

Article

Disturbance Analysis in the Classification of Objects Obtained from Urban LiDAR Point Clouds with Convolutional Neural Networks

Jesús Balado ^{1,2,*} , Pedro Arias ¹ , Henrique Lorenzo ¹  and Adrián Mejjide-Rodríguez ³

¹ CINTECX, GeoTech Group, Universidade de Vigo, 36310 Vigo, Spain; parias@uvigo.es (P.A.); hlorenzo@uvigo.es (H.L.)

² GIS Technology Section, Faculty of Architecture and the Built Environment, Delft University of Technology, 2628 BL Delft, The Netherlands

³ Escola de Enxeñaría Industrial, Universidade de Vigo, 36310 Vigo, Spain; amejjide@alumnos.uvigo.es

* Correspondence: jbalado@uvigo.es

Abstract: Mobile Laser Scanning (MLS) systems have proven their usefulness in the rapid and accurate acquisition of the urban environment. From the generated point clouds, street furniture can be extracted and classified without manual intervention. However, this process of acquisition and classification is not error-free, caused mainly by disturbances. This paper analyses the effect of three disturbances (point density variation, ambient noise, and occlusions) on the classification of urban objects in point clouds. From point clouds acquired in real case studies, synthetic disturbances are generated and added. The point density reduction is generated by downsampling in a voxel-wise distribution. The ambient noise is generated as random points within the bounding box of the object, and the occlusion is generated by eliminating points contained in a sphere. Samples with disturbances are classified by a pre-trained Convolutional Neural Network (CNN). The results showed different behaviours for each disturbance: density reduction affected objects depending on the object shape and dimensions, ambient noise depending on the volume of the object, while occlusions depended on their size and location. Finally, the CNN was re-trained with a percentage of synthetic samples with disturbances. An improvement in the performance of 10–40% was reported except for occlusions with a radius larger than 1 m.

Keywords: mobile laser scanning; mobile mapping systems; LiDAR; occlusions; point density; noise



Citation: Balado, J.; Arias, P.; Lorenzo, H.; Mejjide-Rodríguez, A. Disturbance Analysis in the Classification of Objects Obtained from Urban LiDAR Point Clouds with Convolutional Neural Networks. *Remote Sens.* **2021**, *13*, 2135. <https://doi.org/10.3390/rs13112135>

Academic Editor:
Mohammad Awrangjeb

Received: 26 April 2021
Accepted: 27 May 2021
Published: 28 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

Although point clouds have proven to be a data source with many advantages, they also have some limitations that cause undesired behaviour in processing algorithms. The main disturbances in data acquisition are strong point density variations, noise, and occlusions. These disturbances reach such an extent that a data pre-processing phase is usually required to correct anomalies by density reductions, outliers and hole filling [1].

Point density is one of the main characteristics to consider for each point cloud [2]. The average density and the point distribution vary according to the type of laser scanning [3]. The density defines the number of points in a specific space and it limits the measurement and detection of the elements. Many authors prefer to work directly with high-density point clouds, ensuring there are always enough points to identify the desired targets [4]. However, the density in the point cloud is not constant, since point density varies with the distance to the laser and with the geometry of the environment. In [5], the authors reported a variation up to 41 times in point density between the highest and the lowest density areas of a Mobile Laser Scanning (MLS) acquisition. The distance between points is a determining factor in point cloud processing. Segmentation or classification methods focus the feature extraction in point-to-point relations, either by nearest neighbours, spherical neighbourhood, raster, voxels, or convolutions [6–11]. When extracting features, it is

important to consider both the point density to be handled and the possible variations to give robustness to the proposed method.

Noise is another common disturbance in point clouds, although it does not appear as significant as point density. Isolated points of noise can be found in the acquisitions due to isolated reflections. Besides, a slight amount of noise may appear on the surface of the objects depending on the LiDAR quality [12]. Noise does not affect all methods equally, for example, methods based on edge detection are more sensitive than those based on region growing [13]. To deal with noise, many authors design and implement methods for denoising and removing outliers [14,15]. Some methods are especially robust and filter high-density noise, specific to certain atmospheric and meteorological conditions, such as fog, rain, or airborne dust [16–18], even though usually in those environments, acquisitions are no longer performed. Another option is to design or adapt the methods so that they can deal with a certain amount of noise in the point cloud [19].

Occlusions are a geometrical problem related to the visibility of objects in the environment. The most common occlusions are related to occlusions between several objects (one object occludes another), but also an object can partially occlude itself (most of the time, one side of the object is not acquired). Although occlusions are a common problem in all environments [20], there are no specific datasets for working with occlusions. In autonomous driving, the occlusions considered most important to avoid collisions are those affecting pedestrians, objects already with few acquired points [21]. Few authors who focus on this problem choose to generate synthetic occlusions. Zachmann and Scamati [22] generated occlusions in Aerial Laser Scanning (ALS) data to detect and classify targets under trees. Despite the great difficulty of considering occlusions in point cloud processing, Habib et al. [23] designed a method to classify objects based on the interpretation of occlusion in ALS data.

Generally, works that assume the existence of disturbances, analyse them and take advantage of them achieve better results [5,24]. Although normally, it is easier and more straightforward to work with undisturbed data to show the advantages of the proposed methods in ideal situations, without anomalies that worsen the perception of the results.

The objective of this work is to evaluate how these three disturbances (point density variations, noise, and occlusions) affect the classification of urban objects in point clouds. The classification method is based on the generation of multi-views from point clouds, coupled with the use of a Convolutional Neural Network (CNN). Prior to the conversion of the point cloud into images, synthetic disturbances are generated automatically with different degrees of intensity. Finally, the CNN is re-trained with new objects with synthetic disturbances and the results are compared. To the best of the authors' knowledge, there are no other works that generate and analyse the effect of these three disturbances in a classification method based on multi-view and CNN. The major contributions of the work are listed below:

- Automated generation of point density variation and noise addition at different intensity levels.
- Automated generation of occlusions at five positions with different sizes.
- Analysis of accuracy and confusion in ten classes of urban objects: bench, car, lamppost, motorbike, pedestrian, traffic light, traffic sign, tree, wastebasket, and waste container.
- Analysis of misclassifications by re-training the CNN with synthetic errors.

This paper is structured as follows. In Section 2, the method for image generation, classification and automated disturbance generation is explained. In Section 3, the case study is presented, and the results are analysed and discussed. Section 4 concludes this paper.

2. Method

The method for the classification of the objects in point clouds used in this work is based on multi-view extraction. Multiview-based methods require lower computational cost and enable further data augmentation than working directly on unordered point

clouds, while taking advantage of the wide variety of convolutional neural network (CNN) architectures available [25].

Since disturbances are inherent to the acquisition of point clouds, and the generation of disturbances in this work is performed on the point clouds, even if they are later converted to images. The workflow of the experiment is shown in Figure 1. Disturbances can also influence the extraction and individualisation of objects; however, the behaviour of other algorithms in the presence of disturbances will not be analysed in this paper, as we focus exclusively on classification.

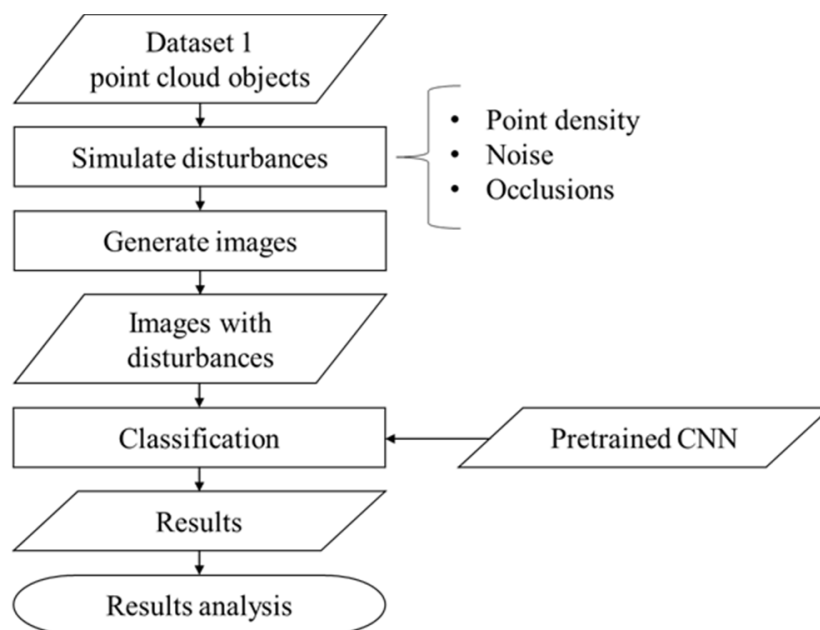


Figure 1. Workflow of the experiment.

2.1. Image Generation

The generation of the images is adapted from [26]. The selected view of acquired objects to project in an image is the lateral view, as it provides more information and improves the success rates in the classification. In addition, only the lateral view is used to the maximum extent to improve visualisation and classification, and to make the disturbances more visible and not hidden by overlapping points. To generate the images in maximum lateral perspective, the point cloud object $P(P_X P_Y P_Z)$ is rotated on the Z-axis based on the maximum distribution of points in the XY plane. The orientation of the points calculated by Principal Component Analysis is applied on the P_X and P_Y components (Equation (1)). From the eigen vectors, the angle of rotation α between the P_X and P_Y components is calculated (Equation (2)). Then, α is used to generate a 3D rotation R matrix over Z, which is applied to the input object point cloud P (Equation (3)).

$$\text{cov}(\overline{P_{XY}}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \frac{\overline{P_{XY}} \overline{P_{XY}}^T}{n-1} \quad (1)$$

$$\text{Where } \overline{P_{XY}} = P_{XY} - \text{mean}(P_{XY})$$

$$\alpha = \text{atan}(c_{12}/c_{11}) \quad (2)$$

$$P_{\text{rotated}} = PR = P \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Once the object point cloud is rotated so that its maximum extent is shown on the Y-plane, the image is generated on the Y-plane. The P_Y component (indicative of depth) of the rotated point cloud is removed. A grid of cells (pixels) is generated on which the cloud

points are structured. The size of the grid is defined by the size of the CNN input. The generated image is a binary image: black in the pixels whose cells contain at least one point and white in the pixels whose cells are empty. Although various cloud characteristics (such as intensity) can be extracted to generate a greyscale image, binary images are generated instead of greyscale images because from [26] it is concluded that the most important factor for the classification of the point cloud is the silhouette and not so much the colour. The result is an image where the silhouette and the object are well defined and contrasted against the white background. The process of image generation is illustrated in Figure 2.

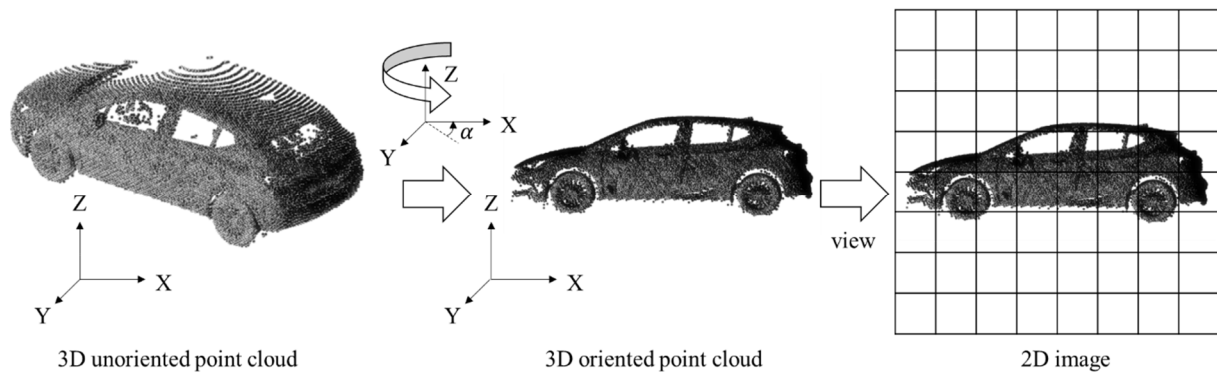


Figure 2. Generation of 2D images in maximum lateral perspective from 3D unoriented point cloud.

2.2. Point Density Reduction

Point density depends on the proximity and orientation of the object to the MLS; therefore, different urban objects have different densities even if they are acquired with the same MLS device. Density reduction is applied on each point cloud before generating the images. Before applying a density reduction, the current average point density must be known to establish values for effective density reduction. In this article, density is calculated as the average point-to-point distance [27] between the $k = 5$ nearest points with the *knn* algorithm, assuming regular scanning patterns.

Point density reduction is performed using the 3D grid method or voxelization [28]. The input point cloud is enclosed in a bounding box, which is divided into 3d cells of side l , with equal width, depth, and height. The value of l is variable and increases, as the distance between output points and point density decreases. Once the input cloud is structured in a 3D grid, the points are filtered by cell. If the cell contains one point, this point is retained. If the cell contains more than one point, the one nearest to the cell centre is selected. In the case of empty cells, no points are selected [29].

2.3. Noise Addition

Ambient noise is produced by airborne dust or rain. This noise is measured as in the previous section. The noise point density is calculated as the average distance from a point to the five nearest points with the *knn* algorithm. However, the generation of noise points is performed by the number of points enclosed in the bounding box object. Since each object has a different volume and the points are generated randomly, it is very difficult to generate noise of an exact point density value d for the test. For this reason, an iterative noise generation process is chosen, generating an initial number of points n which is then adjusted (Figure 3).

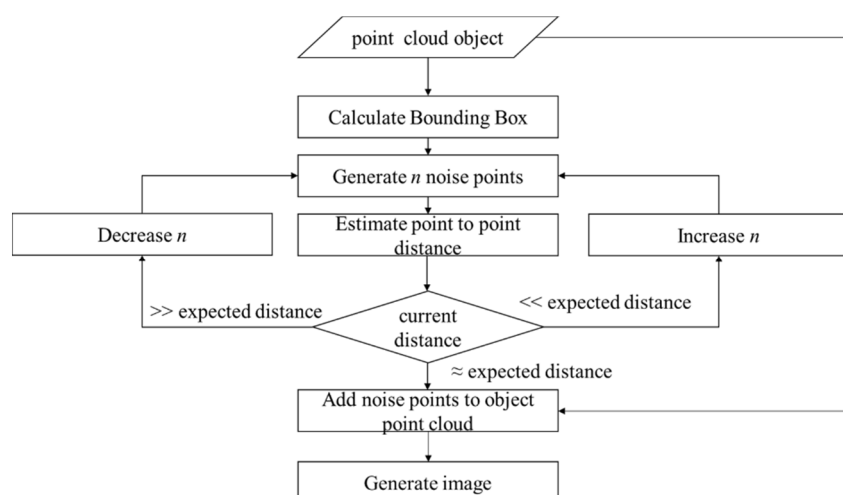


Figure 3. Generation and addition of noise points to point cloud objects.

Before generating any number of noise points, a test is conducted to establish a relationship between the number of points and the point density in 1 m^3 . For 500 points, an average distance of 0.115 m is measured with $k = 5$. The process starts by calculating the bounding box of the object to obtain the volume. From the volume, n points are generated and the average distance between $k = 5$ nearest neighbours is calculated. In case of coincidence of the desired and calculated point density ($\pm 10\%$), the noise points are added to the object point cloud and the image is generated. In case the current density is outside the threshold, the number of points is increased/decreased by 10% accordingly. The process iterates until the point density is within the threshold.

2.4. Occlusion Generation

Occlusions are caused by the existence of an object between the laser and the surface to be acquired. The occluding element can be an object other than the occluded object, or the acquired object itself (hidden unacquired side of the occluding object). The occlusions are based on two variables, location and size, unlike the previously generated disturbances which only depend on one variable.

Location is relevant as it can erase key shapes of the object even if the occlusion is small. Occlusion location describes the area or part of the object where the occlusion is located. To uniform the possible locations in urban objects with large geometry variations, it is decided to locate in each object five different occlusion origins on which occlusions of different sizes are generated. The authors assume that five occlusion origins cover most of the object to distinguish occlusion effects. The location of the occlusion origins is selected using the k -means algorithm applied to the object cloud [13]. The k -means algorithm clusters the point cloud into k groups where each point belongs to the group whose mean value is the nearest. The central point of each group is selected as the origin of the occlusion.

The size of the occlusion indicates how much of the object is hidden. The generation of the occlusion is performed as a sphere from each origin calculated above. Therefore, from the cloud of the input object, points with a radius to the origin smaller than r are detected and removed. Even though this method generates exclusively round-shaped occlusions in the image projection, for the CNN the loss of object information is more relevant than the shape of the occluded object's boundaries.

3. Experiments

3.1. Dataset and Pre-Trained CNN

The dataset was composed of 200 objects acquired by LYNX Mobile Mapper of Optech [30] in the cities of Vigo (Spain) and Porto (Portugal). In order not to propa-

gate errors from the process of object segmentation and individualisation, the objects were manually extracted from the point cloud. However, there are numerous techniques for the extraction of objects from the urban point cloud. One option is to detect ground and façade planes to remove the corresponding points and break the continuity between objects; then, the objects are individualised with connected components [26]. Other options are to structure the cloud into super voxels [31], to cluster points with superpoint graphs [32] or to implement techniques based on Deep Learning [33]. The ten selected classes were: bench, car, lamppost, motorbike, pedestrian, traffic light, traffic sign, tree, wastebasket, and waste container. A total of twenty objects were selected from each class. Ten objects per class were used for the generation of disturbances and analysis of results. All selected objects were checked for correct classification without disturbances by the pre-trained CNN, in order to start from a correct classification and to find the degree of disturbance for the first misclassification. The other ten objects per class were used for the CNN re-training (Section 3.5).

The method selected to test analyse the disturbance was presented in [26]. Although there are many methods for point cloud object classification, it was not possible to analyse them all in the present work, so a state-of-the-art method with good classification rates was chosen, based on the use of views and projections to classify the point cloud as an image with Deep Learning. The pre-trained CNN was described in [26]. The network architecture is an Inception V3 [34]. The softmax layer was changed to obtain a $1 \times 1 \times 10$ output corresponding to the number of classes (Figure 4). The samples used for training were (for each class) 450 images from internet sources (Google Images and ImageNet) and 50 images of objects in point clouds. An additional 50 images of objects in point clouds were used in the validation. The point clouds for pre-training were different from those used in disturbance generation and re-training.

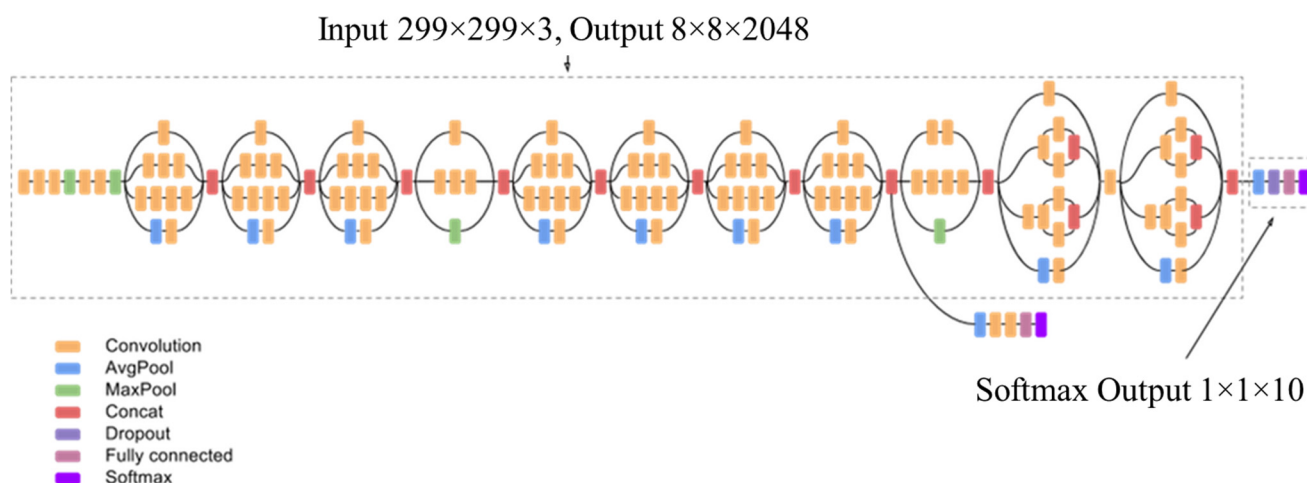


Figure 4. Inception V3 architecture modified from [34].

3.2. Density Results and Analysis

The average point density per class measured using the *knn* algorithm with $k = 5$ is presented in Table 1. The maximum difference between classes with the highest and the lowest point density is 2 cm. The average of all densities was calculated to be 0.0256 m—this value was taken as a starting point to apply density reduction and it was increased on a pseudo-logarithmic scale until 1 m was reached. Examples of the generated images with point density reduction are shown in Figure 5.

Table 1. Average point density of input data.

Class	Average Point-to-Point Distance (Meters)
Tree	0.0335
Bench	0.0219
Car	0.0292
Was. cont.	0.0169
Streetlight	0.0229
Motorbike	0.0359
Was. basket	0.0168
Pedestrian	0.0299
Traffic light	0.0253
Traffic sign	0.0235
Average	0.0256

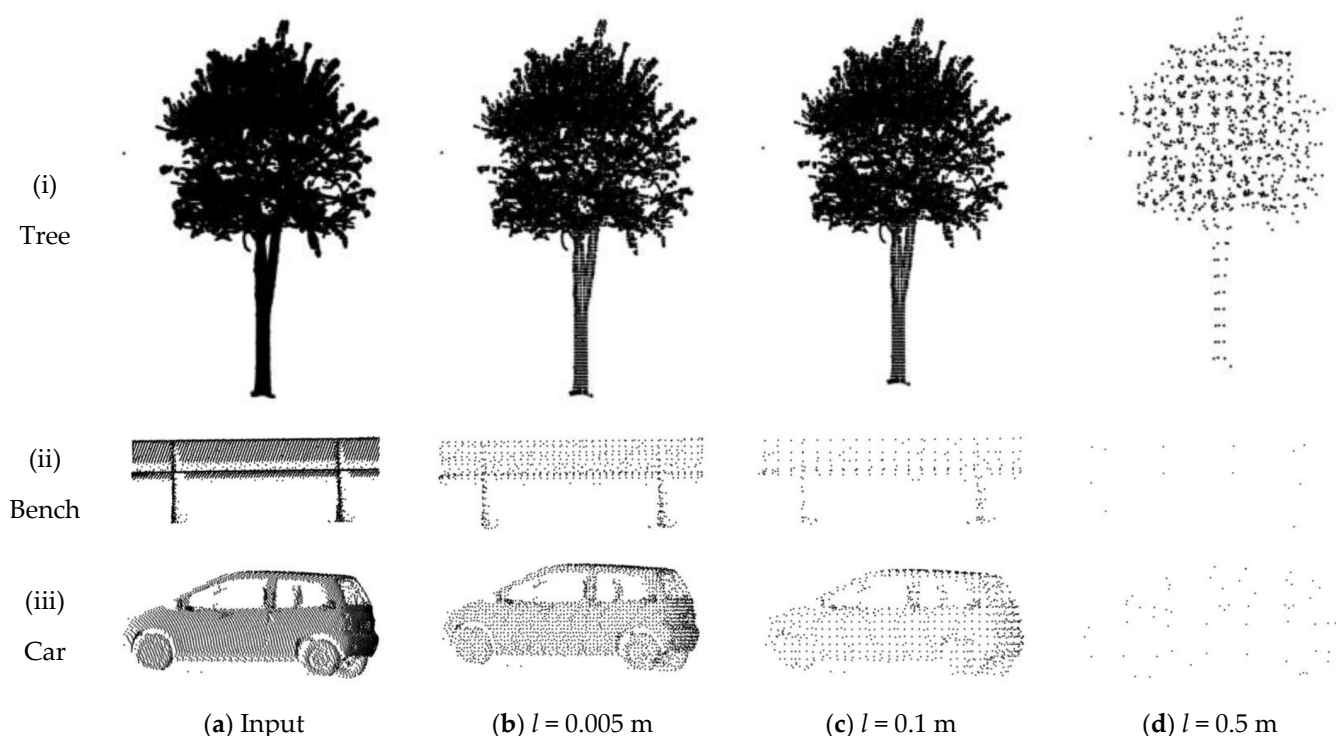
**Figure 5.** Images generated from the point cloud at different l values.

Table 2 shows the precision and Table 3 shows the recall of the classification according to the density value l . Table 4 compiles three confusion matrices along with the l variation. The reduction in the average precision and recall was constant up to $l = 0.1$ m. Above $l = 0.25$ m, practically no class was well identified. The misclassification behaviour was observed in two ways: an abrupt decline from very high accuracy to almost zero accuracy, as in the case of cars and traffic signs; and a gradual decline as traffic signs and streetlights showed for several consecutive l values a medium precision before being completely misclassified.

Table 2. Precision of the classification with point density reduction.

Class	Point Density (Distance / Point-to-Point in Meters)									
	0.03	0.04	0.05	0.065	0.085	0.1	0.25	0.5	0.75	1
Tree	1	1	1	1	0.9	0.9	0.6	0	0	0
Bench	1	1	1	1	1	1	1	0.5	0.8	0.8
Car	1	1	1	1	0.8	0.1	0	0	0	0
Was. cont.	1	0.9	0.5	0	0	0	0	0	0	0
Streetlight	1	1	1	0.8	0.7	0.7	0	0	0	0
Motorbike	1	0.9	0.7	0.1	0	0	0	0	0	0
Was. basket	1	1	1	1	1	0.9	1	0.1	0	0
Pedestrian	1	1	1	1	0.9	0.6	0.1	0	0	0
Traffic light	1	0.9	0.7	0.6	0.6	0.6	0.2	0	0	0
Traffic sign	1	1	1	0.9	1	0.8	0.2	0	0	0
Average	1	0.97	0.89	0.74	0.69	0.56	0.31	0.06	0.08	0.08

Table 3. Recall of the classification with point density reduction.

Class	Point Density (Distance / Point-to-Point in Meters)									
	0.03	0.04	0.05	0.065	0.085	0.1	0.25	0.5	0.75	1
Tree	1	1	1	1	1	1	1	0	0	0
Bench	1	0.9	0.9	0.5	0.4	0.4	0.4	0.2	0.2	0.2
Car	1	1	1	1	1	1	0	0	0	0
Was. cont.	1	1	1	0	0	0	0	0	0	0
Streetlight	1	1	1	1	1	1	0	0	0	0
Motorbike	1	1	1	1	0	0	0	0	0	0
Was. basket	1	0.9	0.6	0.5	0.6	0.3	0.2	0	0	0
Pedestrian	1	1	0.9	0.9	1	0.9	0.3	0	0	0
Traffic light	1	1	1	1	1	1	1	0	0	0
Traffic sign	1	0.9	0.8	0.6	0.6	0.5	0.1	0	0	0
Average	1	0.97	0.92	0.75	0.66	0.61	0.3	0.02	0.02	0.02

In general, it was observed that objects were affected to a greater or lesser extent by density reduction depending on the dimensions and shape of the object. The larger an object was, the less it was affected by density reduction, as the main features could still be seen. In particular, the volumetric point distribution in the tree canopy, produced by the multibeam return, resulted in the preservation of more points than in other classes. As can be seen in Table 4, when other classes such as cars and benches are difficult to distinguish with $l = 0.5$ m, trees remain completely identifiable.

The shape of the object is the most important factor for classification. The CNN looked at certain key points to identify each class. Simple shapes, such as squared waste containers, were very often confused with waste baskets. Additionally, other objects, such as motorbikes with many variations in shapes between models, were affected as the CNN did not find common features between models. Among the pole-like objects, streetlights and traffic lights showed similar behaviour, while traffic signs were more similar to pedestrians.

From the confusion matrix (Table 4) with $l = 0.065$ m, a well-defined diagonal was observed, except for the very early misclassification of the classes “Waste containers” and “Motorbikes”, while with $l = 0.1$ m, the class “Cars” had already joined the misclassified classes and “Pedestrian” and “Traffic lights” showed great confusion. From the confusion matrix with $l = 0.5$ m, the “Bench” and “Waste basket” classes were the ones towards which the confusion was directed. From these confusion results and the gradual precision loss of Table 2 and the reduction in the recall in Table 3, it can be deduced that in the absence of a class “Others”, the network considered waste baskets and benches as such a class.

Table 4. Confusion matrices for density evaluation with $l = 0.065$ m, $l = 0.1$ m, and $l = 0.5$ m.

Predicted ($l = 0.065$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	10	0	0	0	0	0	0	0	0	0
Bench	0	10	0	0	0	0	0	0	0	0
Car	0	0	10	0	0	0	0	0	0	0
Was. cont.	0	2	0	0	0	0	8	0	0	0
Streetlight	0	0	0	0	8	0	0	0	0	2
Motorbike	0	8	0	0	0	1	0	1	0	0
Was. basket	0	0	0	0	0	0	10	0	0	0
Pedestrian	0	0	0	0	0	0	0	10	0	0
Traffic light	0	0	0	0	0	0	0	0	6	4
Traffic sign	0	0	0	0	0	0	1	0	0	9

Predicted ($l = 0.1$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	9	0	0	0	0	0	0	0	0	1
Bench	0	10	0	0	0	0	0	0	0	0
Car	0	8	1	0	0	1	0	0	0	0
Was. cont.	0	0	0	0	0	0	10	0	0	0
Streetlight	0	0	0	0	7	0	0	0	0	3
Motorbike	0	10	0	0	0	0	0	0	0	0
Was. basket	0	0	0	0	0	0	9	1	0	0
Pedestrian	0	0	0	0	0	0	4	6	0	0
Traffic light	0	0	0	0	0	0	1	0	6	3
Traffic sign	0	0	0	0	0	0	2	0	0	8

Predicted ($l = 0.5$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	0	5	0	0	0	1	0	3	0	1
Bench	0	5	0	0	0	0	5	0	0	0
Car	0	10	0	0	0	0	0	0	0	0
Was. cont.	0	1	0	0	0	0	9	0	0	0
Streetlight	0	0	0	0	0	0	2	0	0	8
Motorbike	3	4	0	0	0	0	3	0	0	0
Was. basket	7	2	0	0	0	0	1	0	0	0
Pedestrian	0	0	0	0	0	0	10	0	0	0
Traffic light	0	0	0	0	0	0	5	4	0	1
Traffic sign	0	0	0	0	0	0	10	0	0	0

3.3. Noise Results and Analysis

The generation of random noise points was directly related to the volume of the bounding box that encloses the point cloud object. The objects with the largest volume were trees, streetlights, and traffic lights. The largest objects required a larger number of random points (Table 5). The noise generation with the same density affected the image generation due to the depth of the objects. For the same noise level, some classes could be identified by a human observer, while in other classes, the projection of points generated a practically black image (Figure 6). The selection of the noise values was limited between $d = 0.6$ m and $d = 0.02$ m in a linear distribution, considering no significant changes occurred beyond these two limits.

Table 5. Average volume per class object.

Class	Average Bounding Box Volume (Meters ³)
Tree	453.00
Bench	2.38
Car	17.88
Was. cont.	5.98
Streetlight	101.03
Motorbike	3.14
Was. basket	0.15
Pedestrian	0.84
Traffic light	58.28
Traffic sign	1.55

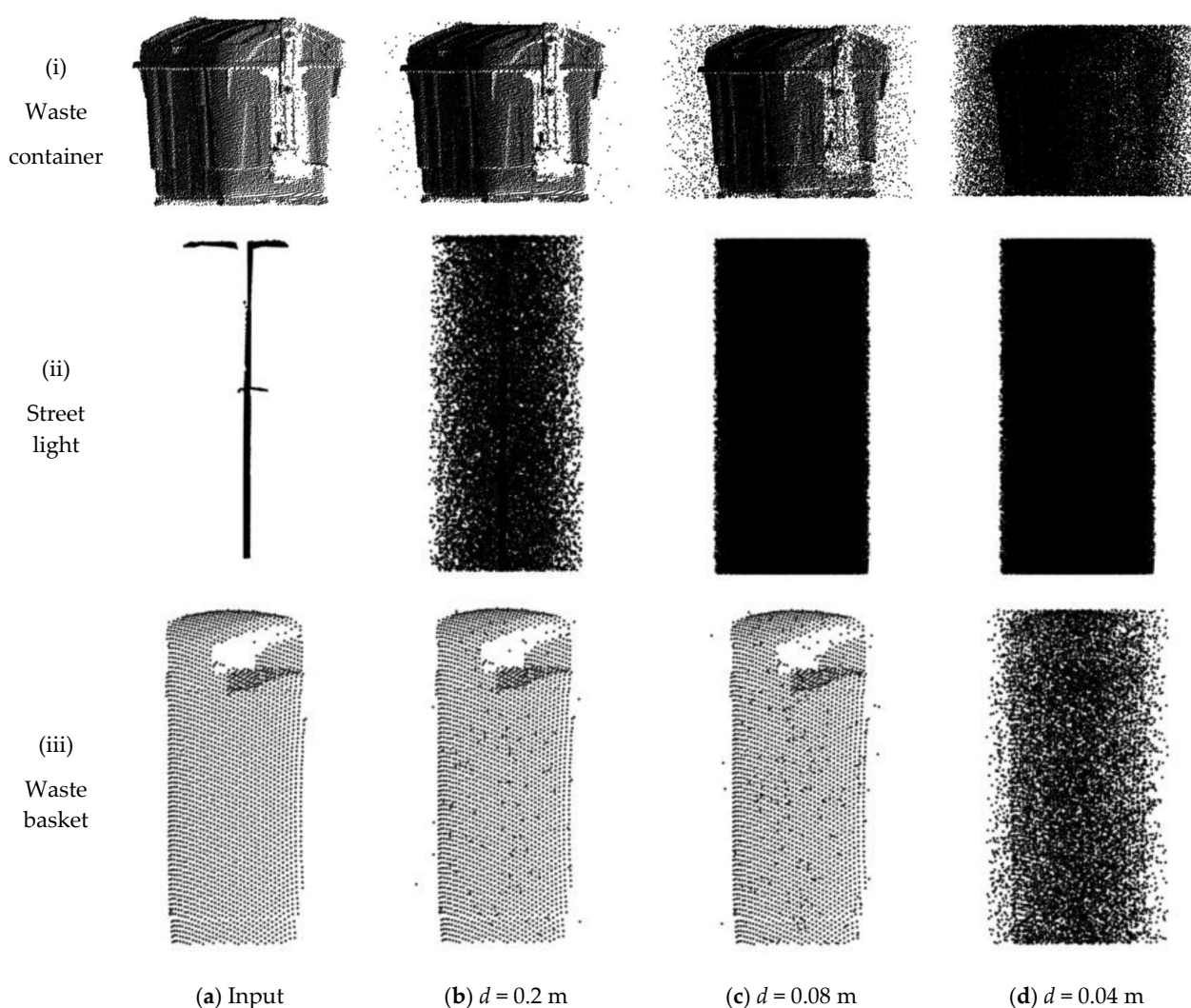
**Figure 6.** Images generated from point cloud objects at different levels of noise.

Table 6 shows the precision and Table 7 shows the recall of the classification results according to ten noise levels, while Table 8 compiles confusion matrices obtained from three noise levels. The precision loss was gradual until a very high misclassification with a noise density of $d = 0.04$ m was reached. Comparing Tables 6 and 8, there is a direct relation between misclassification and volume. The objects with the highest volume (tree light and traffic light) were the first to be confused. They lost the shape of the object due to a large number of projected noise points. Without considering large objects, no specific behaviour

of each class in relation to their shape was observed. The largest drop in precision occurred between $d = 0.2$ m and $d = 0.04$ m. The “Motorbikes” and “Pedestrian” classes, despite losing precision, maintained high recall values until they were misclassified.

Table 6. Precision of classification with different levels of noise.

Class	Noise Point Density d (Distance Point-to-Point in Meters)									
	0.6	0.5	0.4	0.3	0.2	0.1	0.08	0.06	0.04	0.002
Tree	0.8	0.7	0.6	0.2	0	0	0	0	0	0
Bench	1	1	1	1	1	1	1	1	0.7	0
Car	1	1	1	1	1	0.6	0.6	0	0	0
Was. cont.	1	1	1	1	0.9	0.7	0.7	0.3	0	0
Streetlight	1	0.6	0.5	0.2	0.1	0	0	0	0	0
Motorbike	1	0.9	1	0.8	0.7	1	0.5	0.2	0	0
Was. basket	1	1	1	1	1	1	1	1	1	1
Pedestrian	1	1	1	1	0.9	0.7	0.8	0.2	0	0
Traffic light	0.9	0.8	0.9	0.9	0.5	0.3	0.2	0	0	0.2
Traffic sign	1	1	0.9	0.9	1	0.6	0.3	0.3	0	0.5
Average	0.97	0.90	0.89	0.80	0.71	0.59	0.46	0.30	0.17	0.17

Table 7. Recall of classification with different levels of noise.

Class	Noise Point Density d (Distance Point-to-Point in Meters)									
	0.6	0.5	0.4	0.3	0.2	0.1	0.08	0.06	0.04	0.002
Tree	1	1	1	1	0	0	0	0	0	0
Bench	1	1	1	0.8	0.7	0.8	0.7	0.6	1	0
Car	1	1	1	1	1	1	1	0	0	0
Was. cont.	1	1	1	1	1	1	0.9	0.6	0	0
Streetlight	0.8	0.8	0.6	0.3	1	0	0	0	0	0
Motorbike	0.9	1	1	1	1	1	1	1	0	0
Was. basket	1	0.7	0.6	0.6	0.3	0.2	0.2	0.2	0.1	0.2
Pedestrian	1	0.9	1	1	1	1	1	1	0	0
Traffic light	1	1	1	1	1	0.6	0.3	0	0	0.3
Traffic sign	1	0.9	0.8	0.6	0.7	0.6	0.5	0.3	0	0.1
Average	0.97	0.93	0.9	0.83	0.77	0.62	0.56	0.37	0.11	0.06

From the confusion matrices (Table 8), the confusion was gradual, starting with trees and streetlights and predicting all objects erroneously as waste baskets. The network had considered the class “Waste basket” as the class “Others”, as indicated by high precision values and low recall.

3.4. Occlusions Results and Analysis

The analysis of the influence of occlusions on the classification focused on the different origins and sizes of occlusions. Therefore, five possible origins for each object and ten different occlusion sizes per origin were selected. In total, 5000 images were generated in the study of the occlusions. The occlusion origins for an object of each class are shown in Figure 7. The k-mean estimated origins matched between the different objects in each class. An occlusion radius range between $r = 0.1$ m and $r = 5$ m was selected to analyse both small-size occlusions and those with a size potentially affecting larger objects. Large occlusions applicable to small objects imply the disappearance of these objects. An example of the generation of occlusions of different sizes is shown in Figure 8.

Table 8. Confusion matrices for noise evaluation with $d = 0.3$ m, $d = 0.1$ m, and $d = 0.06$ m.

Predicted ($d = 0.3$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	2	1	0	0	3	0	4	0	0	0
Bench	0	10	0	0	0	0	0	0	0	0
Car	0	0	10	0	0	0	0	0	0	0
Was. cont.	0	0	0	10	0	0	0	0	0	0
Streetlight	0	0	0	0	2	0	3	0	0	5
Motorbike	0	2	0	0	0	8	0	0	0	0
Was. basket	0	0	0	0	0	0	10	0	0	0
Pedestrian	0	0	0	0	0	0	0	10	0	0
Traffic light	0	0	0	0	0	0	1	0	9	0
Traffic sign	0	0	0	0	1	0	0	0	0	9
Predicted ($d = 0.1$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	0	0	0	0	0	0	10	0	0	0
Bench	0	10	0	0	0	0	0	0	0	0
Car	0	0	6	0	0	0	4	0	0	0
Was. cont.	0	0	0	7	0	0	2	0	0	1
Streetlight	0	0	0	0	0	0	6	0	1	3
Motorbike	0	0	0	0	0	10	0	0	0	0
Was. basket	0	0	0	0	0	0	10	0	0	0
Pedestrian	0	2	0	0	0	0	1	7	0	0
Traffic light	0	0	0	0	0	0	7	0	3	0
Traffic sign	0	0	0	0	0	0	3	0	1	6
Predicted ($d = 0.06$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	0	0	0	0	0	0	10	0	0	0
Bench	0	10	0	0	0	0	0	0	0	0
Car	0	2	0	0	0	0	3	0	0	5
Was. cont.	0	2	0	3	0	0	5	0	0	0
Streetlight	0	0	0	1	0	0	3	0	4	2
Motorbike	0	2	0	0	0	2	6	0	0	0
Was. basket	0	0	0	0	0	0	10	0	0	0
Pedestrian	0	0	1	0	0	0	7	2	0	0
Traffic light	0	0	0	0	0	0	10	0	0	0
Traffic sign	0	0	0	1	0	0	6	0	0	3

Table 9 shows the precision and Table 10 shows the recall of the classification with different occlusion sizes. In Table 11, three confusion matrices are compiled. The incidence of occlusions varied mainly according to the size of the object, since if the object is large, a large occlusion is needed to remove a relevant part of the points. The location of the occlusion was another important factor, as in certain objects it can remove key features that characterise the object. This is the case of trees, streetlights, some traffic lights, or signs, where if the occlusion is at the top and is large, the image was simply a vertical line. In some traffic lights, no misclassification was observed since they had a double light (top and middle) (Figure 8iii) and the elimination of both only happened in extreme cases. The “Streetlight” class showed very high robustness, with high precision and recall rates even in extreme cases. Other classes, such as trees, also showed good precision, although in the worst case, the CNN tended to over detect them.

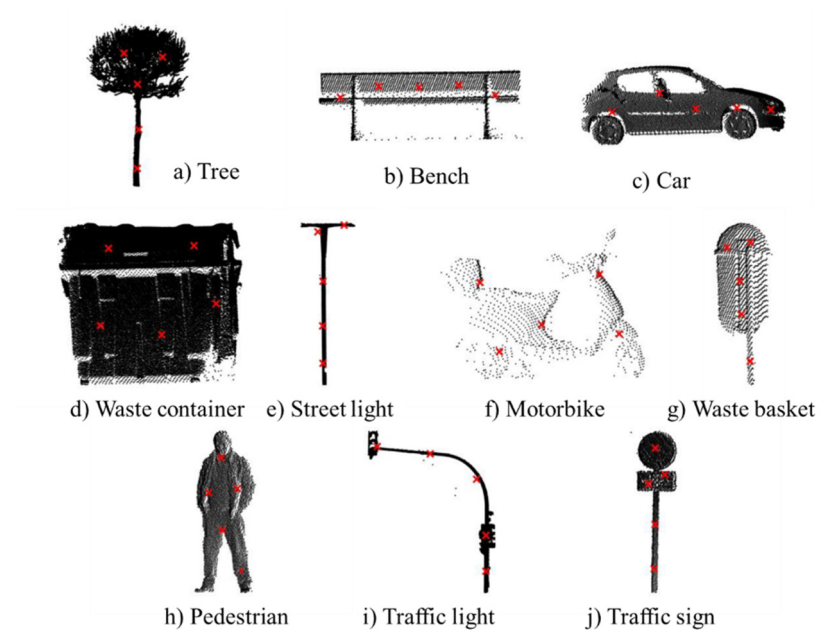


Figure 7. Distribution of occlusion origins (red points) in point cloud objects.

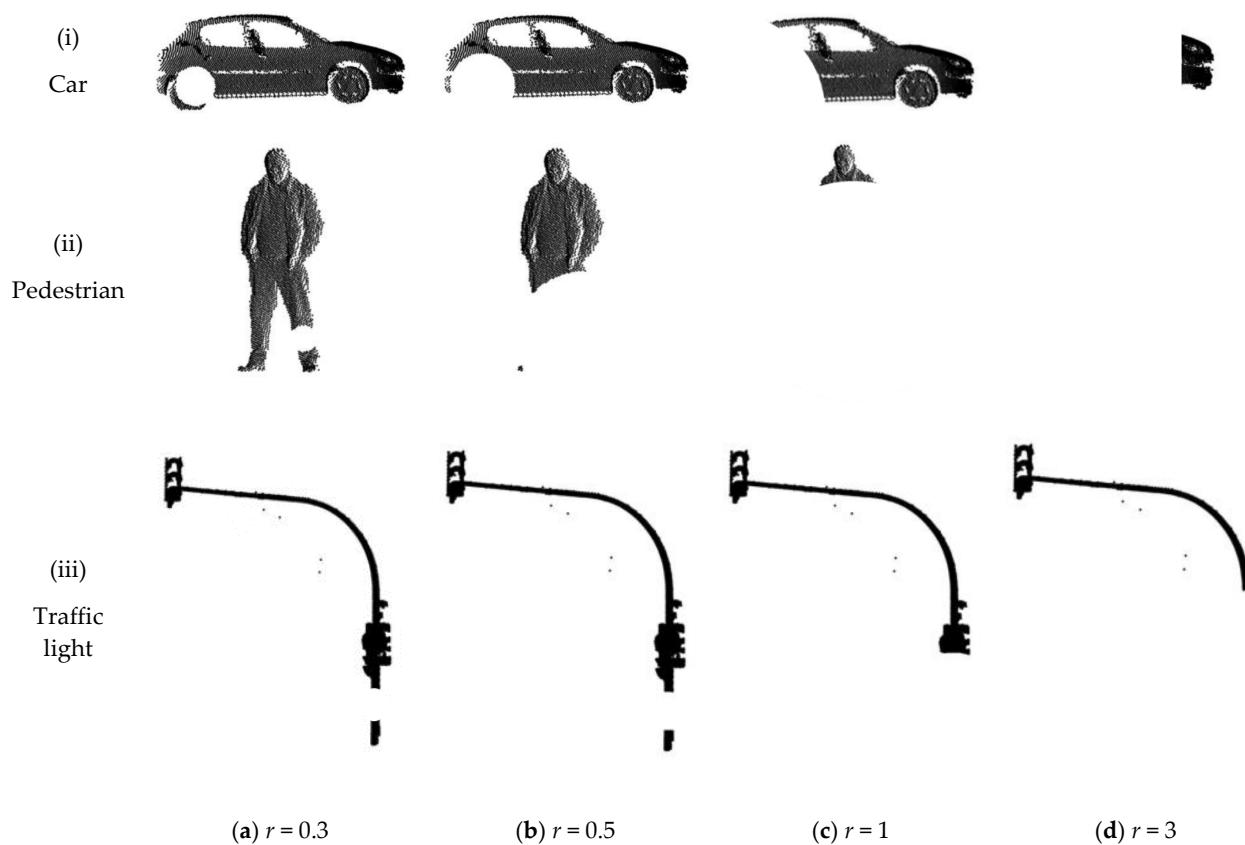


Figure 8. Images generated with occlusions of different sizes.

Table 9. Precision of classification with different occlusion sizes.

Class	Occlusion Size Radius r (in Meters)									
	0.1	0.2	0.3	0.5	0.75	1	1.5	2	3	5
Tree	1	1	1	1	1	1	0.9	0.9	0.6	0.3
Bench	1	1	1	1	1	0.8	0.2	0	0	0
Car	1	1	1	1	0.9	0.8	0.7	0.5	0	0
Was. cont.	1	1	1	1	0.9	0.4	0	0	0	0
Streetlight	1	1	1	1	1	0.9	0.8	0.9	0.7	0.7
Motorbike	1	1	0.9	0.6	0.3	0.2	0	0	0	0
Was. basket	1	0.9	0.7	0.2	0.5	0.9	1	1	1	1
Pedestrian	1	0.8	0.9	0.9	0.5	0.3	0.1	0	0	0
Traffic light	1	1	1	1	0.9	0.9	0.8	0.6	0.4	0.3
Traffic sign	0.9	0.8	0.8	0.7	0.7	0.7	0.5	0.3	0	0
Average	0.99	0.95	0.93	0.83	0.76	0.69	0.51	0.42	0.28	0.22

Table 10. Recall of classification with different occlusion sizes.

Class	Occlusion Size Radius r (in Meters)									
	0.1	0.2	0.3	0.5	0.75	1	1.5	2	3	5
Tree	1	1	1	0.9	0.8	0.8	0.7	0.8	0.8	0.9
Bench	1	1	0.8	0.6	0.7	0.5	0.4	0	0	0
Car	1	1	1	1	1	0.9	1	1	1	0
Was. cont.	1	1	1	1	1	1	0	0	0	0
Streetlight	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.8	0.7	0.7
Motorbike	1	1	1	1	0.7	0.4	0	0	0	0
Was. basket	1	1	0.9	0.8	0.7	0.7	0.3	0.2	0.1	0.1
Pedestrian	1	1	0.9	0.7	0.4	0.3	0.2	0	0	0
Traffic light	1	0.9	0.9	0.9	0.8	0.8	0.6	0.6	0.5	0.5
Traffic sign	1	1	1	0.9	0.7	0.8	0.6	0.9	0.3	0
Average	0.99	0.98	0.94	0.86	0.76	0.7	0.45	0.43	0.34	0.22

The class “Bench”, being small-sized objects, was affected from occlusions of size $r = 1$ m. Although partially occluded benches were classified correctly, occlusions larger than $r = 1.5$ m eliminated the benches completely. The class “Car” showed similar behaviour. Additionally, the occlusions which caused the most errors were the central ones that broke the car in two parts, or the occlusions located on the wheels. The class “Waste containers” was not affected by the position of the occlusions since the origins were centred in the object. Similarly, the motorbike class showed the same behaviour, as there was no area where the occlusion generated the most error.

The waste baskets behaved in two ways depending on the occlusion size. Small-sized occlusions were mistaken for benches. The class “Bench” already showed some behaviour as class “Others” in the point density analysis. As the size of the occlusions increased, the confusions between classes were dispersed, and with large occlusions when many objects were completely occluded, the class “Waste basket” acted as a class “Others”, showing high precision and low recall.

Pedestrians began to be misclassified as having occlusions with more than $r = 0.5$ m located on the torso.

The choice of five points was a compromise solution to cover several parts along the entire length of the object. Many more points could have been chosen; however, given the low effect of small occlusions on the classification, there was no need to increase the number of occlusion origins. On the other hand, the large occlusions covered several origins, so it was not necessary to increase the number of occlusion origins for this reason either.

Table 11. Confusion matrices for occlusion evaluation with $r = 0.5$ m, $r = 1.5$ m, and $r = 3$ m.

Predicted ($r = 0.5$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	50	0	0	0	0	0	0	0	0	0
Bench	0	49	0	0	0	0	0	0	0	1
Car	0	0	49	0	0	1	0	0	0	0
Was. cont.	0	0	0	50	0	0	0	0	0	0
Streetlight	0	0	0	0	48	0	0	0	2	0
Motorbike	0	7	0	0	0	32	0	11	0	0
Was. basket	4	25	1	0	0	0	9	6	0	5
Pedestrian	2	2	0	1	0	0	2	43	0	0
Traffic light	0	1	0	0	1	0	0	0	48	0
Traffic sign	0	0	0	0	9	0	0	0	5	36
Predicted ($r = 1.5$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	46	0	0	0	1	1	0	0	2	0
Bench	2	10	0	1	0	0	21	0	0	16
Car	0	1	35	0	0	8	0	5	1	0
Was. cont.	7	9	0	0	6	1	4	13	9	1
Streetlight	0	0	0	0	44	0	0	0	6	0
Motorbike	3	3	0	0	0	0	29	14	0	1
Was. basket	0	0	0	0	0	0	50	0	0	0
Pedestrian	2	0	0	0	0	0	42	6	0	0
Traffic light	0	0	0	0	6	1	0	1	39	3
Traffic sign	4	1	0	0	5	0	3	1	9	27
Predicted ($r = 3$ m)										
Reference	Tree	Bench	Car	Was. contain.	Streetlight	Motorbike	Was. basket	Pedestrian	Traffic light	Traffic sign
Tree	30	1	0	0	6	2	3	4	4	0
Bench	0	0	0	0	0	0	50	0	0	0
Car	6	2	1	0	1	4	18	10	7	1
Was. cont.	0	0	0	0	0	0	50	0	0	0
Streetlight	0	0	0	0	35	0	2	1	12	0
Motorbike	0	0	0	0	0	0	50	0	0	0
Was. basket	0	0	0	0	0	0	50	0	0	0
Pedestrian	0	0	0	0	0	0	50	0	0	0
Traffic light	0	0	0	0	5	2	20	0	21	2
Traffic sign	1	0	0	0	1	0	44	2	1	1

3.5. Re-Train of CNN

The re-training aims to assess whether adding data with disturbances in the network training can minimise misclassification. A second dataset not previously used in the analysis of results was used to re-train the network (Figure 9). In this second dataset, composed of the same number and classes of objects as the previous dataset, disturbances were generated, added to the objects, and transformed into images. From all the images generated, 10 images with point density variations, 10 images with noise and 30 images with occlusions were randomly selected per class. More occlusion samples were selected due to the variation in occlusion positioning and size, which generated a larger amount of data than the other two disturbances. The selected images were added to the training set that was used to pre-train Inception V3 for the previous classification. The hyperparameters of the training were the same as in the pre-training (optimisation method *sgdm*, learning rate 10–4, Momentum 0.9, L2 Regularisation 10–4, Max Epochs 20, and Mini Batch Size

16). The network was trained in 200 min with a GPU NVIDIA GTX1050 4 GB GDDR5, CPU i7-7700HQ 2.8 Ghz, and 16 GB RAM DDR4.

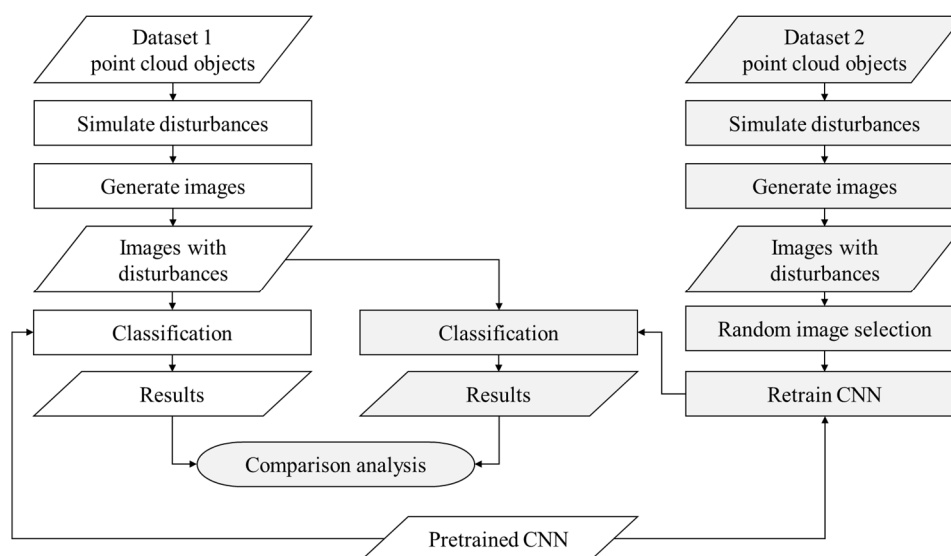


Figure 9. Workflow of the data comparison with the re-trained CNN.

The re-training showed a great improvement in the classification of the point density tests (Figure 10) in the low and central values l . Specifically, up to $l = 0.1$ m, the re-training maintained a high precision and improved the initial results by up to 40% and slowed down the appearance of misclassification effects caused by point density. However, above $l = 0.1$ m, although with much higher precision than the original, the classification results continue to have very high misclassification rates. At the extremes of the l values, where the image showed no change (both the undisturbed point cloud and unrecognisable), the same precision as in the original results was observed.

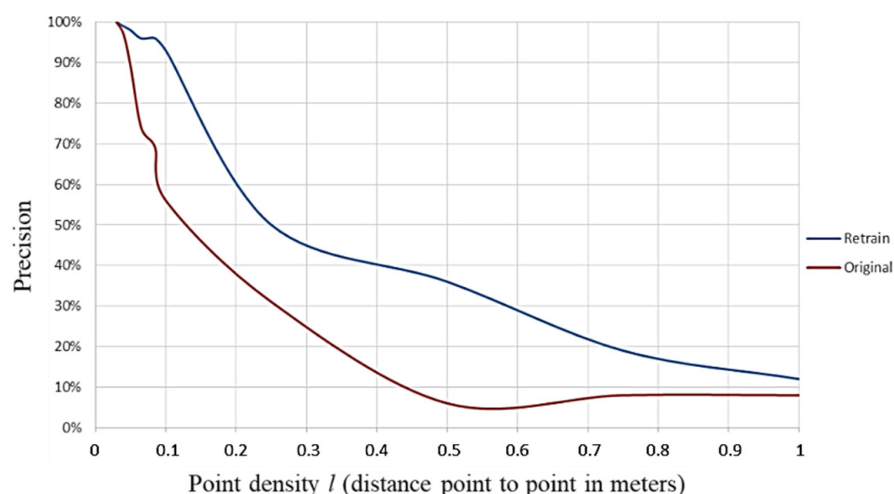


Figure 10. Comparison of precision of the re-trained and original CNN with point density variation.

The use of images with noise points in re-training produced a consistent improvement of 10–20% for all d values (Figure 11). However, in cases with a high level of noise, the CNN was still not able to correctly classify the objects.

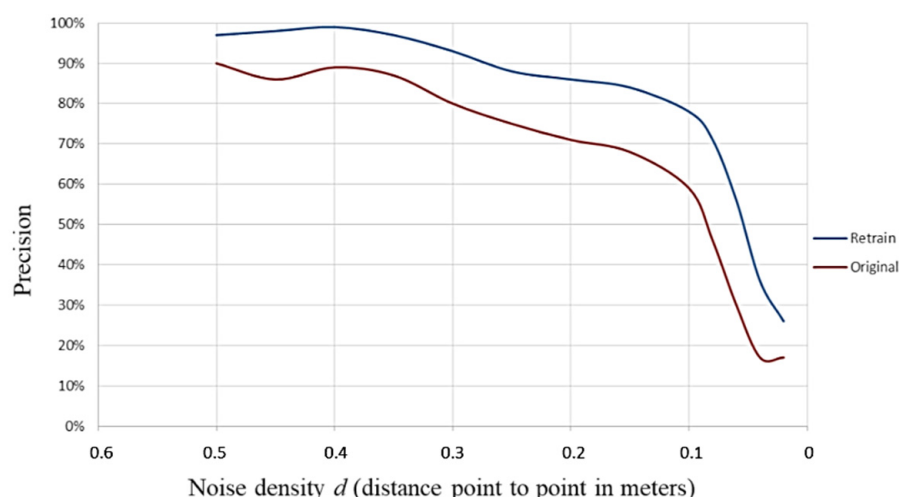


Figure 11. Comparison of precision of the re-trained and original CNN with noise level variation.

Re-training with occlusions showed a turning point in the results obtained (Figure 12). Up to $r = 1$ m, the classification with small occlusions improved up to 10% and the misclassification effect was slowed down. However, from $r = 1$ m onwards, the reverse effect occurred, and the results worsened. This turning point is related to the size of the occlusion concerning the size of the object, as many small objects with occlusions of $r = 1$ m to 1.5 m started to occlude almost entirely. Applying occlusions of radius greater than $r = 1.5$ m, most of the objects that conserved information were the large ones, mainly pole-like objects. When these objects were partially occluded, the resulting sample was usually only a pole, which was very difficult to identify.

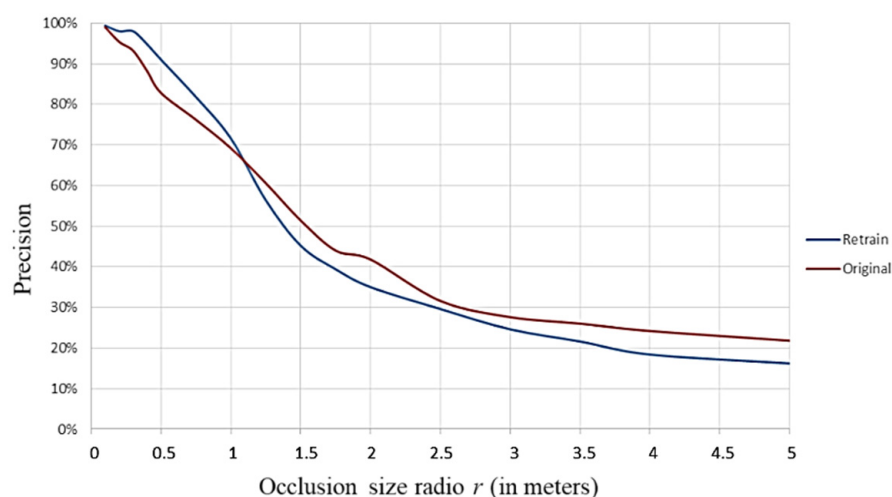


Figure 12. Comparison of precision of the re-trained and original CNN with occlusion size variation.

4. Discussion

The objects were extracted from the urban point cloud manually. Although disturbances may affect detection and individualisation, only the classification of urban objects was evaluated in this work. Effects such as extreme density reduction, or total occlusion, would logically discard the extraction of samples from the environment as objects would not be recognisable by the extracting algorithm. In the opposite case, the existence of extreme noise would not allow clustering algorithms such as DBSCAN or connected components correctly, and occlusions can break the object into several parts producing over segmentation.

The selection of values for the synthetic generation of the disturbances and their distribution was based on several considerations. The values were delimited between those that produced almost no effect on the classification and those that rendered the samples completely useless. Between the two limits, the range of values was selected in such a way as to provide constant information at classification intervals and the range was reduced where changes were more abrupt. The distributions were in linear or pseudo-logarithmic intervals. In addition, the maximum, minimum and intervals had to apply to objects of very different dimensions, ranging from waste baskets of just 1 m to trees of more than 5 m in height.

The process of generating synthetic disturbances was automated to produce all samples easily, reliably, and with the ability to be replicable. As far as possible, the design of the algorithm corresponded to the disturbances produced in the real world, although they were taken to extreme situations to seek to break the classification algorithm. As mentioned above, very distant objects to MLS (with low point density) would not be extractable from the urban point cloud or would be mistaken for noise. On the other hand, in conditions of extreme environmental noise (such as that generated by torrential rain or large amounts of dust-sand in suspension), MLS acquisitions are not scheduled or executed. Occlusions, although generated in various parts of the objects, are typically only located in low areas for the tall objects [35], where it was found to have little effect on the misclassification of pole-like objects, trees, or even pedestrians. Completely occluded objects were also not extracted from the urban point cloud for further classification.

Re-training with samples containing synthetic errors provides robustness to the CNN classifier. The classification of objects with low point density, noise, and small occlusions was improved; however, re-training did not solve the most serious problems. From these results, it is recommended to use some samples with synthesised disturbances in the training to make the classifier less sensitive to point density variations, ambient noise, and small occlusions.

5. Conclusions

In this work, three disturbances (point density, noise, and occlusions) were evaluated as affecting a CNN to classify the main classes of urban objects acquired with MLS. Synthetic disturbances with different levels of intensity were generated and applied to point clouds of urban objects acquired the cities of Vigo and Porto. The CNN was also re-trained with some of the new synthetic samples to assess whether the effects of the disturbances could be mitigated.

The main disturbances affecting the classification were density reduction and large occlusions. Reducing the density to a distance between points of 0.1 m, the precision was halved (0.56), and with a distance of 0.25 m, few samples were classified well. Noise showed a behaviour linked to the volume of the object and did not significantly affect the classification until very extreme levels that would never occur in a real acquisition. Thus, at the same noise level, bulky elements such as streetlights were unrecognisable, while smaller ones, such as waste baskets, were clearly visible. In samples with occlusion of radius 1 m, the precision fell to 0.69, while with 1.5 m the precision stood at 0.51. Small occlusions (less than 1 m of radius) had little effect in misclassification, while large occlusions showed problems when small objects were completely hidden or when occlusion was located on top of tall objects. However, if occlusions obscure an entire object, it would no longer be removable from the environment, and occlusions in high areas of objects do not usually occur in the urban MLS acquisition. With occlusion of radius 1 m, the precision fell to 0.69, while with 1.5 m the precision stood at 0.51.

Re-training the CNN with synthetic disturbance data provided some robustness to the classifier. The classification of samples with low density was improved up to 40%, ambient noise up to 20%, and occlusions up to 1 m radius up to 10%. In extreme cases of disturbances, there was not enough improvement to obtain a good classification. At

the most extreme level of disturbances, none of the precisions exceeded 30%, even after network re-training.

In future work, other classification methods will be tested, and the simultaneity of the disturbances will be compared to see which disturbances are affected to a greater degree.

Author Contributions: Conceptualization, J.B.; methodology, J.B. and A.M.-R.; software, A.M.-R.; validation, J.B., P.A. and A.M.-R.; resources, P.A. and H.L.; data curation, A.M.-R.; writing—original draft preparation, J.B.; writing—review and editing, P.A.; supervision, P.A. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Xunta de Galicia given through human resources grant (ED481B-2019-061), and competitive reference groups (ED431C 2016-038), the Ministerio de Ciencia, Innovación y Universidades -Gobierno de España- (RTI2018-095893-B-C21, PID2019-105221RB-C43/AEI/10.13039/501100011033). This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769255. This document reflects only the views of the author(s). Neither the Innovation and Networks Executive Agency (INEA) nor the European Commission is in any way responsible for any use that may be made of the information it contains. The statements made herein are solely the responsibility of the authors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data can be requested by contacting the corresponding author.

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

References

- Remondino, F. From point cloud to surface: The modeling and visualization problem. *International Archives of the Photogrammetry. Remote Sens. Spat. Inf. Sci.* **2003**, *34*.
- Liao, Y.; Wood, R.L. Discrete and Distributed Error Assessment of UAS-SfM Point Clouds of Roadways. *Infrastructures* **2020**, *5*, 87. [\[CrossRef\]](#)
- Khanal, M.; Hasan, M.; Sterbentz, N.; Johnson, R.; Weatherly, J. Accuracy Comparison of Aerial Lidar, Mobile-Terrestrial Lidar, and UAV Photogrammetric Capture Data Elevations over Different Terrain Types. *Infrastructures* **2020**, *5*, 65. [\[CrossRef\]](#)
- Huang, D.; Du, S.; Li, G.; Zhao, C.; Deng, Y. Detection and monitoring of defects on three-dimensional curved surfaces based on high-density point cloud data. *Precis. Eng.* **2018**, *53*, 79–95. [\[CrossRef\]](#)
- Li, Y.; Wu, B.; Ge, X. Structural segmentation and classification of mobile laser scanning point clouds with large variations in point density. *Isprs J. Photogramm. Remote Sens.* **2019**, *153*, 151–165. [\[CrossRef\]](#)
- Soilán, M.; Riveiro, B.; Martínez-Sánchez, J.; Arias, P. Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory. *Isprs J. Photogramm. Remote Sens.* **2016**, *114*, 92–101. [\[CrossRef\]](#)
- Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *Isprs J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [\[CrossRef\]](#)
- Poux, F.; Billen, R. Voxel-based 3D Point Cloud Semantic Segmentation: Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods. *Isprs Int. J. Geo-Inf.* **2019**, *8*, 213. [\[CrossRef\]](#)
- Li, X.; Wang, L.; Wang, M.; Wen, C.; Fang, Y. DANCE-NET: Density-aware convolution networks with context encoding for airborne LiDAR point cloud classification. *Isprs J. Photogramm. Remote Sens.* **2020**, *166*, 128–139. [\[CrossRef\]](#)
- Lin, C.-H.; Chen, J.-Y.; Su, P.-L.; Chen, C.-H. Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification. *Isprs J. Photogramm. Remote Sens.* **2014**, *94*, 70–79. [\[CrossRef\]](#)
- Soilán, M.; Sánchez-Rodríguez, A.; Del Río-Barral, P.; Perez-Collazo, C.; Arias, P.; Riveiro, B. Review of Laser Scanning Technologies and Their Applications for Road and Railway Infrastructure Monitoring. *Infrastructures* **2019**, *4*, 58. [\[CrossRef\]](#)
- Wang, J.; Xu, K.; Liu, L.; Cao, J.; Liu, S.; Yu, Z.; Gu, X.D. Consolidation of Low-quality Point Clouds from Outdoor Scenes. *Comput. Graph. Forum* **2013**, *32*, 207–216. [\[CrossRef\]](#)
- Grilli, E.; Menna, F.; Remondino, F. A review of point clouds segmentation and classification algorithms. *Isprs Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-2/W3*, 339–344. [\[CrossRef\]](#)
- Castillo, E.; Liang, J.; Zhao, H. *Point Cloud Segmentation and Denoising via Constrained Nonlinear Least Squares Normal Estimates BT Innovations for Shape Analysis: Models and Algorithms*; Breuß, M., Bruckstein, A., Maragos, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 283–299. ISBN 978-3-642-34141-0.
- Rakotosaona, M.; La Barbera, V.; Guerrero, P.; Mitra, N.J.; Ovsjanikov, M. PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds. *Comput. Graph. Forum* **2020**, *39*, 185–203. [\[CrossRef\]](#)

16. Rivero, J.R.V.; Gerbich, T.; Teiluf, V.; Buschardt, B.; Chen, J. Weather Classification Using an Automotive LIDAR Sensor Based on Detections on Asphalt and Atmosphere. *Sensors* **2020**, *20*, 4306. [[CrossRef](#)] [[PubMed](#)]
17. Shamsudin, A.U.; Ohno, K.; Westfechtel, T.; Takahiro, S.; Okada, Y.; Tadokoro, S. Fog removal using laser beam penetration, laser intensity, and geometrical features for 3D measurements in fog-filled room. *Adv. Robot.* **2016**, *30*, 729–743. [[CrossRef](#)]
18. Stanislas, L.; Nubert, J.; Dugas, D.; Nitsch, J.; Sünderhauf, N.; Siegwart, R.; Cadena, C.; Peynot, T. Airborne Particle Classification in LiDAR Point Clouds Using Deep Learning. In *Proceedings of the Experimental Robotics*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2021; pp. 395–410.
19. Chen, D.; Zhang, L.; Mathiopoulos, P.T.; Huang, X. A Methodology for Automated Segmentation and Reconstruction of Urban 3-D Buildings from ALS Point Clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4199–4217. [[CrossRef](#)]
20. Barazzetti, L.; Previtali, M.; Scaioni, M. Roads Detection and Parametrization in Integrated BIM-GIS Using LiDAR. *Infrastructures* **2020**, *5*, 55. [[CrossRef](#)]
21. Kwon, S.K.; Hyun, E.; Lee, J.-H.; Lee, J.; Son, S.H. A Low-Complexity Scheme for Partially Occluded Pedestrian Detection Using LIDAR-RADAR Sensor Fusion. In *Proceedings of the 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Daegu, Korea, 17–19 August 2016; p. 104.
22. Zachmann, I.; Scarnati, T. A comparison of template matching and deep learning for classification of occluded targets in LiDAR data. In *Automatic Target Recognition XXX*; SPIE-International Society for Optics and Photonics: Bellingham, WA, USA, 2020; Volume 11394, p. 113940H.
23. Habib, A.F.; Chang, Y.-C.; Lee, D.C. Occlusion-based Methodology for the Classification of Lidar Data. *Photogramm. Eng. Remote Sens.* **2009**, *75*, 703–712. [[CrossRef](#)]
24. Zhang, X.; Fu, H.; Dai, B. Lidar-Based Object Classification with Explicit Occlusion Modeling. In *Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, 24–25 August 2019; Volume 2, pp. 298–303.
25. Griffiths, D.; Boehm, J. A Review on Deep Learning Techniques for 3D Sensed Data Classification. *Remote Sens.* **2019**, *11*, 1499. [[CrossRef](#)]
26. Balado, J.; Sousa, R.; Díaz-Vilariño, L.; Arias, P. Transfer Learning in urban object classification: Online images to recognize point clouds. *Autom. Constr.* **2020**, *111*, 103058. [[CrossRef](#)]
27. Gézero, L.; Antunes, C. Automated Three-Dimensional Linear Elements Extraction from Mobile LiDAR Point Clouds in Railway Environments. *Infrastructures* **2019**, *4*, 46. [[CrossRef](#)]
28. Cabo, C.; Ordoñez, C.; García-Cortés, S.; Martínez-Sánchez, J. An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. *Isprs J. Photogramm. Remote Sens.* **2014**, *87*, 47–56. [[CrossRef](#)]
29. Poux, F. *How to Automate LiDAR Point Cloud Sub-Sampling with Python*; Towards Data Science: Toronto, ON, USA, 2020.
30. Puente, I.; González-Jorge, H.; Martínez-Sánchez, J.; Arias, P. Review of mobile mapping and surveying technologies. *Measurement* **2013**, *46*, 2127–2145. [[CrossRef](#)]
31. Babahajiani, P.; Fan, L.; Gabbouj, M. *Object Recognition in 3D Point Cloud of Urban. Street Scene BT Computer Vision ACCV 2014 Workshops*; Jawahar, C.V., Shan, S., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 177–190.
32. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 4558–4567.
33. Zhang, J.; Zhao, X.; Chen, Z.; Lu, Z. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE Access* **2019**, *7*, 179118–179133. [[CrossRef](#)]
34. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [[CrossRef](#)]
35. Balado, J.; González, E.; Verbree, E.; Díaz-Vilariño, L.; Lorenzo, H. automatic detection and characterization of ground occlusions in urban point clouds from mobile laser scanning data. *Isprs Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *VI-4/W1-20*, 13–20. [[CrossRef](#)]