDAR point cloud; Deep Learning; ordinal classification; soft labeling

## 1. Introduction

The importance of aerial Light Detection And Ranging (LiDAR), one of the most relevant remote-sensing tools for terrestrial data acquisition, has increased thanks to recent contributions in the field of Artificial Intelligence (AI). Within this area of knowledge, Deep Learning (DL) has played a key role since it has provided the scientific community with further tools to classify and segment 3D point clouds, thus benefiting a wide spectrum of fields, such as Forestry Engineering [1,2], Agricultural Engineering [3,4], and Urban Planning [5,6]. For example, concerning the former, LiDAR technology makes it possible to obtain information at a larger scale. This technology provides information about vegetation structure, density, canopy height model, and canopy percentage cover, which allows a thorough understanding of the forest. To achieve this, LiDAR data need to be previously filtered and classified to analyze forest areas [7].

According to the classification system developed by the American Society for Photogrammetry and Remote Sensing (ASPRS), vegetation can be classified into three classes: low, medium, and high [8]. Low vegetation is that which ranges from ground level up to 0.5 m; medium vegetation is that which ranges from 0.5 m up to 2 m; and high vegetation is that which is more than 2 m [9,10]. Following this differentiation, LiDAR point clouds are used to categorize the points corresponding to the ground in order to generate a Digital Elevation Model (DEM) [11]. Subsequently, thanks to the DEM, the height of the points is normalized, and classified as low vegetation, medium vegetation, or high vegetation. In this context, issues arise when normalized heights are close to 0.5 m or 2 m, as they

could be classified as either medium or high vegetation, respectively [12,13]. Once the points are classified, forestry metrics can be obtained, namely percentage cover, tree height, and percentiles [14]. This allows for the calculation of biomass [15], wood estimates [16], or fuel models, all of which help in designing high-precision forest fire prevention and firefighting models [17].

During the last few decades, numerous researchers have focused on classifying and segmenting 3D point clouds using DL approaches. This is because point clouds contain a larger amount of raw information—regarding space, color, etc.—when compared to images. In order to properly classify and segment point clouds, labeling techniques have been considered an essential stage. In the context of labeling point clouds, algorithms can be broadly divided into two groups: those based on methods that operate directly on the 3D point clouds without altering their original nature [18,19], and those based on methods that convert the point clouds to collections of images or voxel grids [20,21]. On the one hand, the first group employs a point-wise discriminative model to assign semantic labels to each element in the point cloud. This model operates on point features and is designed to be simple yet effective. For example, Ref. [22] suggested the use of geometrical and spectral features from the LiDAR point cloud for the semantic labeling task in urban scenarios. On the other hand, the second group employs multiview transformation approaches or volumetric methods to learn local and global features. For instance, Ref. [23] suggested the conversion of the point cloud into regularly distributed 2D images, which allows the classification of a point to be approached as a pixel classification problem. In the case of volumetric methods, Ref. [24] proposed the partition of a given LiDAR point cloud into regular voxel grids, using 3D Convolutional Neural Networks (3D-CNNs) to label each voxel according to the information of its centroid. However, the use of these transformations has been criticized because of the significant computational overhead they introduce, their tendency to increase model complexity, and the potential loss of valuable information.

In this context, to avoid problems with data conversion, many studies have used the PointNet neural network [25,26], which is the pioneering architecture proposed for 3D object recognition in indoor scenarios [27]. This network is specifically designed to process point cloud data, offering a highly adaptable framework with a vast capacity and minimal overhead, enabling efficient operations. For outdoor scenarios, different DL models have been reported, all based on PointNet-like architectures: PointNet++ [28], SE-PointNet++ [29], CropPointNet [30], etc. Meanwhile, other authors have suggested ameliorating the performance of PointNet by exploring the local structure of point clouds [31,32] or incorporating crucial handcrafted features into the deep neural network [33]. Despite all these improvements, the classification and segmentation of large-scale airborne point clouds in complex environments are still challenging [34,35]. For example, in Forestry Engineering, segmenting vegetation represents a challenge due to the intricate interplay between objects and the background, and the need to set thresholds for height differences to normalize and filter LiDAR point clouds. Thus, few studies have assessed the potential of DL-based methods in forest areas, with most of them related to tree species classification [36,37].

In any case, these point cloud labeling algorithms have primarily addressed the resolution of problems where the ordering of the labels has been ignored (nominal classification). Despite the success of the application of DL-based methods in nominal classification tasks in recent years [38,39], ordinal classification has been the focus of researchers in the recognition community. The aim of using ordinal classification is to predict the label of a given pattern, where a natural order among the different possible categories can be assumed. For instance, Ref. [40] aims to predict the human head pose angle (pitch, roll, and yaw) based on ordinal regression techniques with soft labels, operating on 3D point clouds derived from a depth image. In the field of Forestry Engineering, Ref. [41] predicts the number of strata in three datasets based on ordinal regression techniques to support forest management. This task is different from the classification of points as ground, low vegetation, medium vegetation, and high vegetation, which has not yet been explored.

The interest and novelty of the present study lie in developing ordinal classification models for forest areas and integrating them into aerial LiDAR point clouds to refine the per-point classification. The study uses an effective soft labeling technique applied to the PointNet deep network. The main interest of the proposed methodology, compared to the state of the art, is the use of a generalized exponential distribution, where two hyperparameters are introduced ( $p$  and  $\alpha$ ) for improving the flexibility of the distribution. By adjusting the value of these parameters, better results can be achieved for the real problem addressed.

The core concept is to enhance the quality of airborne point cloud segmentation by reducing errors in the labeling procedure, such as human operator variability and post-processing steps. The reason for using DL approaches in this research is to explore the potential of this advanced technique to ameliorate the classification of LiDAR data in complex forest environments. The use of DL algorithms can help extract more complex features and patterns from the data, leading to better classification results.

To summarize, this research has focused on the following contributions:

1. The development of an ordinal classification model that utilizes an effective soft labeling technique applied to the loss function using unimodal distributions to ameliorate the classification of LiDAR data.
2. The application of the proposed methodology in forest areas where LiDAR point clouds can be classified into four distinct ordinal labels: ground, low vegetation, medium vegetation, and high vegetation. In the field of Forestry Engineering, it is the first time that this kind of problem is treated as an ordinal regression problem. This contribution is particularly relevant for the forestry industry as it enables more accurate estimation of important forest metrics such as biomass and wood estimates.

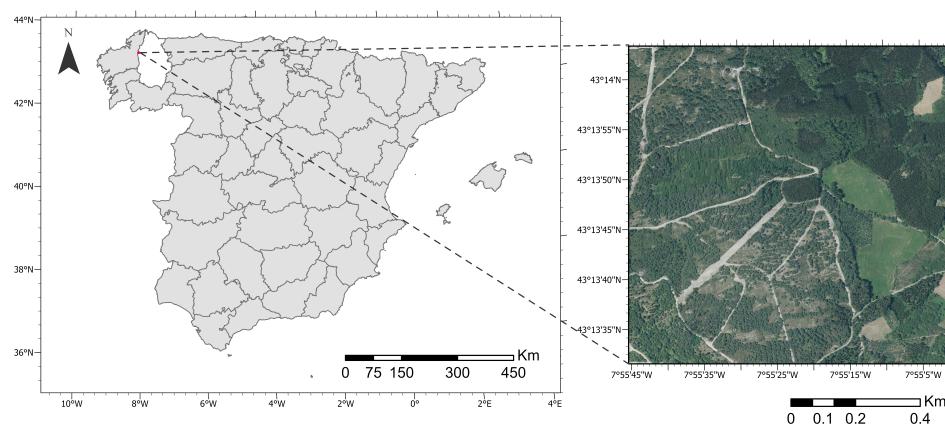
This manuscript is organized as follows: the LiDAR point cloud dataset employed, the core concepts such as “ordinal classification” and “soft labeling”, the proposed regularized loss function used for soft labeling (generalized exponential function), and the PointNet network architecture are described in Section 2; the experimental results are shown in Section 3 and discussed in Section 4; and, finally, the conclusions are drawn in Section 5.

## 2. Materials and Methods

### 2.1. LiDAR Data

The LiDAR point cloud generated covered a forest area of 1 km<sup>2</sup>, located in the province of Lugo, Spain (43°13'50'' N, 7°55'25'' W, WGS-84) (Figure 1). The aerial LiDAR point cloud, which comprises around 46 million points, had a point density of 49 points/m<sup>2</sup> and a point spacing of 0.143 m with six returns. Moreover, the LASer (LAS) format file of the point cloud contains several point attributes, from which the coordinates ( $x, y, z$ ), the red, green, and blue color values (RGB), the intensity ( $I$ ), and the return number ( $Rn$ ) were utilized in this work.

The LiDAR flight was performed on 18 February 2018 using an AS-350 B2 Eurocopter manned helicopter. This platform was equipped with a RIEGL laser scanner, model VQ-480i (Figure 2), which offers an 80 mm footprint. The Inertial Measurement Unit used was iMar Navigation, model iIMU-FSAS-HP-SI-SME1-SP, with an angular measurement range of ±500°/s, a drift lower than 0.1° × hour<sup>-1</sup>, and a resolution of 0.1 arcsec × LSB<sup>-1</sup> (Least Significant Bit), providing data at a rate of up to 500 Hz. The RGB camera was a Phase One, model IXU-RS 1000, with a resolution of 11,608 × 8708 pixels and a focal length equal to 50 mm. The GNSS modem utilized was a Javad TR-G3T, incorporating GPS, Galileo, GLONASS, and SBAS frequencies, connecting to up to 32 satellites with an Antcom antenna. The flight altitude of the mission was 300 m above ground level at a speed of 20 m × s<sup>-1</sup>. The system operated at a pulse repetition of 400 kHz and with a scan angle (FOV) of 60°, an altitude above ground level equal to 300 m with a reflectivity higher than 20% for a 30% side-lap overlap.



**Figure 1.** Location of the LiDAR data. The coordinates are georeferenced in the ETRS89/UTM Zone 29N coordinate system.



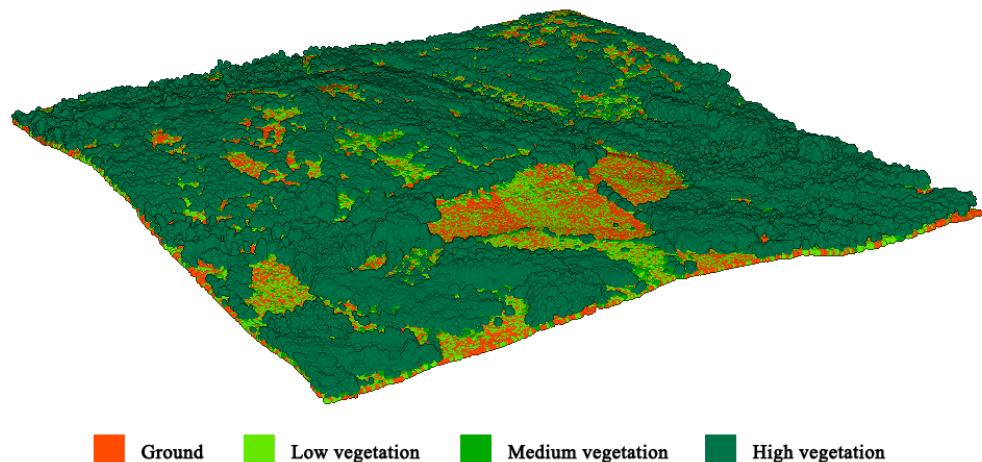
**Figure 2.** On the left, the laser scanner, two Phase One cameras (zenith and oblique), and the IMU. On the right, the system installed before the flight.

After acquiring the LiDAR data from the experimental site, Python's `pktk` package [42] was used to analyze, visualize, and extract features from the 3D point clouds. To ensure the reliability of the labels, conventional and well-established labeling methods were used. In this sense, two filtering methods were applied to label the point clouds as ground truth, using a semi-automatic labeling procedure to evaluate the classification results.

The first one referred to normalized heights of vegetation. Thanks to the Python's `laserchicken` package [43], a Digital Elevation Model (DEM) of the study area was generated. In the DEM, each point was assigned a normalized height ( $H$ ) value based on the height of the lowest point in a  $1 \times 1$  m neighborhood to distinguish vegetation classes. The elevation threshold was established as follows: point clouds from 0 to 0.5 m, from 0.5 to 2 m, and above 2 m were classified as low vegetation, medium vegetation, and high vegetation, respectively.

The second referred to return numbers in order to discriminate ground points from low vegetation points. This filtering was specifically used in areas where only two classes coexist: ground and low vegetation. In these areas, the emitted laser pulse of LiDAR technology first encounters the low vegetation and then the ground class. For this reason, this approach assumes that the first return belongs to the low vegetation class and the other returns belong to the ground class. As previously discussed, the LiDAR point cloud contained six returns. If the return value of a point was higher than or equal to 2, it would mean that the said point could be classified as ground. However, those points with  $Rn$  equal to 1 were classified as low vegetation.

All the selected LiDAR point clouds were refined by manually editing them using the interactive segmentation tool of CloudCompare [44]. After that, the LiDAR point clouds were unified and converted into an HDF5 file format [45]. The dataset contained two parts: data and labels. Concerning the former, the attributes of several points from four different categories were recorded. These attributes include  $x, y, z, RGB$  values,  $I, R$ , and  $H$ . In the label section, the labels were categorized into four semantic classes: ground, low vegetation, medium vegetation, and high vegetation (Figure 3).



**Figure 3.** Labels of the LiDAR point cloud. The legend represents the color corresponding to each category.

An amount of 2048 points was randomly sampled from 5738 blocks of size  $9 \times 9$  m without replacement. After that, the dataset was split into a 60/20/20 percent ratio as training, validation, and test sets, respectively (Table 1). Moreover, all the features were normalized— $x_n, y_n, z_n, R_n, G_n, B_n, I_n$  and  $H_n$ —except the  $R_n$  attribute, which was encoded as a one-hot numeric array— $Rn_1, Rn_2, Rn_3, Rn_4, Rn_5, Rn_6$ . Lastly, the point clouds in each block constituting the training, validation, and testing sets were converted into an HDF5 file. Each set includes the following: (1) an array of  $N \times 2048 \times 14$ , where  $N$  is the total number of segmented input blocks, 2048 is the number of points that are randomly sampled per block, and 14 corresponds to the 14-dimensional feature vector (Equation (1)); and (2) a categorical label encoded as a one-hot numeric array (Equation (2)).

$$p_i = (x_n, y_n, z_n, R_n, G_n, B_n, I_n, H_n, Rn_1, Rn_2, Rn_3, Rn_4, Rn_5, Rn_6). \quad (1)$$

$$C_q = \begin{cases} [1, 0, 0, 0] & \text{ground category,} \\ [0, 1, 0, 0] & \text{low vegetation category,} \\ [0, 0, 1, 0] & \text{medium vegetation category,} \\ [0, 0, 0, 1] & \text{high vegetation category,} \end{cases} \quad (2)$$

**Table 1.** The number of points in each category.

Category	Training	Validation	Test
Ground	2,578,600	863,096	835,056
Low vegetation	1,000,907	324,547	354,049
Medium vegetation	1,204,729	412,513	400,521
High vegetation	2,264,980	750,948	761,478
	7,049,216	2,351,104	2,351,104

## 2.2. Ordinal Classification

The purpose of ordinal classification problems is to predict the real class  $y$  based on an input  $K$ -dimensional vector  $x$  ( $x \in X \subseteq \mathbb{R}^K$ ). For this sort of problem, the dependent variable of the classification models is an ordinal scale variable chosen from a set of different categories ( $y \in Y = \{C_1, C_2, \dots, C_Q\}$ ) which have a natural ordering ( $C_1 \prec C_2 \prec \dots \prec C_Q$ ) associated with the real problem [46]. The precedence operator ( $C_i \prec C_j$ ) designates that the category  $C_i$  is prior to the class  $C_j$  in the ordinal scale and  $Q$  is the number of classes defined in the real problem.

Ordinal classification models, which exploit the ordering information in ordinal class attributes, can lead to better segmentation results by reducing misclassification errors in extreme classes, limiting errors to adjacent categories, and maximizing the number of properly classified patterns.

## 2.3. Soft Labeling

Soft labeling is an effective regularization technique that operates on the labels, enhancing the model's robustness during training. Instead of utilizing hard labels (1 for the target class and 0 for the others), soft labeling assigns some probability to each class [47]. When the labels are encoded as one-hot, the probability distribution of each class is defined as  $q(i) = \delta_{i,c}$ ,  $i$  being the predicted class with ranges from 1 to  $Q$ ,  $c$  the ground truth class, and  $\delta_{i,c}$  the Dirac delta, which equals 1 when  $i = c$  and 0 otherwise [48].

Based on the above, label smoothing can be incorporated into the standard cross-entropy loss function (Equation (3)). The cross-entropy loss function is considered one of the most popular methods for training deep neural networks, as it aims to maximize the likelihood of the correct prediction given the ground truth in the training set [49].

$$L = \sum_{i=1}^Q q(i) [-\log p(y = C_i | x)], \quad (3)$$

where  $p(y = C_i | x)$  is the probability predicted by the model of  $x$  belonging to the class  $y$  for each of the  $Q$  categories. By applying label smoothing, the  $q(i)$  in Equation (3) can be replaced with a soft version  $q'(i)$ :

$$q'(i) = (1 - \eta)\delta_{i,c} + \eta\left(\frac{1}{Q}\right). \quad (4)$$

A discrete uniform distribution is established associated with the number of classes  $Q$ , where  $\eta$  is a smoothing parameter defined in the range  $[0, 1]$  that controls the linear combination.

Within the context of ordinal classification, the misclassification of patterns is a common issue that occurs in adjacent classes in relation to the ground truth on the ordinal scale. To address this challenge and improve the accuracy of loss computation, it is recommended to substitute the use of a uniform distribution with unimodal distributions. The mode of the unimodal distributions should be positioned in the center of the interval for middle classes or at the upper or lower bounds for extreme classes [48]. Additionally, to favor more confident decisions, it is crucial for the probability distribution to exhibit a small variance and for the majority of its probability mass to be concentrated within the interval corresponding to the ground truth class [48].

Various low-variance distribution functions considered in previous studies [48,50–52] have ameliorated the performance of ordinal classifiers by applying a soft labeling technique regarding the standard one-hot encoding. Some of the proposed methods require experimental adjustment of different parameters. Ref. [50] proposed using a binomial distribution, which takes into account the number of classes and the probability. The mean and variance of the binomial distribution are given by different expressions, allowing for enhanced flexibility to place the mean in the center of the class interval while keeping the variance small. According to [51], the Poisson distribution represents an alternative

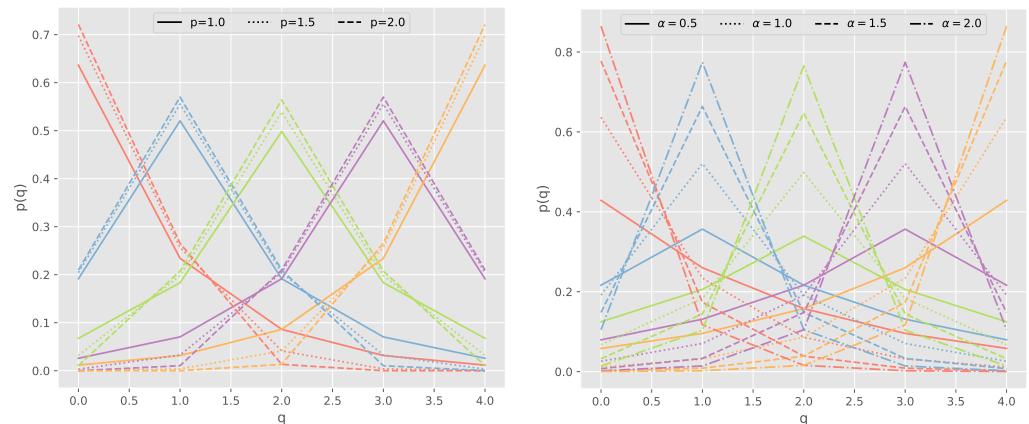
for modeling the probabilities. However, since the mean and variance of the Poisson distribution are both determined by the parameter  $\lambda$ , it is not possible to center the mean of the distribution in the class interval while achieving a small variance, leading to poor performance. In the work of [52], an exponential function followed by a Softmax normalization is explained. Besides its simplicity, this distribution flexibly adjusts the shape of the target label distribution in order to smooth the label with the unimodal distribution. Hence, it is feasible to manipulate the mean and variance of the exponential distribution to foster label predictions that are closely distributed to the ground truth class. Lastly, Ref. [48] proposed sampling from a beta distribution that is defined within the range of 0 to 1; thus, no normalization is mandatory and no high variance is achieved. However, an improvement of this distribution could be considered when the number of classes is low.

#### 2.4. Generalized Exponential Function

All the abovementioned distributions, especially when the number of classes is low, are characterized by high variance or lack the necessary flexibility to center the mode of each distribution within the interval corresponding to its class. For this reason, a generalized exponential function based on [52] is proposed:

$$f(q, p, \alpha) = e^{-\alpha|q-y|^p}, \quad (5)$$

where  $q$  and  $y$  designate the predicted and real classes, respectively, and  $1 \leq p \leq 2$  and  $0 < \alpha \leq 2$  are two hyperparameters that need to be adjusted experimentally. After that, a Softmax normalization procedure is employed to calculate the corresponding probabilities. Figure 4 presents the class distributions for the proposed generalized exponential function, where the color indicates the ground truth class of a given pattern, the  $x$ -axis specifies the class under examination, and the  $y$ -axis indicates the applied soft label. For each class, the distributions for  $p \in \{1.0, 1.5, 2.0\}$  and  $\alpha \in \{0.5, 1.0, 1.5, 2.0\}$  are represented.



**Figure 4.** The generalized exponential distribution was used to address a classification problem with five distinct classes. In the resulting visualization, the  $x$ -axis designates the evaluated class, while the  $y$ -axis displays the corresponding smoothed label value. The color coding corresponds to the ground truth class, with red representing class 0, blue representing class 1, green representing class 2, purple representing class 3, and orange representing class 4. Each line in the visualization represents the probability distribution for a specific true label. The line type indicates the different values that the hyperparameter  $p$  and the hyperparameter  $\alpha$  can take.

These probabilities replace the uniform distribution and are subsequently used to generate the corresponding  $q'(i)$  (Equation (4)). In this way, the standard definition of the loss function is formulated as:

$$L = \sum_{i=1}^Q q'(i) [-\log p(y = C_i | x)]. \quad (6)$$

Given that  $q'(i)$  presents a continuous decrease with respect to the farther distance from the ground truth class, it is considered as a weight of  $-\log p(y = C_i|x)$ .

In the present study, the uniform distribution is replaced with other low-variance distributions that provide more flexibility when centering each of the four categories in the problem. In these low-variance distributions,  $f(x, \theta)$  is the probability value sampled from a binomial, Poisson, beta, and generalized exponential plus a Softmax function:

$$q'(i) = (1 - \eta)\delta_{i,c} + \eta f(x, \theta). \quad (7)$$

To analyze the behavior of the proposed distribution, an ordinal classification model with four classes ( $Q = 4$ ) is considered. To find the hyperparameters  $(p, \alpha)$  from the generalized exponential function, one value is decided while fixing the other one, reducing the computational cost of simultaneously finding both values. The value for the hyperparameter  $p$  is obtained by fixing the hyperparameter  $\alpha$  (Equation (8)). After that, to obtain the value of the hyperparameter  $\alpha$ , the hyperparameter  $p$  is fixed (Equation (9)).

$$f(q, p, \alpha = 1) = e^{-|q-y|^p}, q = 1, \dots, Q. \quad (8)$$

$$f(q, p = 1, \alpha) = e^{-\alpha|q-y|}, q = 1, \dots, Q. \quad (9)$$

## 2.5. PointNet Network Architecture

A Convolutional Neural Network (CNN), namely PointNet [27], is used for the classification of LiDAR point clouds, as the main interest of this study is to offer a point-by-point classification. Nevertheless, some modifications in the inputs and outputs are included for the proper implementation of the original network [27,31].

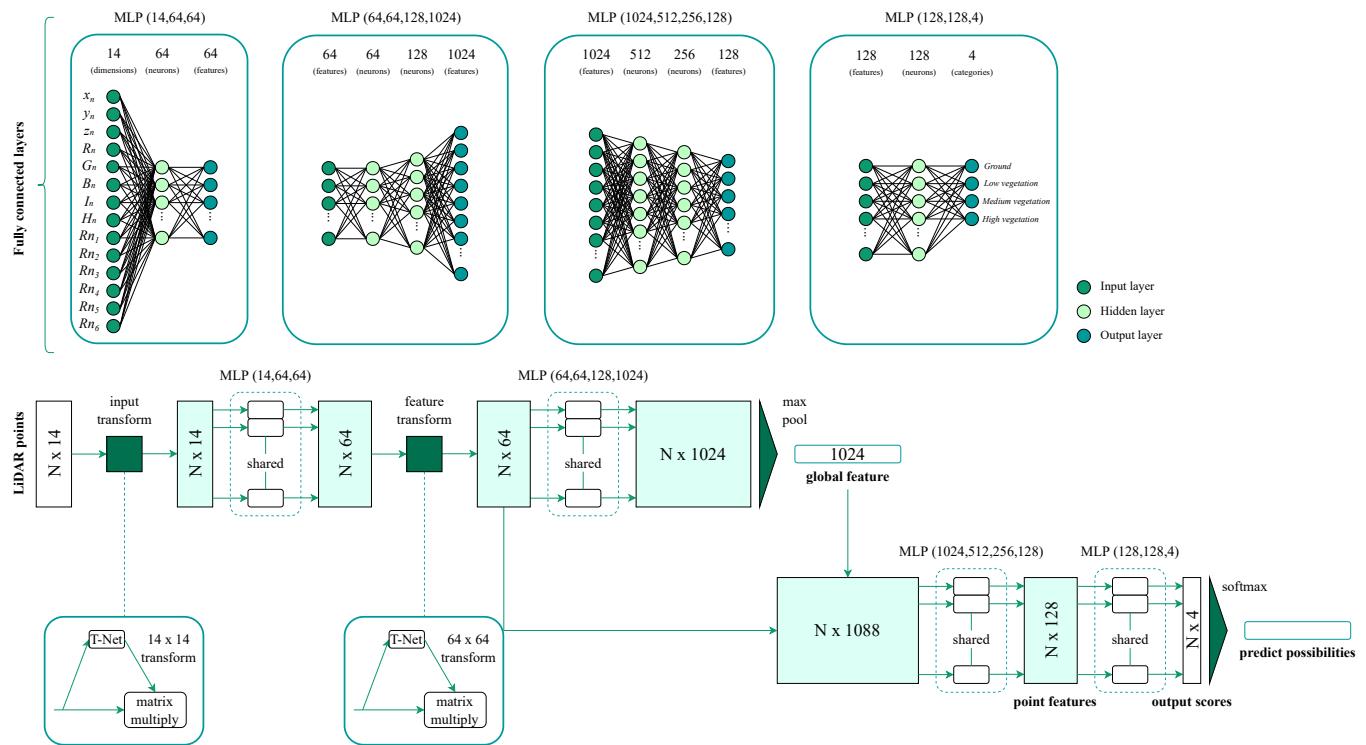
Firstly, all the point features employed in the present study—the coordinates  $x, y$ , and  $z$ ; the spectral values red, green, and blue ( $RGB$ ); the intensities ( $I$ ); and the height of each point within the  $1 \times 1$  m block ( $H$ )—are normalized— $x_n, y_n, z_n, R_n, G_n, B_n, I_n$  and  $H_n$ —instead of only adding only the normalized coordinates of the point [31].

Secondly, considering that the return number attribute ( $Rn$ ) is a discrete variable defined in the range [1, 6] (the LiDAR system provides six returns), it is encoded as a one-hot integer array— $Rn_1, Rn_2, Rn_3, Rn_4, Rn_5, Rn_6$ . Thus, this encoding allows for evaluating the impact of  $Rn$  in the model, as the other parameters are measured on a continuous scale.

Thirdly, prior to data training and following the design of the PointNet model, the point cloud is partitioned into non-overlapping blocks of size  $9 \times 9$  m. The number of points to be randomly sample per block is set to 2048 in order to define consistent data batches. According to [31], if the number of points within the block exceeds the maximum allowed, a random selection process is employed to determine which points to include. Hence, the network is fed by a  $B \times N \times 2048 \times 14$  array, where  $B$  denotes the batch size,  $N$  denotes the total number of segmented input blocks, 2048 denotes the number of points randomly sampled per block, and 14 represents the 14-dimensional feature vector ( $x_n, y_n, z_n, R_n, G_n, B_n, I_n, H_n, Rn_1, Rn_2, Rn_3, Rn_4, Rn_5, Rn_6$ ).

Fourthly, as the described model is intended to segment 3D point clouds, a vector output is obtained for each point that contains the probabilities of each class for the given point.

Lastly, a Softmax function is used in the output layer to attain the probability of point cloud prediction in each class within the range [0, 1]: ground, low vegetation, medium vegetation, and high vegetation (Figure 5). In addition, a dropout rate of 0.3 is applied for the last two dense layers with ReLU activation [27].



**Figure 5.** The architecture of the refined DL network PointNet. In the resulting visualization, there are four boxes in the upper part displaying the MLP layers and two boxes in the lower part displaying the T-Net layers. The CNN processes the 14-dimensional features that contain  $N$  grids and 2048 points within the block. The T-net layers are responsible for transforming the input and point features, while the MLPs and the max-pooling layers are responsible for aggregating and extracting the point features. The individual point-level and global features are concatenated to acquire the point features. The output includes the predicted probabilities of belonging to each of the four classes.

## 2.6. Experiment Settings

The loss function was weighted following the methodology described in [48] to alleviate the imbalance of different categories: the minority classes had a higher weight than the main classes.

Five different loss functions used for the optimization algorithm and derived from the standard cross-entropy loss were established: categorical cross-entropy (CCE), cross-entropy loss with binomial regularization (CE-B) [52], cross-entropy loss with Poisson regularization (CE-P) [52], cross-entropy loss with  $\beta$  regularization (CE- $\beta$ ) [48], and the proposed cross-entropy loss with generalized exponential regularization (CE-GE).

A confusion matrix  $Q \times Q$  was used to analyze the potential of the classifiers (Table 2). In the context of an ordinal classification problem involving  $Q$  categories and  $n$  patterns,  $n_{qk}$  denotes the frequency with which a classifier assigns patterns belonging to class  $q$  to class  $k$ ,  $n_{q\bullet}$  denotes the total number of patterns in class  $q$ , and  $n_{\bullet k}$  denotes the total number of patterns predicted to be in class  $k$  [46].

Different metrics chosen for evaluation were applied to quantitatively compare and analyze the classification performance. These metrics included the Quadratic Weighted Kappa (QWK), the Minimum Sensitivity (MS), the Mean Absolute Error (MAE), the Correct Classification Rate (CCR), the 1-off accuracy (1-off) and the Mean Intersection-Over-Union (mIoU) [46,48]. Thanks to Python's dordinal package [53], the ordinal metrics MS and 1-off were computed.

**Table 2.** Confusion matrix.

		Predicted Class					
		1	...	k	...	Q	
True class	1	$n_{11}$	...	$n_{1k}$	...	$n_{1Q}$	$n_{1\bullet}$
	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	$q$	$n_{q1}$	...	$n_{qk}$	...	$n_{qQ}$	$n_{q\bullet}$
	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	Q	$n_{Q1}$	...	$n_{Qk}$	...	$n_{QQ}$	$n_{Q\bullet}$
	$n_{\bullet 1}$	...	$n_{\bullet k}$	...	$n_{\bullet Q}$		$n$

Note. Table adapted from [46].

$$\text{QWK} = \frac{\rho_{o(w)} - \rho_{e(w)}}{1 - \rho_{e(w)}}, \quad (10)$$

$$\rho_{o(w)} = \frac{1}{n} \sum_{q=1}^Q \sum_{k=1}^Q w_{qk} n_{qk}, \quad (11)$$

$$\rho_{e(w)} = \frac{1}{n^2} \sum_{q=1}^Q \sum_{k=1}^Q w_{qk} n_{q\bullet} n_{\bullet k}, \quad (12)$$

$$\text{MS} = \min \left\{ S_q = \frac{n_{qq}}{n_{q\bullet}}; q = 1, \dots, Q \right\}, \quad (13)$$

$$\text{MAE} = \frac{1}{n} \sum_{q,k=1}^Q |q - k| n_{qk} = \frac{1}{n} \sum_{i=1}^n e(x_i), \quad (14)$$

$$\text{CCR} = \frac{1}{n} \sum_{q=1}^Q n_{qq}, \quad (15)$$

$$\text{1-off} = \frac{1}{n} \sum_{q=1}^Q \sum_{k=\max(0,q-1)}^{\min(Q,q+1)} n_{qk}, \quad (16)$$

$$mIoU = \frac{1}{n} \sum_{q=1}^Q \frac{S_{qq}}{\sum_{k=1}^Q S_{qk} + \sum_{k=1}^Q S_{kq} - S_{qq}}, \quad (17)$$

where  $w_{qk} = |q - k|$ ,  $q$  and  $k$  being the true and the predicted category, respectively;  $S_q$  denotes the sensitivity of the  $q$ -th class;  $e(x_i) = |\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|$  denotes the distance between the true ( $y_i$ ) and the predicted ( $y_i^*$ ) ranks;  $\mathcal{O}(C_q) = q$  denotes the position of each category in the ordinal scale;  $S_{qq}$ ,  $S_{qk}$ , and  $S_{kq}$  denote the number of true positives, false positives, and false negatives, respectively.

The proposed methodology was implemented based on the TensorFlow Keras frameworks using the novel ordinal dataset. The experiments followed a hold-out scheme which was executed 20 times with 20 distinct seeds to ensure a comprehensive evaluation. This rigorous approach enabled the attainment of 20 independent executions. The training process was run for 150 epochs with a batch size of 64 and a momentum of 0.9 [54]. The number of training epochs and the batch size were increased with respect to the previous study [54], from 30 to 150 and from 32 to 64, respectively, due to the amount of points per-block sampled. The model was optimized using the Adam method [55], with an initial learning rate of 0.001. In addition, decay on the learning rate was also employed when the validation loss had not decreased for 10 epochs, multiplying it by a factor of 0.85 until it reached  $10^{-7}$ .

### 3. Results

#### 3.1. Calculating $p$ and $\alpha$ from Generalized Exponential Function

A preliminary experiment was conducted to obtain the hyperparameters  $p$  and  $\alpha$  from the generalized exponential function (Appendix A). In Table A1, the results obtained by defining  $p$  as  $p \in \{1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0\}$  showed that the optimum value of  $p$  was 1, based on the six metrics previously described. After that, the value of the hyperparameter  $\alpha$  was also selected by fixing  $p = 1$ . Likewise, as shown in Table A2, the results obtained by defining  $\alpha$  as  $\alpha \in \{0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00\}$  showed that the optimum value was 1.

Considering these results, the performance of each method is evaluated in Table 3. All the values introduced include the average and standard deviation of the 20 executions. The best result for each metric is highlighted in bold font, while the second best is in italics. Hence, the CE-GE method exhibited the best mean results compared to other methodologies.

**Table 3.** Comparison of methodologies for each metric.

Method <sup>1</sup>	QWK	MS	MAE	CCR	1-off	mIoU
CCE	$0.9535_{0.0258}$	$0.8077_{0.1281}$	$0.0920_{0.0495}$	$0.9378_{0.0310}$	$0.9703_{0.0187}$	$0.8757_{0.0393}$
CE-B	$0.9460_{0.0277}$	$0.7687_{0.1301}$	$0.1095_{0.0585}$	$0.9244_{0.0402}$	$0.9665_{0.0189}$	$0.8419_{0.0800}$
CE-P	$0.9395_{0.0485}$	$0.7451_{0.2223}$	$0.1198_{0.0951}$	$0.9209_{0.0562}$	$0.9609_{0.0361}$	$0.8381_{0.1054}$
CE- $\beta$	$0.9504_{0.0335}$	$0.7746_{0.1702}$	$0.0999_{0.0674}$	$0.9320_{0.0422}$	$0.9682_{0.0253}$	$0.8583_{0.0817}$
CE-GE	<b><math>0.9667_{0.0119}</math></b>	<b><math>0.8515_{0.0600}</math></b>	<b><math>0.0667_{0.0225}</math></b>	<b><math>0.9539_{0.0147}</math></b>	<b><math>0.9794_{0.0084}</math></b>	<b><math>0.8885_{0.0303}</math></b>

<sup>1</sup> CCE: categorical cross-entropy; CE-B: cross-entropy loss with binomial regularization; CE-P: cross-entropy loss with Poisson regularization; CE- $\beta$ : cross-entropy loss with  $\beta$  regularization; and CE-GE: cross-entropy loss with generalized exponential regularization.

#### 3.2. Statistical Analysis

To ensure the reliability of these results, statistical analyses were conducted, in which each of the metrics presented in Section 2.6 was studied independently. The Kolmogorov–Smirnov test [56] for the QWK, MS, MAE, CCR, 1-off, and mIoU was performed using the results obtained from the 20 executions. The  $p$ -values achieved were higher than 0.050, indicating that the null hypothesis of normality for the distribution of the values of these metrics was accepted. Consequently, each pair of algorithms, CE-GE with the other methods, was compared for each metric by employing Student's t-test for paired data [56]. A significance level of  $\alpha = 0.100$  was established and its corresponding adjustment based on the number of comparisons was added. Since four algorithms were compared, the corrected significance level was calculated ( $\alpha = \frac{0.100}{4} = 0.025$ ).

Table 4 shows the mean and the standard deviation ( $SD$ ) of each method's executions on the test set, together with the  $p$ -values of the test. In general terms, statistically significant differences were observed in the results of the paired t-test. For all of the metrics, the CE-GE method exhibited the best mean results. Taking into account the QWK and 1-off accuracy metrics, the CE-GE achieved better mean results than CCE, CE-B, and CE-P for  $\alpha = 0.050$  and CE- $\beta$  for  $\alpha = 0.100$ . On the other hand, for the MS and mIoU metrics, the CE-GE reached better mean results than CE-B and CE-P for  $\alpha = 0.050$  and CE- $\beta$  for  $\alpha = 0.100$ . Finally, based on the MAE and CCR metrics, the CE-GE got better mean results than the other methods for  $\alpha = 0.050$ .

**Table 4.** Paired sample  $t$ -test results for the QWK, MS, MAE, CCR, 1-off, and mIoU metrics.

Metric	Paired Sample <sup>1</sup>	Mean	SD	$p$ -Value
QWK	CE-GE vs. CCE	0.014	0.026	0.024
	CE-GE vs. CE- $\beta$	0.017	0.041	0.081
	CE-GE vs. CE-B	0.021	0.027	0.003
	CE-GE vs. CE-P	0.028	0.052	0.023

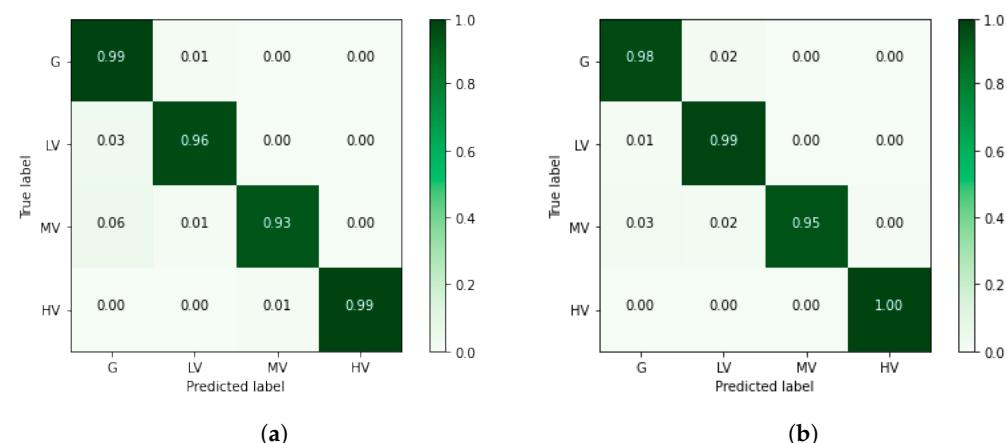
**Table 4.** Cont.

Metric	Paired Sample <sup>1</sup>	Mean	SD	p-Value
MS	CE-GE vs CCE	0.044	0.142	0.182
	CE-GE vs. CE- $\beta$	0.077	0.172	0.061
	CE-GE vs. CE-B	0.083	0.130	0.010
	CE-GE vs. CE-P	0.106	0.216	0.040
MAE	CE-GE vs CCE	-0.025	0.050	0.034
	CE-GE vs. CE- $\beta$	-0.033	0.071	0.050
	CE-GE vs. CE-B	-0.043	0.055	0.002
	CE-GE vs. CE-P	-0.053	0.094	0.021
CCR	CE-GE vs CCE	0.016	0.030	0.024
	CE-GE vs. CE- $\beta$	0.022	0.046	0.044
	CE-GE vs. CE-B	0.030	0.039	0.003
	CE-GE vs. CE-P	0.033	0.056	0.017
1-off	CE-GE vs CCE	0.010	0.019	0.033
	CE-GE vs. CE- $\beta$	0.012	0.030	0.089
	CE-GE vs. CE-B	0.014	0.018	0.003
	CE-GE vs. CE-P	0.019	0.038	0.035
mIoU	CE-GE vs CCE	0.013	0.049	0.174
	CE-GE vs. CE- $\beta$	0.030	0.083	0.082
	CE-GE vs. CE-B	0.047	0.082	0.012
	CE-GE vs. CE-P	0.050	0.111	0.022

<sup>1</sup> CCE: categorical cross-entropy; CE-B: cross-entropy loss with binomial regularization; CE-P: cross-entropy loss with Poisson regularization; CE- $\beta$ : cross-entropy loss with  $\beta$  regularization; and CE-GE: cross-entropy loss with generalized exponential regularization.

### 3.3. LiDAR Classification

The performance of PointNet for each category of points is evaluated by analyzing the confusion matrices of two methods: the proposed best alternative (CE-GE + Softmax) and the standard method (CCE + Softmax). Normalized confusion matrices from the smoothed ordinal classification and the nominal classification are shown in Figure 6. Additionally, Tables 5 and 6 present an in-depth analysis of the confusion matrices.



**Figure 6.** Normalized confusion matrices from (a) the nominal classification and (b) the smoothed ordinal classification. Each number represents the ratio of patterns to the total number of patterns in the class. G: ground; LV: low vegetation; MV: medium vegetation; and HV: high vegetation.

**Table 5.** Confusion matrix from the nominal classification results.

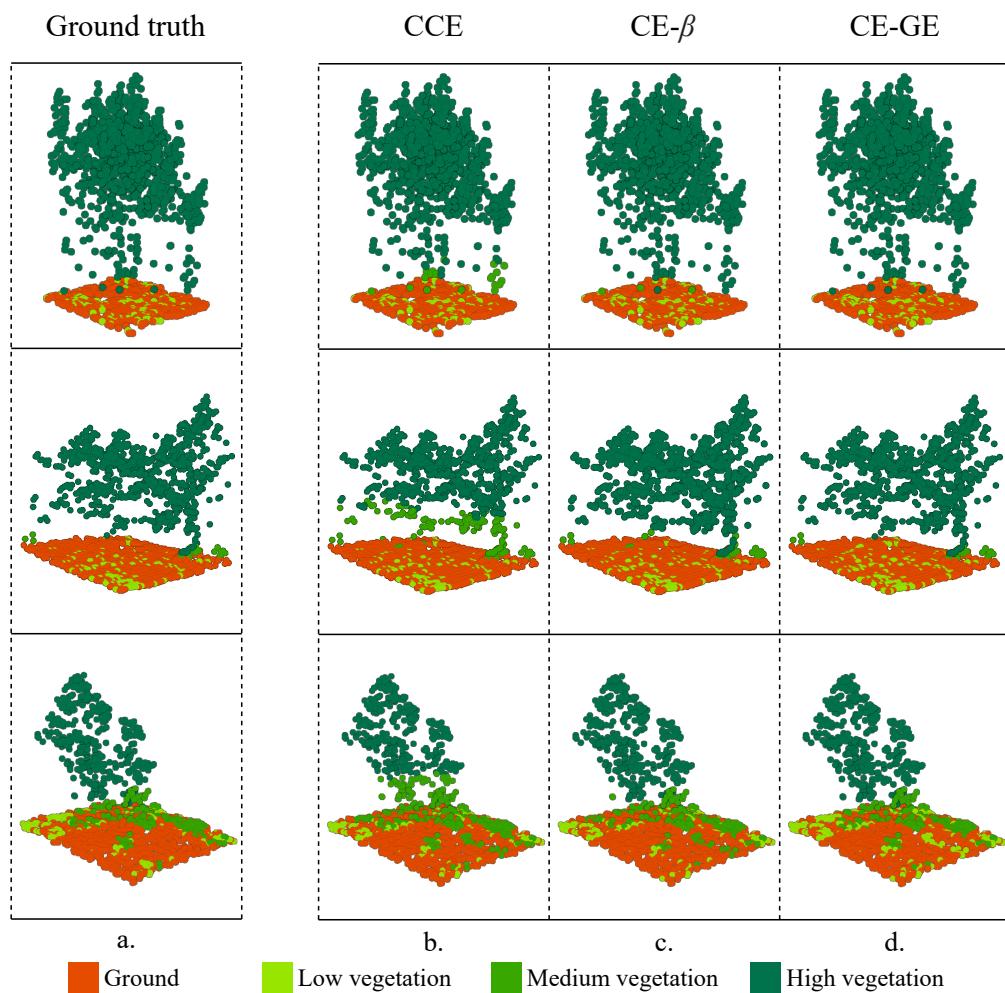
	Ground	Low Vegetation	Medium Vegetation	High Vegetation
Ground	827,294	7667	95	0
Low vegetation	12,368	341,629	52	0
Medium vegetation	23,385	4259	372,858	19
High vegetation	0	0	8083	753,395

**Table 6.** Confusion matrix from the smoothed ordinal classification results.

	Ground	Low Vegetation	Medium Vegetation	High Vegetation
Ground	815,576	17,125	2355	0
Low vegetation	3973	350,035	41	0
Medium vegetation	12,363	7723	378,657	1778
High vegetation	0	0	1258	760,220

Using nominal classification (Figure 6b and Table 5), the ground class was the only one correctly labeled when compared to the confusion matrix from the smoothed ordinal classification. In this sense, in nominal classification, distinguishing between ground and vegetation classes is easier than in ordinal classification because the former is treated as a binary classification problem (e.g., ground vs. vegetation). On the other hand, smoothed ordinal classification uses the ordering information between the four classes, reinforcing the decision boundaries among low, medium, and high vegetation classes. Therefore, in the case of the smoothed ordinal classification (Figure 6a and Table 6), points classified as low vegetation, medium vegetation, and high vegetation were correctly labeled for the majority of the test set.

As an example, Figure 7 shows a partial view of the segmentation results of the PointNet model in the study area. As can be observed, the best alternative proposed, CE-GE + Softmax, which incorporates an ordinal structure and effective soft labeling simultaneously, performed better than the standard method CCE + Softmax. As previously mentioned, most of the differences obtained by the nominal and the smoothed classification results are related to the misclassification of the middle classes. In Figure 7, the main differences between the ground truth, the nominal method (CCE), and the ordinal methodologies (CE- $\beta$  and CE-GE) were depicted. The proposed method CE-GE + Softmax was closer to the ground truth than both the nominal point cloud-based method and the beta ordinal method as it generated the correct label predictions for most point clouds.



**Figure 7.** Segmentation result: (a) ground truth, (b) nominal methodology (CCE), (c) ordinal methodology based on beta distribution ( $\text{CE-}\beta$ ), and (d) proposed ordinal methodology based on generalized exponential distribution (CE-GE).

#### 4. Discussion

The classification of LiDAR point clouds has gained popularity due to the high demand for accurate classifications. To contribute to this field, previous research has increasingly utilized DL methods, training models with different point attributes to simulate operator labeling [57]. Nevertheless, most studies have focused on point attributes rather than the nature of the labels themselves [38,39]. This could be because most studies have focused on urban scenarios where large-scale point clouds often contain distinct classes such as ground, vegetation, building, or water, which can easily be distinguished based on LiDAR point characteristics [58]. In these scenarios, previous studies have achieved an overall accuracy of 87% in classifying the following classes: ground, vegetation, and building [31,32].

In this study, the classification of LiDAR point clouds in forest areas was addressed. In these areas where vegetation is the dominant class, it becomes crucial to accurately distinguish between different vegetation strata [6]. Points belonging to the middle classes are difficult to classify because their normalized height presents similar geographical distribution and topological features compared to other categories [6]. In this sense, the low vegetation class can easily be confused with ground or medium vegetation, and the medium vegetation category can be difficult to differentiate from low vegetation or high vegetation. In our proposal, to determine the most suitable approach for accurately classifying these categories, five different methodologies (CCE, CE-B, CE-P, CE- $\beta$ , and CE-GE) were compared [48,50–52].

Based on the results obtained, the CE-GE methodology was found to be the most effective approach for tackling this classification problem. By assigning a probability to the classes (soft labeling) and considering the order between the categories, the issue of misclassification regarding the middle classes (low vegetation and medium vegetation) was mitigated. Utilizing the CE-GE methodology, the study achieved an overall accuracy of 95% in classifying the following classes: ground, low vegetation, medium vegetation, and high vegetation. Nevertheless, it is important that the data must have a clear ordering between the different classes, which is the main challenge that may arise when applying the ordinal model.

For those cases where an ordinal structure is not found in the labels, future projects could incorporate this methodology as a pre-classification technique for the following four classes: ground, low vegetation, medium vegetation, and high vegetation. For upcoming projects, the proposed methodology could explore agricultural areas and urban green spaces, in addition to forest scenarios.

## 5. Conclusions

The present study focused on optimizing the existing DL network PointNet for classification in complex environments. It analyzed the effects of adding an ordinal classification and an effective soft labeling technique to point-by-point classification. Specifically, for four categorical classes, the study proposed the application of a CE-GE function, which helps achieve more accurate labels. In addition, a comparison was performed with different loss functions (CCE, CE-B, CE-P, and CE- $\beta$ ) to demonstrate the benefits of the smoothed ordinal classification for the PointNet model.

Given the results obtained, the following conclusions should be highlighted:

- Significant differences in the paired t-test were observed, in which the CE-GE method reached the best mean results for all the metrics (QWK, MS, MAE, CCR, and 1-off) when compared to the other methods.
- Regarding the confusion matrices of the best alternative conceived (CE-GE + Softmax) and the standard method (CCE + Softmax), the smoothed ordinal classification achieved notably better results than the nominal one. Consequently, our methodology reduce the errors in distinguishing between the middle classes (low vegetation and medium vegetation).

Overall, the proposed methodology has significantly improved the point-by-point classification of PointNet. The application of the suggested methodology in forest areas was possible, contributing to obtaining forestry metrics up ahead, which allows thorough understanding of the forest, as it permits the calculation of biomass, wood estimates, and fuel models for high-precision forest fire prevention and firefighting. Therefore, as long as LiDAR point clouds have an underlying ordinal structure, they could be used in a diverse range of forest ecology and management applications, from monitoring land-cover changes in complex environments to identifying tree species in future scenarios.

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

**Funding:** This research was funded by European Commission, project Test and Experiment Facilities for the Agri-Food Domain (AgriFoodTEF), DIGITAL-2022-CLOUD-AI-02, grant number 101100622; by ENIA International Chair in Agriculture, University of Córdoba, grant number TSI-100921-2023-3; by Agencia Española de Investigación (España), grant number PID2020-115454GB-C22/AEI/10.13039/501100011033; by European Social Fund (ESF), Operational Programme for Youth

Employment (POEJ), grant number EJ21-TIC148-AGR124; and by FPU Predoctoral Program of the Spanish Ministry of Science, Innovation and Universities (MCIU), grant number FPU21/03433.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets presented in this article are not readily available because the data are part of an ongoing study.

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

## Appendix A. Hyperparameter from the Generalized Exponential Function

**Table A1.** Results comparing different values of the hyperparameter  $p$  from the generalized exponential function.

$p$	QWK <sup>1</sup>	MS <sup>1</sup>	MAE <sup>1</sup>	CCR <sup>1</sup>	1-off <sup>1</sup>	mIoU <sup>1</sup>
1.0	<b>0.9667</b> <sub>0.0119</sub>	<b>0.8515</b> <sub>0.0600</sub>	<b>0.0667</b> <sub>0.0225</sub>	<b>0.9539</b> <sub>0.0147</sub>	<b>0.9794</b> <sub>0.0084</sub>	<b>0.8885</b> <sub>0.0311</sub>
1.1	0.9518 <sub>0.0358</sub>	0.7942 <sub>0.1782</sub>	0.0963 <sub>0.0727</sub>	0.9341 <sub>0.0505</sub>	0.9698 <sub>0.0225</sub>	0.8615 <sub>0.0899</sub>
1.2	0.9464 <sub>0.0324</sub>	0.7910 <sub>0.1122</sub>	0.1055 <sub>0.0607</sub>	0.9289 <sub>0.0382</sub>	0.9658 <sub>0.0230</sub>	0.8505 <sub>0.0708</sub>
1.3	0.9517 <sub>0.0265</sub>	0.7911 <sub>0.1293</sub>	0.0961 <sub>0.0489</sub>	0.9343 <sub>0.0298</sub>	0.9698 <sub>0.0196</sub>	0.8593 <sub>0.0586</sub>
1.4	0.9482 <sub>0.0417</sub>	0.8016 <sub>0.1312</sub>	0.1010 <sub>0.0699</sub>	0.9317 <sub>0.0407</sub>	0.9673 <sub>0.0300</sub>	0.8581 <sub>0.0706</sub>
1.5	0.9620 <sub>0.0187</sub>	0.8397 <sub>0.0705</sub>	0.0788 <sub>0.0359</sub>	0.9433 <sub>0.0241</sub>	0.9780 <sub>0.0121</sub>	0.8775 <sub>0.0471</sub>
1.6	0.9622 <sub>0.0203</sub>	0.8464 <sub>0.0711</sub>	0.0780 <sub>0.0366</sub>	0.9438 <sub>0.0232</sub>	0.9782 <sub>0.0144</sub>	0.8790 <sub>0.0453</sub>
1.7	0.9525 <sub>0.0379</sub>	0.8197 <sub>0.1140</sub>	0.0939 <sub>0.0688</sub>	0.9362 <sub>0.0431</sub>	0.9701 <sub>0.0262</sub>	0.8662 <sub>0.0761</sub>
1.8	0.9635 <sub>0.0169</sub>	0.8446 <sub>0.0943</sub>	0.0743 <sub>0.0310</sub>	0.9479 <sub>0.0196</sub>	0.9779 <sub>0.0125</sub>	0.8862 <sub>0.0405</sub>
1.9	0.9473 <sub>0.0630</sub>	0.8056 <sub>0.1244</sub>	0.1007 <sub>0.1052</sub>	0.9320 <sub>0.0654</sub>	0.9680 <sub>0.0378</sub>	0.8637 <sub>0.0986</sub>
2.0	0.9537 <sub>0.0377</sub>	0.8039 <sub>0.1553</sub>	0.0934 <sub>0.0722</sub>	0.9354 <sub>0.0453</sub>	0.9712 <sub>0.0272</sub>	0.8644 <sub>0.0817</sub>

<sup>1</sup> The best performance of each metric is highlighted in bold face font, while the second one is in italics.

**Table A2.** Results comparing different values of the hyperparameter  $\alpha$  from the generalized exponential function.

$\alpha$	QWK <sup>1</sup>	MS <sup>1</sup>	MAE <sup>1</sup>	CCR <sup>1</sup>	1-off <sup>1</sup>	mIoU <sup>1</sup>
0.25	0.9360 <sub>0.0440</sub>	0.7578 <sub>0.1665</sub>	0.1237 <sub>0.0861</sub>	0.9180 <sub>0.0585</sub>	0.9587 <sub>0.0285</sub>	0.8343 <sub>0.0994</sub>
0.50	0.9477 <sub>0.0353</sub>	0.7775 <sub>0.1754</sub>	0.1040 <sub>0.0709</sub>	0.9295 <sub>0.0453</sub>	0.9673 <sub>0.0260</sub>	0.8516 <sub>0.0856</sub>
0.75	0.9459 <sub>0.0332</sub>	0.7585 <sub>0.1676</sub>	0.1103 <sub>0.0692</sub>	0.9241 <sub>0.0471</sub>	0.9657 <sub>0.0228</sub>	0.8400 <sub>0.0925</sub>
1.00	<b>0.9667</b> <sub>0.0119</sub>	<b>0.8515</b> <sub>0.0600</sub>	<b>0.0667</b> <sub>0.0225</sub>	<b>0.9539</b> <sub>0.0147</sub>	<b>0.9794</b> <sub>0.0084</sub>	<b>0.8885</b> <sub>0.0311</sub>
1.25	0.9585 <sub>0.0158</sub>	0.8167 <sub>0.0824</sub>	0.0842 <sub>0.0284</sub>	0.9409 <sub>0.0175</sub>	0.9749 <sub>0.0122</sub>	0.8721 <sub>0.0353</sub>
1.50	0.9548 <sub>0.0297</sub>	0.8046 <sub>0.1298</sub>	0.0903 <sub>0.0558</sub>	0.9382 <sub>0.0348</sub>	0.9715 <sub>0.0217</sub>	0.8682 <sub>0.0682</sub>
1.75	0.9582 <sub>0.0158</sub>	0.8169 <sub>0.0769</sub>	0.0827 <sub>0.0277</sub>	0.9436 <sub>0.0173</sub>	0.9738 <sub>0.0118</sub>	0.8776 <sub>0.0355</sub>
2.00	0.9616 <sub>0.0183</sub>	0.8345 <sub>0.0966</sub>	0.0778 <sub>0.0360</sub>	0.9459 <sub>0.0236</sub>	0.9764 <sub>0.0132</sub>	0.8821 <sub>0.0483</sub>

<sup>1</sup> The best performance of each metric is highlighted in bold face font, while the second one is in italics.

## References

- Li, L.; Mu, X.; Chianucci, F.; Qi, J.; Jiang, J.; Zhou, J.; Chen, L.; Huang, H.; Yan, G.; Liu, S. Ultrahigh-resolution Boreal Forest Canopy Mapping: Combining UAV Imagery and Photogrammetric Point Clouds in a Deep-learning-based Approach. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *107*, 102686. [[CrossRef](#)]
- Luo, Z.; Zhang, Z.; Li, W.; Chen, Y.; Wang, C.; Nurunnabi, A.A.M.; Li, J. Detection of Individual Trees in UAV LiDAR Point Clouds Using a Deep Learning Framework Based on Multichannel Representation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [[CrossRef](#)]
- Jin, S.; Su, Y.; Gao, S.; Wu, F.; Ma, Q.; Xu, K.; Ma, Q.; Hu, T.; Liu, J.; Pang, S.; et al. Separating the Structural Components of Maize for Field Phenotyping Using Terrestrial LiDAR Data and Deep Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 2644–2658. [[CrossRef](#)]
- Zhou, C.; Ye, H.; Sun, D.; Yue, J.; Yang, G.; Hu, J. An automated, high-performance approach for detecting and characterizing broccoli based on UAV remote-sensing and transformers: A case study from Haining, China. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *114*, 103055. [[CrossRef](#)]
- Fang, L.; Sun, T.; Wang, S.; Fan, H.; Li, J. A graph attention network for road marking classification from mobile LiDAR point clouds. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *108*, 102735. [[CrossRef](#)]

6. Kalinicheva, E.; Landrieu, L.; Mallet, C.; Chehata, N. Predicting Vegetation Stratum Occupancy from Airborne LiDAR Data with Deep Learning. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102863. [[CrossRef](#)]
7. Moudrý, V.; Klápková, P.; Fogl, M.; Gdulová, K.; Barták, V.; Urban, R. Assessment of LiDAR Ground Filtering Algorithms for Determining Ground Surface of Non-natural Terrain Overgrown with Forest and Steppe Vegetation. *Measurement* **2020**, *150*, 107047. [[CrossRef](#)]
8. ASPRS. *LAS Specification Version 1.4-R13*; The American Society for Photogrammetry and Remote Sensing: Bethesda, MD, USA, 2013.
9. Vosselman, G.; Maas, H.G. Adjustment and Filtering of Raw Laser Altimetry Data. In Proceedings of the OEEPE workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Elevation Models, Stockholm, Sweden, 1–3 March 2001.
10. Sithole, G.; Vosselman, G. Experimental Comparison of Filter Algorithms for Bare-Earth Extraction from Airborne Laser Scanning Point Clouds. *ISPRS J. Photogramm. Remote Sens.* **2004**, *59*, 85–101. [[CrossRef](#)]
11. Yunfei, B.; Guoping, L.; Chunxiang, C.; Xiaowen, L.; Hao, Z.; Qisheng, H.; Linyan, B.; Chaoyi, C. Classification of LIDAR Point Cloud and Generation of DTM from LIDAR Height and Intensity Data in Forested Area. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 313–318.
12. Shan, J.; Aparajithan, S. Urban DEM Generation from Raw LiDAR Data Vegetation. *Photogramm. Eng. Remote Sens.* **2005**, *71*, 217–226. [[CrossRef](#)]
13. Samadzadegan, F.; Hahn, M.; Bigdeli, B. Automatic Road Extraction from LiDAR Data Based on Classifier Fusion. In Proceedings of the Joint Urban Remote Sensing Event, Shanghai, China, 20–22 May 2009; pp. 1–6. [[CrossRef](#)]
14. Andersen, H.E.; McGaughey, R.J.; Reutebuch, S.E. Estimating Forest Canopy Fuel Parameters Using LiDAR Data. *Remote Sens. Environ.* **2005**, *94*, 441–449. [[CrossRef](#)]
15. Gleason, C.J.; Im, J. Forest Biomass Estimation from Airborne LiDAR Data Using Machine Learning Approaches. *Remote Sens. Environ.* **2012**, *125*, 80–91. [[CrossRef](#)]
16. Dassot, M.; Constant, T.; Fournier, M. The Use of Terrestrial LiDAR Technology in Forest Science: Application Fields, Benefits and Challenges. *Ann. For. Sci.* **2011**, *68*, 959–974. [[CrossRef](#)]
17. González-Olabarria, J.; Rodríguez, F.; Fernández-Landa, A.; Mola-Yudego, B. Mapping Fire Risk in the Model Forest of Urbión Based on Airborne LiDAR Measurements. *For. Ecol. Manag.* **2012**, *282*, 149–256. [[CrossRef](#)]
18. Wang, Z.; Zhang, L.; Zhang, L.; Li, R.; Zheng, Y.; Zhu, Z. A Deep Neural Network With Spatial Pooling (DNNSP) for 3-D Point Cloud Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4594–4604. [[CrossRef](#)]
19. Zhao, P.; Guan, H.; Li, D.; Yu, Y.; Wang, H.; Gao, K.; Marcato Junior, J.; Li, J. Airborne Multispectral LiDAR Point Cloud Classification with a Feature Reasoning-based Graph Convolution Network. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *105*, 102634. [[CrossRef](#)]
20. Shichao, J.; Yanjun, S.; Shang, G.; Fangfang, W.; Tianyu, H.; Jin, L.; Wenkai, L.; Wang, D.; Chen, S.; Jiang, Y.; et al. Deep Learning: Individual Maize Segmentation From Terrestrial Lidar Data Using Faster R-CNN and Regional Growth Algorithms. *Front. Plant Sci.* **2018**, *9*, 866. [[CrossRef](#)]
21. Chen, Q.; Wang, L.; Waslander, S.L.; Liu, X. An end-to-end shape modeling framework for vectorized building outline generation from aerial images. *ISPRS J. Photogramm. Remote Sens.* **2020**, *170*, 114–126. [[CrossRef](#)]
22. Ramiya, A.; Nidamanuri, R.R.; Krishnan, R. Semantic labelling of urban point cloud data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *XL-8*, 907–911. [[CrossRef](#)]
23. Hu, X.; Yuan, Y. Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud. *Remote Sens.* **2016**, *8*, 730. [[CrossRef](#)]
24. Huang, J.; You, S. Point Cloud Labeling Using 3D Convolutional Neural Network. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2670–2675. [[CrossRef](#)]
25. Kowalcuk, Z.; Szymański, K. Classification of objects in the LIDAR point clouds using Deep Neural Networks based on the PointNet model. *IFAC-PapersOnLine* **2019**, *52*, 416–421. [[CrossRef](#)]
26. Eroshenkova, D.A.; Terekhov, V.I.; Khusnetdinov, D.R.; Chumachenko, S.I. Automated Determination of Forest-Vegetation Characteristics with the Use of a Neural Network of Deep Learning. In *Advances in Neural Computation, Machine Learning, and Cognitive Research III*; Kryzhanovsky, B., Dunin-Barkowski, W., Redko, V., Tiumentsev, Y., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 295–302. [[CrossRef](#)]
27. Qi, C.; Su, H.; Mo, K.; Guibas, L. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [[CrossRef](#)]
28. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
29. 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](#)]
30. Jayakumari, R.; Nidamanuri, R.; Ramiya, A. Object-level Classification of Vegetable Crops in 3D LiDAR Point Cloud using Deep Learning Convolutional Neural Networks. *Precis. Agric.* **2021**, *22*, 1617–1633. [[CrossRef](#)]
31. Soilán, M.; Lindenbergh, R.; Riveiro, B.; Sánchez-Rodríguez, A. PointNet for the Automatic Classification of Aerial Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *IV-2/W5*, 445–452. [[CrossRef](#)]

32. Soilán, M.; Riveiro, B.; Balado, J.; Arias, P. Comparison of Heuristic and Deep Learning-Based Methods for Ground Classification from Aerial Point Clouds. *Int. J. Digit. Earth* **2020**, *13*, 1.115–1.134. [CrossRef]
33. Hsu, P.H.; Zhuang, Z.Y. Incorporating Handcrafted Features into Deep Learning for Point Cloud Classification. *Remote Sens.* **2020**, *12*, 3713. [CrossRef]
34. Gamal, A.; Husodo, A.Y.; Jati, G.; Alhamidi, M.R.; Ma’sum, M.A.; Ardhianto, R.; Jatmiko, W. Outdoor LiDAR Point Cloud Building Segmentation: Progress and Challenge. In Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS), Depok, Indonesia, 23–25 October 2021; pp. 1–6. [CrossRef]
35. Nurunnabi, A.; Teferle, F.; Li, J.; Lindenbergh, R.; Parvaz, S. Investigation of Pointnet for Semantic Segmentation of Large-Scale Outdoor Point Clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *46*, 397–404. [CrossRef]
36. Briechle, S.; Krzystek, P.; Vosselman, G. Silvi-Net – A Dual-CNN Approach for Combined Classification of Tree Species and Standing Dead Trees from Remote Sensing Data. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *98*, 102292. [CrossRef]
37. Chen, X.; Jiang, K.; Zhu, Y.; Wang, X.; Yun, T. Individual Tree Crown Segmentation Directly from UAV-Borne LiDAR Data Using the PointNet of Deep Learning. *Forests* **2021**, *12*, 131. [CrossRef]
38. Winiwarter, L.; Mandlburger, G.; Schmohl, S.; Pfeifer, N. Classification of ALS point clouds using end-to-end deep learning. *PFG-Photogramm. Remote Sens. Geoinf. Sci.* **2019**, *87*, 75–90. [CrossRef]
39. Chen, Y.; Wu, R.; Yang, C.; Lin, Y. Urban vegetation segmentation using terrestrial LiDAR point clouds based on point non-local means network. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *105*, 102580. [CrossRef]
40. Xiao, S.; Sang, N.; Wang, X.; Ma, X. Leveraging Ordinal Regression with Soft Labels For 3d Head Pose Estimation From Point Sets. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 1883–1887. [CrossRef]
41. Arvidsson, S.; Gullstrand, M. Predicting Forest Strata from Point Clouds Using Geometric Deep Learning. Master’s Thesis, Department of Computing, JTH, Jönköping, Sweden, 2021.
42. HERE Europe B.V. PPTK 0.1.1 Documentation Point Processing Toolkit. 2018. Available online: <https://heremaps.github.io/pptk/index.html> (accessed on 15 February 2024).
43. Netherlands eScience Center. Laserchicken 0.4.2 Documentation. 2019. Available online: <https://laserchicken.readthedocs.io/en/latest/> (accessed on 15 February 2024).
44. Girardeau-Montaut, D. CloudCompare: 3D Point Cloud and Mesh Processing Software. Version 2.12.0. 2021. Available online: <https://www.danielgm.net/cc/> (accessed on 15 February 2024).
45. Manduchi, G. Commonalities and Differences Between MDSplus and HDF5 Data Systems. *Fusion Eng. Des.* **2010**, *85*, 583–590. [CrossRef]
46. Cruz-Ramírez, M.; Hervás-Martínez, C.; Sánchez-Monedero, J.; Gutiérrez, P. Metrics to Guide a Multi-objective Evolutionary Algorithm for Ordinal Classification. *Neurocomputing* **2014**, *135*, 21–31. [CrossRef]
47. Zhang, C.B.; Jiang, P.T.; Hou, Q.; Wei, Y.; Han, Q.; Li, Z.; Cheng, M.M. Delving Deep Into Label Smoothing. *IEEE Trans. Image Process.* **2021**, *30*, 5984–5996. [CrossRef] [PubMed]
48. Vargas, V.; Gutiérrez, P.; Hervás-Martínez, C. Unimodal Regularisation based on Beta Distribution for Deep Ordinal Regression. *Pattern Recognit.* **2022**, *122*, 108310. [CrossRef]
49. Li, L.; Doroslovački, M.; Loew, M.H. Approximating the Gradient of Cross-Entropy Loss Function. *IEEE Access* **2020**, *8*, 111626–111635. [CrossRef]
50. Pinto da Costa, J.F.; Alonso, H.; Cardoso, J.S. The Unimodal Model for the Classification of Ordinal Data. *Neural Netw.* **2008**, *21*, 78–91. [CrossRef]
51. Beckham, C.; Pal, C. Unimodal Probability Distributions for Deep Ordinal Classification. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 411–419.
52. Liu, X.; Fan, F.; Kong, L.; Diao, Z.; Xie, W.; Lu, J.; You, J. Unimodal Regularized Neuron Stick-Breaking for Ordinal Classification. *Neurocomputing* **2020**, *388*, 34–44. [CrossRef]
53. Bérchez-Moreno, F.; Barbero, J.; Vargas, V.M. Deep Learning Utilities Library. 2023. Available online: <https://dlordinal.readthedocs.io/en/latest/index.html> (accessed on 15 February 2024).
54. Yousefhussien, M.; Kelbe, D.J.; Ientilucci, E.J.; Salvaggio, C. A Multi-Scale Fully Convolutional Network for Semantic Labeling of 3D Point Clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 191–204. [CrossRef]
55. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
56. Dodge, Y. *The Concise Encyclopedia of Statistics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
57. Zhang, L.; Zhang, L. Deep Learning-Based Classification and Reconstruction of Residential Scenes From Large-Scale Point Clouds. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1887–1897. [CrossRef]
58. Zhou, Y.; Ji, A.; Zhang, L.; Xue, X. Sampling-attention deep learning network with transfer learning for large-scale urban point cloud semantic segmentation. *Eng. Appl. Artif. Int.* **2023**, *117*, 105554. [CrossRef]