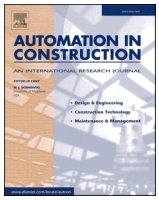
Automation in Construction 141 (2022) 104425

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Point cloud semantic segmentation of complex railway environments using deep learning

Javier Grandio [a,*], Belén Riveiro [a], Mario Soilán [b], Pedro Arias [a]

[a] Centro de Investigación en Tecnoloxías, Enerxía e Procesos Industriais (CINTECX), Applied Geotechnologies Research Group, Campus Universitario de Vigo, Universidade de Vigo, As Lagoas, Marcosende, 36310 Vigo, Spain
[b] Department of Cartographic and Terrain Engineering, University of Salamanca, Calle Hornos Caleros 50, 05003 Avila, Spain

ARTICLE INFO

Keywords:
Point clouds
Deep learning
Semantic segmentation
Railway infrastructure

ABSTRACT

Safety of transportation networks is of utmost importance for our society. With the emergency of digitalization, the railway sector is accelerating the automation in inventory and inspection procedures. Mobile mapping systems allow capturing three-dimensional point clouds of the infrastructure in short periods of time. In this paper, a deep learning methodology for semantic segmentation of railway infrastructures is presented. The methodology segments both linear and punctual elements from railway infrastructure, and it is tested in four scenarios: i) 90 km-long railway; ii) 2 km-long low-quality point clouds; iii) 400 m-long high-quality point clouds; iv) 1.4 km-long railway recoded with aerial mapping system. The longest one is used for training and testing, obtaining mean accuracy greater than 90%. The other scenarios are used only for testing, and qualitative results are discussed, proving that the method can be applied to new scenarios that significantly differ in terms of data quality and resolution.

1. Introduction

The railway infrastructure is crucial in modern society, used daily as a transportation method for both people and goods. In fact, the railway passenger transport in Europe shows an increasing trend over the years [1], accounting for the 7.8% of the total passenger transport in the European Union in 2017 [2]. Regarding freight transport, railway mode accounted for the 18.7% of the total in the European Union during 2018 [3]. In consequence, the well-functioning of the infrastructure has a high impact in the well-being of the population. In order to ensure the safeness of the infrastructure, it is necessary to perform a correct and regular maintenance. Many accidents occur as a result of unknown deterioration of the assets [4,5]. However, in many cases, those types of issues can be avoided by a correct maintenance of the infrastructure [6,7]. In the case of the railway infrastructure, the main drawback to perform a correct maintenance is its massive scale. For example, due to the extension of the European railway system, the European countries allocate in activities for inspection and maintenance 15–25 billion EUR annually [8]. This situation reflects the necessity of automated methods that allow to perform the task in a more productive and secure way. The productivity of the inspection tasks can be greatly improved by the digitalization of the assets to study. For example, Mobile Mapping systems (MMS) are appropriate methods to record data of railway and road infrastructures, because they allow to generate massive amount of geometrical data in short periods of time [9,10,11].

Due to the emerging necessity of the digitalization of the infrastructures, different technologies have risen their popularity. For example, point cloud data allows to have a 3D representation of the environment in a digital manner, so its geometry can be studied in an automated way. Also, Building Information Modeling (BIM) is gaining importance over the years, because it is an evolution of the traditional design systems and it provides a centralized solution that has all the information of the infrastructure in a single model[12]. This is especially relevant because BIM systems are able to digest new information about the infrastructure, and thus, resulting in a good solution to continuously update the digital model, and so, contribute to optimize the information handling and exploitation [13,14,15].

Mobile Mapping Systems (MMS) is an emerging technology to capture actual data of the railway infrastructure in an automated way. Depending on the sensors equipped, various types of data can be captured by MMS. The simplest type is image data. For this case, the system is usually equipped with a 360° camera that allows to record images in all directions. Also, more sophisticated sensors as thermographic cameras can be equipped. Finally, the system can also generate

* Corresponding author.
E-mail addresses: javier.grandio.gonzalez@uvigo.es (J. Grandio), belenriveiro@uvigo.es (B. Riveiro), msoilan@usal.es (M. Soilán), parias@uvigo.es (P. Arias).

https://doi.org/10.1016/j.autcon.2022.104425
Received 21 January 2022; Received in revised form 7 June 2022; Accepted 8 June 2022
Available online 18 June 2022

3D data, that are represented by 3D point clouds [11,16]. On one side, RGB images rely on the availability of light to capture the environment, since its absence degrades the quality of the information obtained. On the other side, 3D representations allow to exploit all the geometrical characteristics, in contrast to the plane views provided by images.

To record 3D data, Light detection and ranging (LiDAR) is one of the main technologies, since it allows to capture the environment with high resolution and unprecedent accuracy [17,7,18,19]. These sensors allow to record even small details of the geometry of an object, presenting the information as point cloud data. As explained earlier, the point cloud data is a highly accurate 3D representation of the environment. An individual point cloud is defined by a set of points in a 3D coordinate system that represent the surfaces of the objects present in the environment. Also, depending on the sensor used for the capture, and other sensors that can be integrated with it, the points may have additional attributes that give information about the surface where they are found: i) Color field recorded with a 360° camera can be integrated to give information of the color of the surface where a point is found; ii) Intensity attribute is calculated as a representation of the reflectance of the surface; iii) The number of returns of the laser can also provide information about the object recoded; iv) finally, other fields regarding the capture itself such as sensor angle and timestamp are recorded to provide additional information.

The point cloud data open the possibilities to work with as-is infrastructure data. There are several typical applications for point clouds. The most popular one is related to the modeling of buildings and infrastructure, but it can be also used for inventory tracking, geometry quality inspection, construction progress analysis, and others (Q. [20]). In particular, due to the evolution of point clouds technologies and BIM, the development of methods to create as-is BIM models from point cloud data has also gained great importance along the years [21,22]. Also, the development of techniques for forest inventory from point clouds are widely spread [23,24], and it can be extrapolated to railway and road infrastructures.

Regarding the application of point clouds in the railway infrastructure, the trends in research include the semantic segmentation of the environment [25,26,27] in order to generate as-is BIM models of the infrastructure. The semantic segmentation of the point cloud consists of providing a classification value to the individual points based on the object that they belong to. By doing this, all the points of the clouds that belong to the same type of object get the same classification value. This segmentation step allows to locate the different objects that constitute the model of the infrastructure.

However, to the best of the authors' knowledge, most of the existing methods in the literature used for semantic segmentation in railway environments are based in heuristic approaches. The main drawback of these approaches is their strong dependence on parameters given by the designer, and their low generalization capacity. They rely on parameters that have been calculated based on the characteristics of the point clouds used during development. This creates a strong dependence on the homogeneity of the point clouds studied, and small changes such as the LiDAR sensor used, may prevent the methods from working properly.

On the other hand, deep learning has been taking more and more importance in several fields during the past few years, and several methods focused on point clouds have been developed. The main tasks of the deep learning methods developed for point cloud analysis can be divided into classification, object detection, object segmentation and semantic segmentation [28].

Considering all the information presented, this paper presents a methodology based in deep learning to segment point clouds from railway environments. The method is applied to different railways environments, and point clouds registered from different sensors, to show its capability to generalize. The architecture of the neural network has been designed according to the data treated to obtain the best possible results. Also, the input data is properly pre-processed to enhance the capability of the neural network.

As a result, the main contribution of this paper is to present a method to segment point clouds from the railway infrastructure, presenting the following characteristics:

- It achieves the segmentation of the most relevant assets of the railway environment including both punctual and continuous objects.
- It generalizes to work in different environments and point clouds obtained with sensors of different quality.

The remaining sections of the paper are structured as follows. Section 2 presents different works related to the topic, section 3 explains the methodology developed for the segmentation, section 4 presents the results obtained in the different scenarios studied, and section 5 discusses the results. Finally, in section 6 the conclusions are presented.

2. Related works

In this section, the state of the art of the methods and domain of our work is presented. Those works can be divided into two broad categories: (1) Semantic segmentation of railway infrastructure assets. (2) Deep learning methods for semantic segmentation of point clouds.

2.1. Semantic segmentation of railway infraestructure assets

Most of the methods found in the literature for segmentation of point clouds in railway environments are based in heuristic approaches. Those methods are based in studying the morphology of the assets, and logic rules are applied to segment them.

Some of the approaches focus only on some assets of the infrastructure. For example, in [29] a method to detect the rails of the track is presented. Also, in [30] the rails of one side of the track are detected, and in this research, they extend the work to automatically generate alignment entities of the rails, following the Industry Foundation Classes (IFC) standard, which allows translating the information extracted to BIM models. A more extended work is found in [31]. In this case, the author presents a method to detect both linear and punctual objects in rural railways. The main drawback of this work is that is has been only tested in 550 m of railroad, so it is hard to determine if the method is robust enough to generalize to different scenarios. Finally, in our previous work [32], we present a heuristic method that segments all the relevant assets found in the railway environment in a 90 km long track.

Other approaches that rely on machine and deep leaning techniques have been also designed for the task. In [33], they present a method that uses a combination of heuristic calculations with Support Vector Machines. However, it is only destinated to work in railway tunnels and it cannot be applied to other infrastructure assets. The results obtained in the railway tunnels were later used in [34] to train the deep leaning neural networks PointNet [35] and KPConv [36], obtaining similar performance to the heuristic approach.

Finally, there are also methods based in deep learning to segment some assets of the railway environment, but they rely on image data instead of point clouds. In [37], the authors present a method based in deep neural networks to segment the railway track. This method is based in image segmentation using Convolutional Neural Networks (CNN). By using images instead of geometrical data, part of the geometry of the environment is lost. Also, the time spots available for the recordings are usually at night, which highly affect the capacity of this method if no light is available.

2.2. Deep learning semantic segmentation

Regardless of the lack of implemented deep learning methods for point cloud segmentation in railway infrastructure, the creation of these type of methods for general purposes has been exponentially growing over the past years. In fact, due to the broad nature of the approaches taken in this field, it is necessary classify them in different groups. The

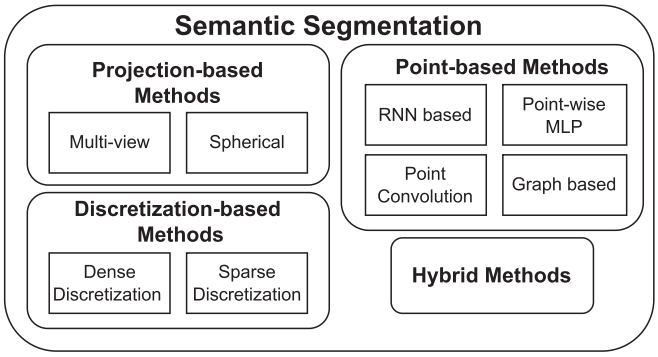


Fig. 1. Deep Learning Semantic Segmentation Taxonomy.

taxonomy to divide the methods in different categories followed in this work is found in [28], and it is represented in Fig. 1.

Projection-based methods work by generating images from the point clouds. Those images are segmented using Convolutional Neural Networks, that are known to provide state of the art results for the task, and then, they are projected back to the original point clouds to generate per-point classification. In [38], the authors generate several images from the point cloud, and output a prediction score for each pixel of the images, then the final label of each point is calculated by using the different scores obtained from the images. However, by relying in image data, the methods are sensitive to the point of view chosen to generate the images, and the full geometry of the point cloud may not be exploited.

The **discretization-based methods** rely on generating a discrete representation of the original point clouds. The most used method for the discretization is voxelization, which consist of generating a regular 3D grid with information of the points contained in each cell. This type of representation generates grid data such as an image, but adding a new dimension. In consequence, the CNN approach usually taken for images can be generalized to the use of 3D-CNNs as done in [39]. The main drawback of these techniques is the need of memory and precision loss. By discretizing, the size of the cell determines the maximum precision, and the memory usage grows cubically with the size of the point cloud, so these methods are not to be applied for large scale point clouds. The drawback related to memory usage has been solved later using sparse representations [40], that ignore the empty cells that usually represent the greatest part of the grids.

Point-wise methods work directly with the coordinates of the point clouds. As a solution to the infeasibility to apply CNN to the raw point clouds, PointNet [35] was proposed as a network that applies Multi-Layer Perceptron (MLP) to the individual points to extract features. PointNet is considered the pioneer of point-wise methods. With PointNet as a starting point, different types of approaches have been proposed, and they can be divided into: i) Point-wise MLP methods. These methods are based in PointNet, but they introduce improvements. PointNet ++ [41] applies PointNet hierarchically from larger to local regions to capture features at different scales. RandLA-Net [42] is proposed as a lightweight network that can be applied to large-scale point clouds. ii) Point Convolution Methods propose convolution operation for point clouds. PCCN is proposed in (S. [43]), which is based in parametric continuous convolutions. KPConv [36] presents Kernel Point Convolutions that determine their weights based on Euclidian distances to the points. iii) RNN-based methods use Recurrent Neural Networks (RNN) that are fed sequentially [44,45]. Finally, iv) Graph-based Methods build graphs from the original point clouds and apply graph networks ([46]; L. [47]).

Finally, with respect to the use of deep learning methods to segment point clouds from the transport infrastructure, there are several works

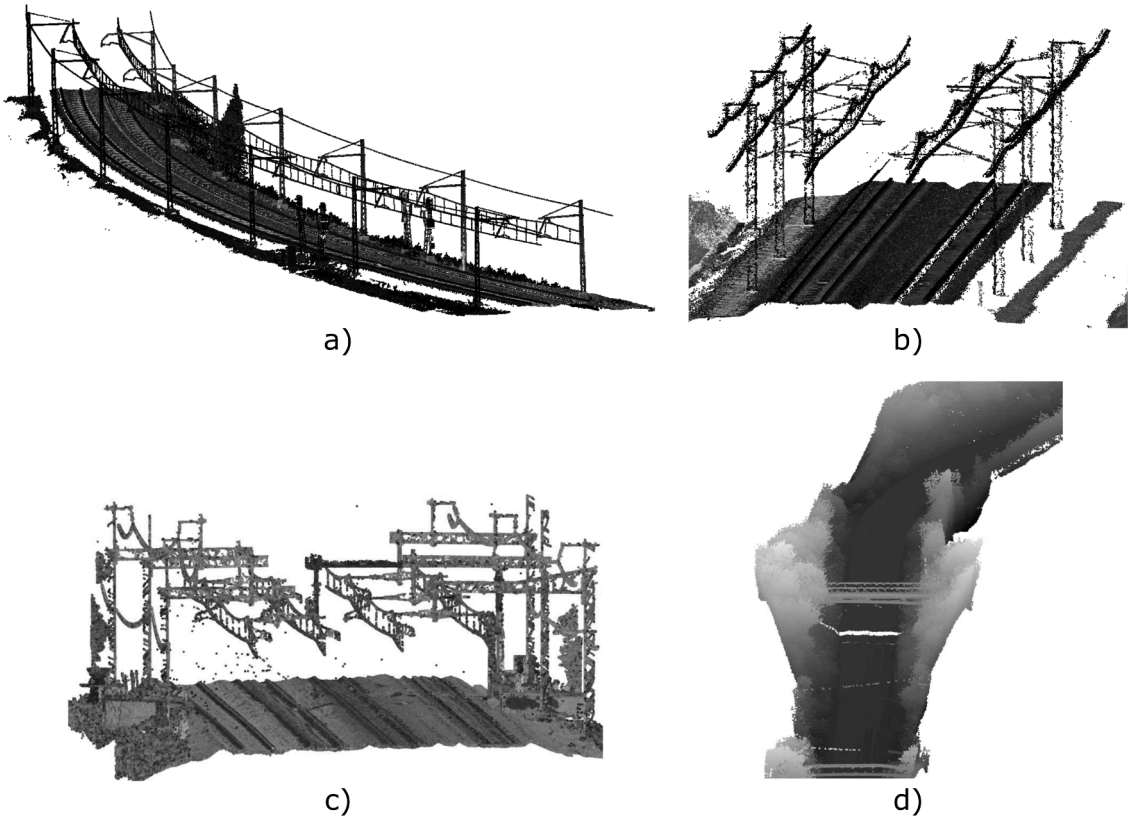


Fig. 2. Railway scenarios: (a) presents a sample cloud from the scenario used for training; (b) represents a portion of the second scenario, recorded with a low quality LiDAR; (c) corresponds to a high quality point cloud; and (d) has been recorded with a UAV, so its point density is the lowest.

Table 1
Distribution of classes.

Total Points	Background	Traffic Signs	Informative Signs	Traffic Lights	Masts	Cables	Droppers	Rails
2,433,806,176	2,287,128,527	472,901	741,339	618,608	29,125,569	37,251,096	1,257,623	77,210,513

focused on segmentation and object detection on roads[48,49,50]. However, regarding railway infrastructure, to the best of the authors' knowledge, there are no deep learning applications found in the literature. The only work related to the topic focuses on the segmentation of railway tunnels using KPConv and PointNet [34].

3. Methodology

3.1. Case of study

The work presented in this paper consists of performing semantic segmentation on railway infrastructure point clouds. In order to validate the methodology and results, the approach taken is tested in four scenarios, each one of them containing different railway data. The scenarios have been recorded with different sensors, and some of them are even recorded from different countries, which makes the geometry of the railway to be slightly different among the cases.

The first scenario is the largest one, and it is used to build the training dataset. CloudCompare software [51] have been used to display point clouds in this paper, such as the sample cloud of this scenario, shown in Fig. 2 (a). This dataset is presented in [30]. It consists of 90 km of railway, and it was surveyed with an average speed of approximately 10 km/h. The system used for the survey is the LYNX Mobile Mapper by Optech (*Teledyne[52]*), which uses two LiDAR sensors. To make the information manageable by conventional hardware, it was divided in 450 individual georeferenced point clouds of 200 m long saved in .las format. Each point cloud has an average of 7 million points, so the complete dataset comprises more than 2000 million points. Also the point clouds present a point density of 980points/m^2 and range precision of 5 mm. Aside from the 3D information of each point, intensity values are also provided. The dataset has been previously segmented in [32] using an heuristic method, and those results are used as ground truth to train the neural network presented in this work, and to calculate the quantitative metrics of the results obtained. Table 1 shows the number of points belonging to each class.

The second scenario consists of a 2 km long georeferenced point cloud in a single .las format file, this cloud is shown in Fig. 2 (b). The point cloud has been recorded with an economic system called G_lidar (*Ingenieria[53]*). It provides lower precision than the dataset used for training, having a total of 39 million points with a point density of 644 points/m^2, and range precision of 30 mm. This point cloud has not been previously labelled, so the results presented are studied qualitatively.

The third scenario consist of two 200 m long point clouds in .las format, a sample cloud is found in Fig. 2 (c). The point clouds have been recorded with a RIEGL LiDAR [54], which provides a point density of 11,000 points/m^2 and range precision of 5 mm, having a total of 129 million points in 400 m. These point clouds have not been previously labelled either.

Finally, the last scenario consists of an aerial capture, the cloud is found in Fig. 2 (d). It is known that aerial mapping systems provide much lower quality captures than mobile mapping systems, so this is the worst point cloud in terms of data quality. This low quality is translated into a point density of 23 points/m^2, which is not comparable to any of the other three datasets. The railway captured is 1.4 km long, and the point cloud has 910,000 points. Also, since this point cloud has not been labelled either, quantitative metrics are not calculated.

As it can be seen, the characteristics of the point clouds captured differ among them, so the neural network must be able to generalize for different precisions and densities. This makes the task harder than the case of segmenting only one scenario.

3.1.1. Objects to be segmented

The main goal is to segment the point clouds to separate different objects.

As it has been explained earlier, the dataset used for training has been previously segmented using a heuristic algorithm. As expected, the heuristic algorithm is not perfect, and it has some misclassifications. The misclassifications have not been manually corrected, since they are isolated and, in many cases, hard to locate manually. However, the neural network is expected to overcome those errors and be able to generalize.

The labels used to train the neural network differentiate eight different categories in the point clouds.

- **Background**. Refers to all the data that do not have semantic meaning.
- **Informative signs**. Small signs with low intensity values. Their geometry is simple, as shown in Fig. 3 (a).
- **Rails**. This category includes all the rails present in the point clouds. Transversal profiles are shown surrounded by boxes in Fig. 3 (b).
- **Cables**. This class includes all the cables present in the railway except for the droppers.

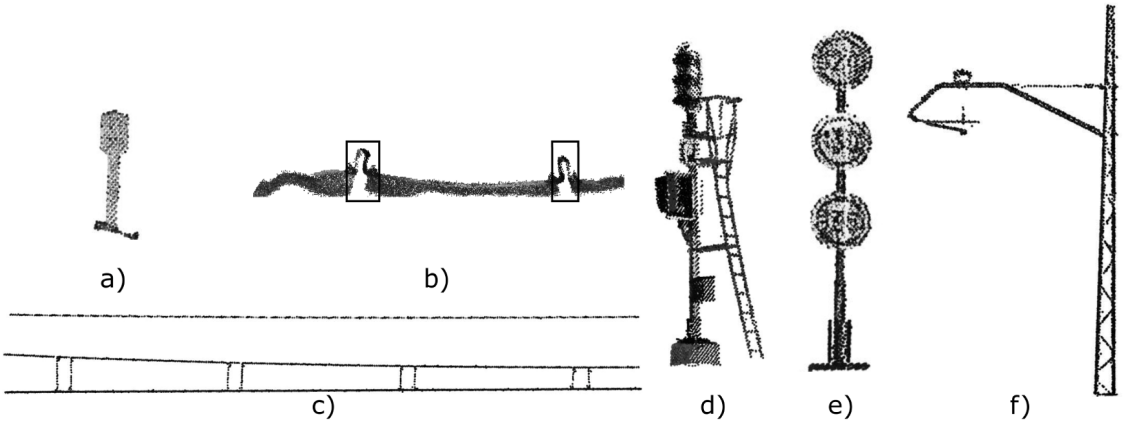


Fig. 3. Objects to detect.

- **Droppers**. These are vertical cables that join the catenary cables with the contact wires. Cables and droppers are shown in Fig. 3 (c).
- **Traffic lights**. An example is shown in Fig. 3 (d).
- **Traffic signs**. Includes signs that refer to speed restrictions and related. They usually have high intensity values. An example is found in Fig. 3 (e).
- **Masts**. This class includes the masts present in the railway. An example is shown in Fig. 3 (f).

3.2. Methodology

This section presents the methodology developed for the semantic segmentation of the railway point clouds and its implementation. To have a fully understanding of the method, it is necessary to explain the neural network architecture used for the task, the steps taken to train the neural network, and data pre-processing.

3.2.1. Neural network architecture

The use of deep learning on point clouds has grown over the last decade [28], and many new architectures have been designed. The one that is considered the pioneer is PointNet, which has been improved by different authors over the years. PointNet is one of the methods that are denominated point-based methods, since it works directly with the coordinates of the point clouds to predict the results.

In this case, the architecture used for the task is based on PointNet++[41], which is an improved version of PointNet. PointNet++ has been widely tested in the literature, providing good results and performance. As backbone architecture, U-Net like architecture is used. The main characteristic of this backbone architecture is that it has residual connections between the encoder and the decoder parts of the neural network. It is widely used in computer vision tasks [55], and it has also been used for point clouds. Specifically, all encoder layers have symmetrical residual connections with the decoder.

PointNet uses a function f that maps an unordered set of points to a vector [35]:

$$f(x_1, x_2, \ldots, x_n) = \gamma\left(MAX_{i=1,\ldots,n}\{h(x)\}\right)$$

Where γ and h are multilayer perceptron.

As an improvement to PointNet, PointNet++ proposes a hierarchical approach where points are sampled at different scales and PointNet is applied to subsets of points. Fig. 2 from [41] illustrates PointNet++ base architecture. This allows not only to capture global features of the point cloud, but also local features, increasing the performance of the network.

Although PointNet++ is used as base architecture. When facing different tasks, the architecture of the network should be modified to be adapted for the given task. In the results sections, the performances of three different architectures are compared to determine which one fits better for the objective of this work.

The starting point for the network architectures is the one presented in [41] to segment Scannet dataset [56]. This approach is taken because it is the only segmentation architecture presented in the original implementation, and it provides good performance. Scannet is an annotated dataset of indoor point clouds. Compared to Scannet, the objects present in railway infrastructure are much larger. In consequence, the main modifications applied to the architecture proposed for Scannet are: i) number of points sampled per layer, ii) radius size for each point, iii) number of points sampled within the radius.

In first place, while handling Scannet, the authors use 8192 input points, this is not enough to represent the scenes in the railway environment, so a minimum of 16,384 points is proposed in this case. In second place, since the input points have been increased, the sampling points in the network layers must be also increased. A minimum of 4096 sampling points is proposed for the first layer. Taking into consideration the values proposed, a new architecture is proposed using those minimum values, and having the rest of the sampling points dimensioned proportionally to the parameters proposed. A summary of the architecture is presented in Table 2, and detailed explanations of the parameters presented in the table are found in [35,41].

Table 2
Simple network architecture. Smaller size in return of better computational performance.

Simple architecture					
Downsampling layers					
Layer	**0**	**1**	**2**	**3**	**4**
Point Features	[7, 32, 32, 64]	[67, 64, 64, 128]	[131, 128, 128, 256]	[259, 256, 256, 512]	[515, 512, 512, 1024]
N Points	4096	1024	256	64	16
Radius	0.1	0.2	0.4	0.8	1.2
N Samples	32	32	32	32	32
Upsampling layers					
Layer	**5**	**6**	**7**	**8**	**9**
Point Features	[1024 + 512, 512, 256]	[256 + 256, 256, 256]	[256 + 128, 256, 128]	[128 + 64, 128, 128]	[128 + 4, 128, 128, 128]

Table 3
Complex network architecture. Uses twice as many sampling points in each of the layers, and the sample radius is reduced. Bigger size in return of a worse computational performance.

Complex architecture					
Downsampling layers					
Layer	**0**	**1**	**2**	**3**	**4**
Point Features	[7, 32, 32, 64]	[67, 64, 64, 128]	[131, 128, 128, 256]	[259, 256, 256, 512]	[515, 512, 512, 1024]
N Points	8192	2048	512	128	32
Radius	0.05	0.1	0.2	0.4	0.8
N Samples	64	32	32	32	16
Upsampling layers					
Layer	**5**	**6**	**7**	**8**	**9**
Point Features	[1024 + 512, 512, 256]	[256 + 256, 256, 256]	[256 + 128, 256, 128]	[128 + 64, 128, 128]	[128 + 4, 128, 128, 128]

Several versions of the architecture were tested. However, the influence in the results was not significant in most cases when the networks were very similar. In consequence, only two new architectures are proposed. The second architecture uses twice as many sampling points as in the first architecture for the first layer. Also, the radius to sample features of each point is half size of the one initially proposed, while the sample points within the radius is doubled in the first layer. With these changes, there are more features extracted in each layer, but in a smaller scale. A summary of this architecture is presented in Table 3.

3.2.2. Training

The neural networks applied for point clouds found in the literature are developed using different deep learning frameworks and versions. As a solution to this problem, in [57], the authors introduce an open-source framework to work with deep learning methods on 3D data based in Pytorch. The framework has a modular design, and it allows customization, making it suitable for research purposes. Due to the mentioned advantages, this framework has been chosen to develop the neural network training, creating all the additional modules needed for the method and modifying some of the existing ones.

The hardware used for the training process consist of a GPU Nvidia

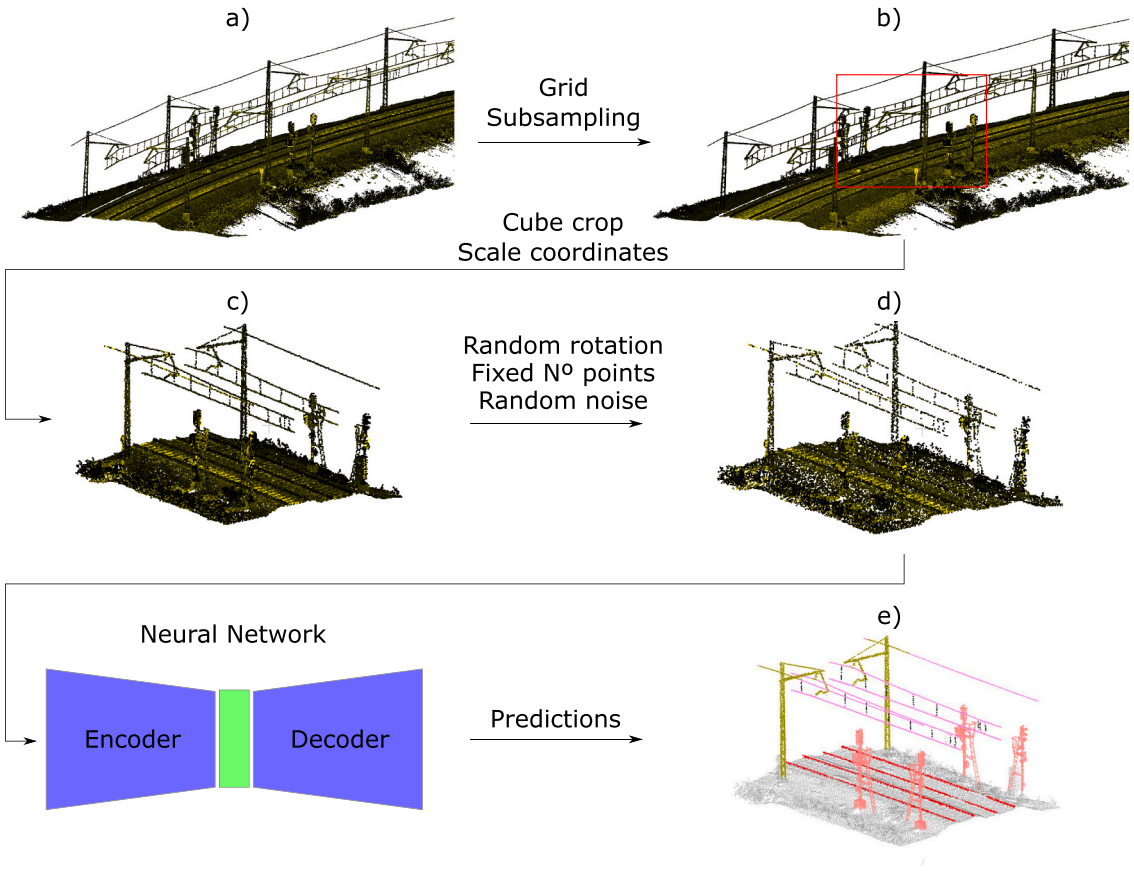


Fig. 4. (a) Presents the raw point cloud provided by the sensor; (b) shows the subsampled point cloud; (c) is a cropped cube and (d) shows the point cloud fully preprocessed that is fed to the neural network. Finally, (e) represents the predictions obtained by the neural network.

GeForce RTX 2060 Mobile, an Intel Core i7-10750H CPU, and a 16GB DDR4 RAM.

The data need to be split for training and testing. So, 80% of the clouds are taken for the training set, and the rest of them are used for testing. This subsection explains the steps followed to train the neural network and how the data has been processed. A graphical summary of the processing steps taken is presented in Fig. 4.

3.2.3. *Training data preprocessing*

Since the point clouds may have several attributes to characterize each point, first, it is necessary to declare the features of the point cloud that are fed into the network. In the case of this work, Euclidian coordinates and intensity values are taken into account by the neural network, having as input vector: $X_{Nx4} = \begin{bmatrix} x_1 & y_1 & z_1 & I_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & I_N \end{bmatrix}$. Where N is the number of input points in the network, (x,y,z) represent the Euclidian coordinates of each point, and I corresponds to the intensity value.

Once the input features of the network are defined, a pre-processing step must be applied to the point clouds. In this work, the pre-processing steps taken for training are the following:

- **Balance training data**. Table 1 shows that the labels present in the dataset are clearly unbalanced. While most of the points belong to the background, only a few points are part of punctual objects such as traffic signs. This forces the neural network to learn features from the classes that are more populated, and ignore the others, because their contribution in the loss function is relatively low. Two different techniques are applied to solve the issue. First, weighted loss function is applied for training [58], increasing loss values when the least populated classes are misclassified. Secondly, data augmentation is also applied. The data augmentation consists of duplicating clouds from the training set that have traffic signs and traffic lights, applying them geometric transformations. Clouds containing those objects have been replicated, cropped taking only the space surrounding the objects of interest, rotated, and gaussian noise is applied to their points. With this, the number of points with those labels was incremented.
- **Scale intensity**. Not all the scanners provide intensity data in the same format. The values are always represented as integers, but depending on the number of bits that they use to represent the integers, the scale of the values is different, using always as maximum the largest value allowed by the number of bits used. The main consequence of this variability is that the precision of some scanners is lower than others. To overcome the issue of the different scales, the intensity values are scaled to [0,1] using the number of bits that the given sensor provides as reference to calculate the scale factor.
- **Grid sampling**. As mentioned earlier, the density of points varies depending on the sensor used during the recording of the clouds, or the velocity of the vehicle when surveying. This makes mandatory the capability of processing point clouds with different point densities. In order to alleviate the task to the neural network, all the point clouds are first subsampled using grid subsampling. This preprocessing step helps to homogenize the density of the clouds across all the available scenarios. The size of the grid used is one of the parameters studied for the results, and the effect in the segmentation accuracy is presented in results section.
- **Cube crop**. In order to provide a homogeneous point cloud size to the neural network, it is fed by cubes of 10 m each side. During the

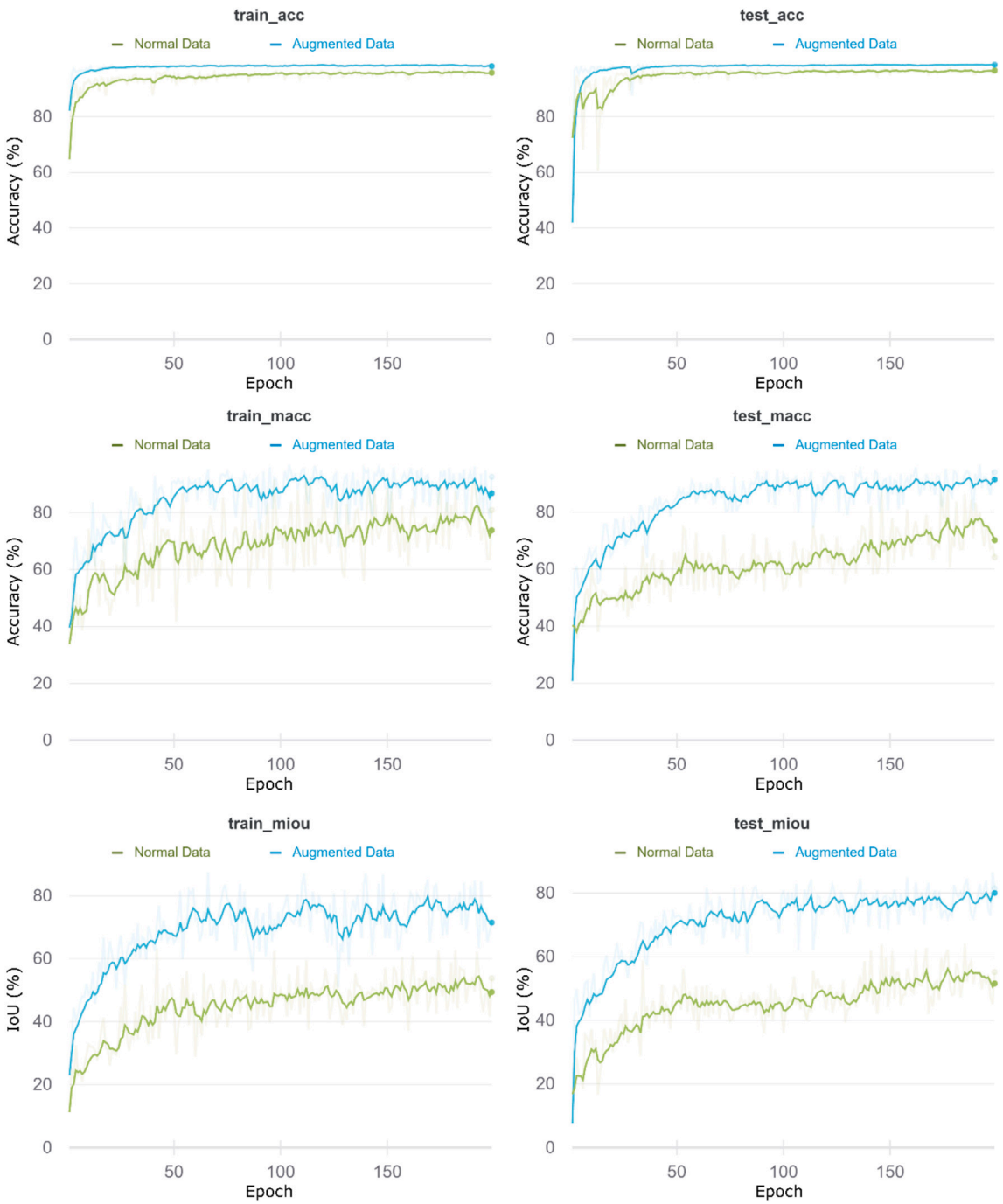


Fig. 5. Effect of data augmentation in the metrics. Accuracy (acc), mean accuracy (macc) and mIoU (miou) evolution for training and testing data at the end of each epoch, comparing the original training with the one using augmented data.

training, only one cube from each cloud is sampled in each epoch. For this purpose, a random point of the original cloud is taken in each epoch, and the cube around that point is sampled. With this, different sub-clouds are fed from the same cloud at different epochs.

- **Scale coordinates**. The coordinates of the point clouds are recorded in different reference systems depending on the scenario. However, in all cases, these values are usually high, in the order of thousands of meters. As is well known, having high input values may cause instability while training neural networks. To avoid this issue, once a cube is taken sampled, its coordinates are scaled to [0,1] values in all the axis, using the 10 m as the scale factor.
- **Random rotation**. Having more variability available in the training data helps the network to generalize better afterwards. So, to avoid the network to depend on the orientation of the railway track to segment correctly, the cubes used for training are randomly rotated. On the one side, the maximum rotation about **x** and **y** axes are restricted to 15°, because the railway tracks avoid having slopes as much as possible. On the other side, the rotation about **z** axis is not restricted because the track will always be randomly oriented on that axis.
- **Fixed number of points**. Since PointNet++ architectures need to be fed always by a given number of points, the cube cloud is randomly subsampled with replacement to N points. This N must be big enough to be representative of the cloud, but the smaller it is, the faster than the network will work. So, the trade-off between precision and velocity is studied in the results section.

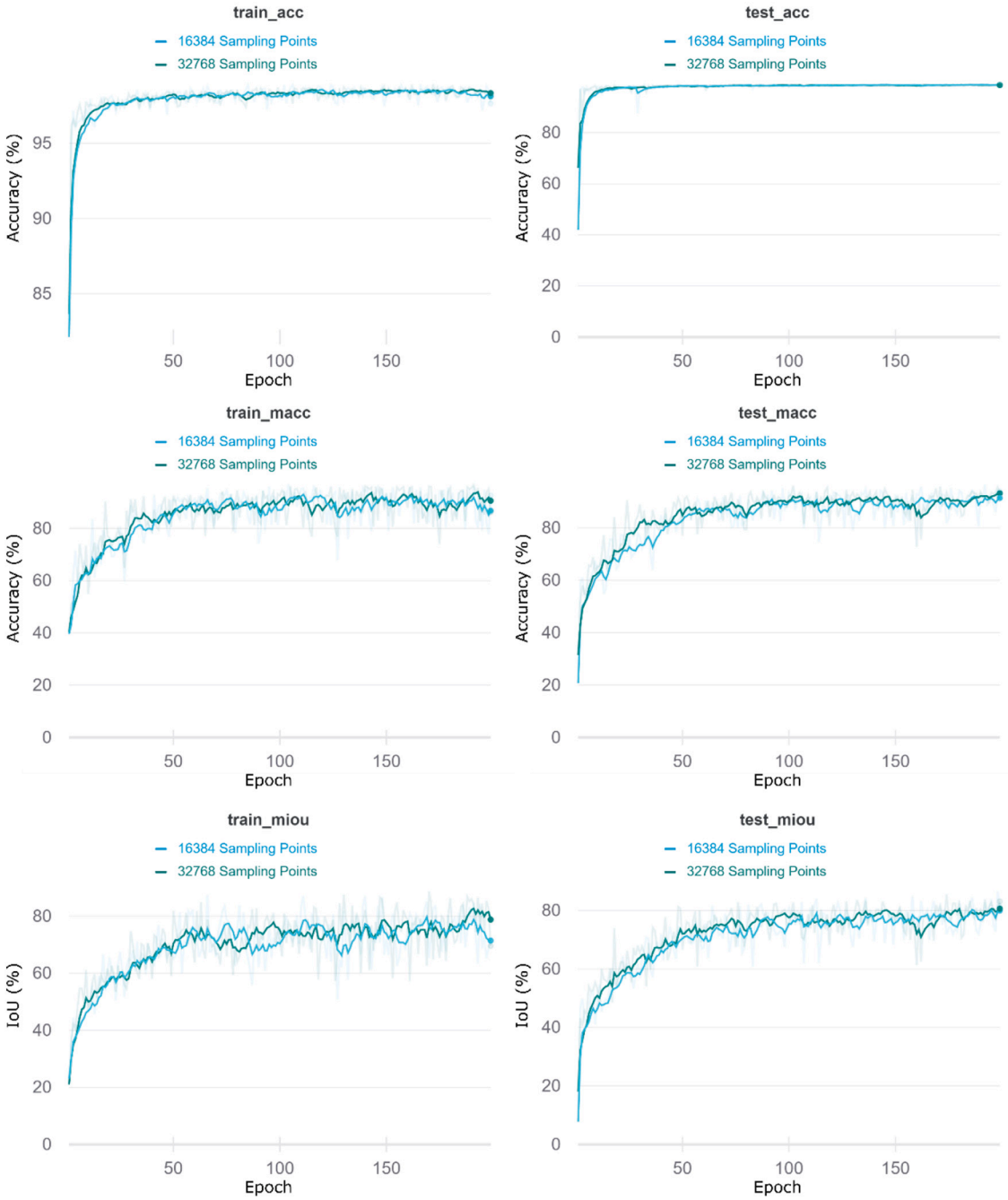


Fig. 6. Effect of the number of input points in the network. Accuracy (acc), mean accuracy (macc) and mIoU (miou) evolution for training and testing data at the end of each epoch, comparing the training with 16,384 input points with the one using 32,768 input points.

- **Random noise**. To add variability to the training clouds, random noise following a normal distribution is added to the coordinates before feeding the network.

Finally, cross entropy loss function and Adam optimizer are used with the following hyperparameters: learning rate = 0.001, batch normalization momentum = 0.1 and batch size = 5, which has been limited by hardware capabilities.

3.2.4. Testing process

After finishing the training, the neural network must be tested against data that it has not been trained on. The main difference during the testing process with respect to the training is the pre-processing of the data. In this case, the point clouds are divided into regular cubes, and those cubes are fed to the network applying only grid and random subsampling. Then, the results are compared to the ground truth of the dataset in order to calculate metrics.

4. Results

The method has been tested generating different cases where pre-processing steps are modified, as well as using different architectures for the neural network trained. In all cases, the neural network is trained for 200 epochs, and the metrics over random samples in the test set are calculated after each epoch in order to study the training performance. The metrics graphs obtained are smoothed to overcome the variance due to the randomness of the crops taken to calculate the metrics after each epoch.

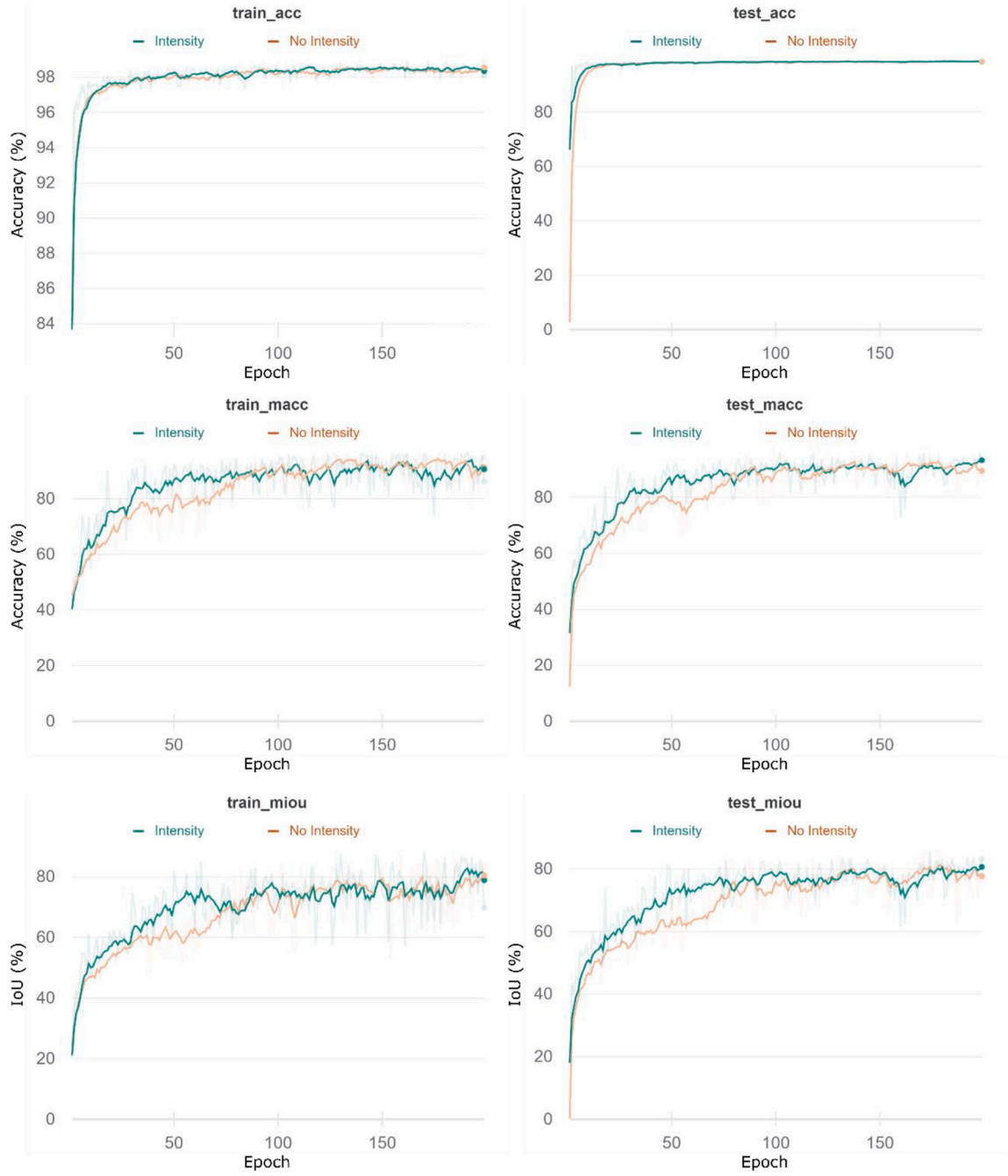


Fig. 7. Effect of the intensity feature in the results. Accuracy (acc), mean accuracy (macc) and mIoU (miou) evolution for training and testing data at the end of each epoch, comparing the training using the intensity as input feature and the training that only uses the Euclidean coordinates.

4.1. Metrics

Depending on the task to evaluate, there are different metrics that measure better the performance. In this case, the metrics used to evaluate the results are widely used for semantic segmentation, both for images and 3D data. The metrics are the following [59]:

- **Overall Accuracy (OA):** It represents the percentage of points correctly classified, regardless of its class. This metric is not significative for those cases where unbalanced data is present in the dataset. For example, in case of classifying all the point as background, the OA would be higher than 90%.
- **Mean Accuracy:** This metric takes into account the accuracies of all the classes and calculates their mean. This overcomes the issue caused by the unbalanced data.
- **Intersection Over Union (IoU):** It is calculated following Eq. 1. It measures the number of points common between the label and prediction masks, divided by the total number of points present across both masks.

$$IoU = \frac{True\ Positives}{True\ Positives + False\ Positives + False\ Negatives} \tag{1}$$

- **Mean IoU:** It takes into account the IoUs of all the classes and calculates their mean. The version of the network that gets the highest validation mIoU is saved as the best version.

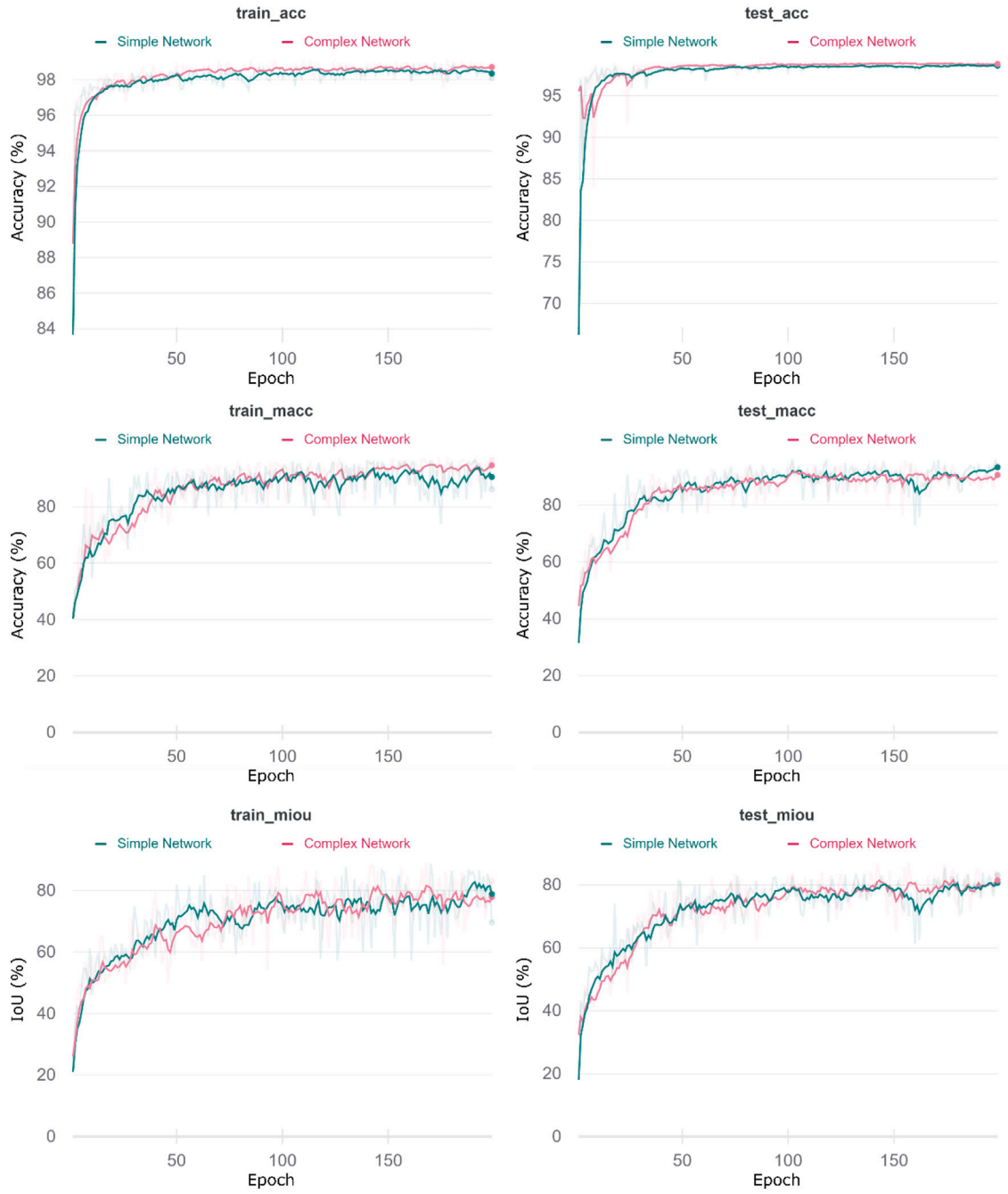


Fig. 8. Effect of the network architecture in the result metrics. Accuracy (acc), mean accuracy (macc) and mIoU (miou) evolution for training and testing data at the end of each epoch, comparing simple and complex networks.

4.2. Data augmentation

Several versions of the method are compared to study the effect of the changes in the results.

In this first comparison, two cases are studied. In the first case, during the pre-processing step, the point clouds are not augmented, and the network is trained only using the original data. In the second case, clouds containing traffic lights and signs are duplicated and modified applying geometric transformations to augment the training data.

Results in Fig. 5 show the metrics curves obtained for both cases during training. For the first 100 epochs, the metrics improve fast because the model is starting to fit the data. Then, since the network is closer to converge, the improvement is slower. However, it is known that overfitting is not happening because the testing metrics also improve. The results obtained due to the data augmentation are considerably better than the other ones, getting overall improvements of 10% for the test mean accuracy and 25% for test mIoU. Since the improvement is so significant, all the posterior comparatives are using the augmented data for training.

4.3. Grid size and number of input points

The grid size determines the maximum precision that an input point cloud may have. Having a small grid size results in a high number of voxelized points. On the other side, since random points are sampled after the grid sampling, in case of sampling randomly a significantly lower number of points than the ones left after the first sampling, significative points of the point cloud may be lost. However, if the

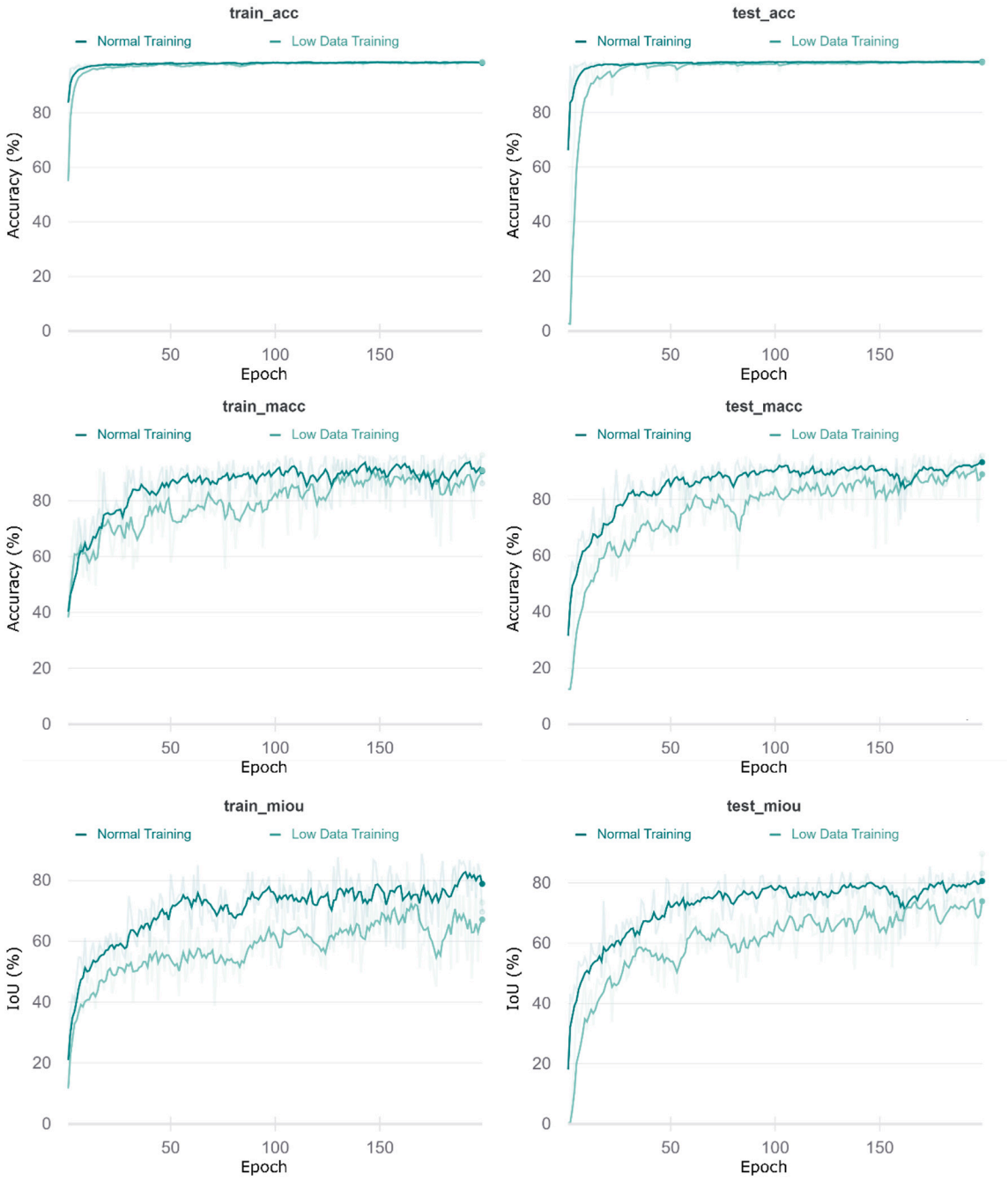


Fig. 9. Effect of the dataset size during the training. Accuracy (acc), mean accuracy (macc) and mIoU (miou) evolution for training and testing data at the end of each epoch, comparing the training using the original dataset and the training using the testing set for training.

number of remaining points after grid sampling is similar to the number of points that are randomly sampled, important information is not likely to be lost. In consequence, these two parameters are supposed to be modified proportionally for the experiment, in order to have a good compromise between the two of them.

In this comparison, two different approaches are tested. The first one consists of a grid sampling of 0.1 m, and a random sampling of 16,384 points. The second one uses a grid of 0.05 m and samples 32,768 points. Using more points would require more computational resources and increase running time, while low-density point clouds cannot take advantage of a more fine-grained sampling.

Fig. 6 shows how results are similar, and there is no improvement with the increment of points. However, when using 16,384 points, the output cloud loses quality compared to the most populated one, so 32,768 points are used for the rest of the experiments.

4.4. Intensity

As explained in the previous section, one of the inputs of the network is the intensity. This intensity value depends on the number of bits and the sensor used during the capture, and it can differ from one point cloud to another. It is also interesting to study if the intensity has a significative impact on the performance of the network, or if it is worth to have a more general neural network that does not rely on the intensity to perform well on the task. In consequence, the training results using and obviating the intensity are compared.

Fig. 7 shows how the usage of intensity does not change the overall results. The main change is the time to converge. When using the

Table 4
Final results obtained in the experiments. Overall Accuracy, Mean Accuracy, mIoU and IoU for each asset are presented.

					Class IoU			
Training Data	Network	Input points	OA	Test Mean Acc	Background	Traffic signs	Informative signs	Traffic lights
Intensity	Grid Size (m)			Test mIoU	Masts	Cables	Droppers	Rails
Full	Simple	16,384	98.86%	90.37%	98.88%	60.75%	49.33%	**70.51%**
Yes	0.1			**76.33%**	80.92%	**92.39%**	**75.35%**	82.48%
Full	Simple	32,768	98.87%	89.86%	98.89%	54.46%	53.15%	64.77%
Yes	0.05			74.41%	**81.49%**	91.53%	71.05%	79.98%
Full	Simple	32,769	98.86%	**90.95%**	98.88%	55.65%	**59.71%**	59.17%
No	0.05			74.89%	79.22%	91.99%	73.58%	80.89%
Full	Complex	32,769	**98.97%**	88.41%	**99.01%**	58.70%	50.76%	69.85%
Yes	0.05			75.38%	80.09%	90.07%	70.81%	**83.76%**
Reduced	Simple	32,769	98.57%	86.15%	98.59%	61.30%	33.27%	53.75%
Yes	0.05			69.68%	76.02%	90.14%	66.25%	78.13%
Full	PointNet Original	32,769	97.61%	89.71%	97.61%	64.19%	57.49%	46.71%
Yes	0.05			68.70%	76.45%	90.80%	58.76%	57.65%
Full	KPConv	32,769	98.57%	86.18%	98.59%	**65.01%**	55.12%	59.82%
Yes	0.05			71.97%	79.81%	89.94%	52.34%	75.13%

intensity as an input, the results improve quicker than in the other case, converging about 30 epochs earlier. This is probably due to the traffic signs and lights, since they are known to have high intensity values, the network does not have to learn their geometrical features in case of having the intensity available.However, when working with the point clouds from the third scenario, the network that uses the intensity is not able to segment the point clouds correctly. This situation has been studied and the possible cause is presented in discussion section.

4.5. Network architecture

All the results shown have been tested using the same network architecture. In this section, a new network architecture is compared with the one used until the moment. For the sake of understandability, the network used until now is denominated simple network, and the other one is denominated complex network, since the second one has more parameters. In both cases, they are working with 32,768 input points and uses intensity as input feature, so their inputs are $\boldsymbol{X}_{32768x4} = [\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{I}.]$

Table 2 and Table 3 show how the main difference between the networks is the number of points remaining in the subsequent downsampling layers, and the radius considered in each one of them. These changes are translated into an increase of running time for the training, raising from 5.5 h when using the simple network, to 13 h when using the complex one. However, metrics in Fig. 8 show how there is not a significant improvement in the results, so the simple network is considered the best option for the task, and it will be used for the remaining the experiments. The possibility of increasing more the complexity of the network is not considered due to GPU memory restrictions and computational time.

4.6. Training set size

In this case, the availability of data for training is considerably high. To test the performance on cases with less data available, the network has been trained using the test set as training set. With this, the network is trained with a small dataset, and later it is tested against the whole old training set.

The metrics obtained during the training compared to the full version of the training are shown in Fig. 9. These results show how the convergence time is longer and the mIoU results for the low data training are slightly lower than in the standard case. In summary, although the overall results are worse, the method could be valid for cases where the data available is lower.

Table 5
Training runtimes.

Network architecture	Input points	Training runtime (minutes)
Simple	16,384	**113**
Simple	32,768	337
Complex	32,768	783
PointNet Original	32,768	248
KPConv	32,768	611

4.7. Results summay

Finally, all the networks that were trained using data augmentation pre-processing are tested against the whole test dataset. In order to study the impact of the PointNet++ architectures proposed, two state of the art semantic segmentation architectures have also been trained and tested. First, PointNet++ segmentation architecture proposed in the original implementation is trained and tested. Second, the original implementation of KPConv [36] is also evaluated.

For this test, all the test clouds are predicted completely to calculate the metrics shown in Table 4, which are discussed in the following section.

In addition, training runtimes for different network architectures and input points are presented in Table 5.

Besides the metrics obtained in the test set, several random clouds have been plotted to provide qualitative results about how good the method segments. This is important to determine if the noise can be easily postprocessed, since the error could be due to a small difference when defining the boundaries of the objects or having some objects that are completely misclassified. The qualitative studies for all the remaining scenarios have been tested with the neural network trained with the following characteristics: 32768 input points, simple architecture, and no intensity as input feature.

In Fig. 10, several random examples of the segmentation are shown. In these samples, most of the objects are correctly segmented. The most recurring error consist of segmenting parts of the masts´claws as cables, this might be caused because they are thin as the cables. On the other hand, it is remarkable how the method works both in exterior railway track as well as in the tunnels. Also, among all the testing clouds containing traffic lights and traffic signs, one of each has been selected randomly to study the performance of the network for each situation. Fig. 10 (a) has two signs and three traffic lights, that are correctly segmented by the network. Also, Fig. 10 (b) presents four different traffic lights that have been segmented.

Finally, 2 km of testing point clouds that overlap with the testing dataset from [32] have been manually evaluated for comparison. This

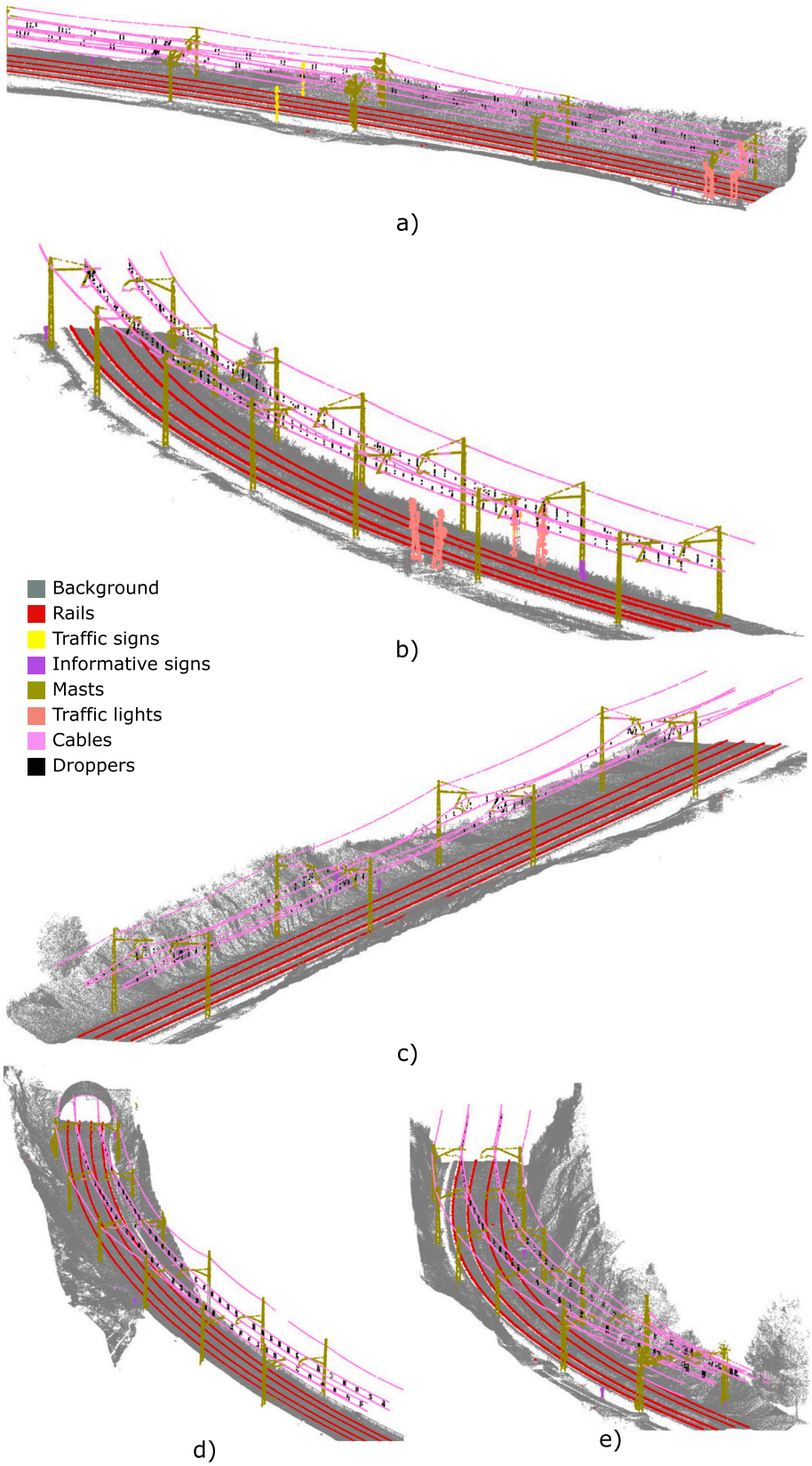


Fig. 10. Qualitative results in the first scenario. Track (a) has all elements present in the railway, being all and them correctly segmented. Tracks (b) and (c) are common cases in the dataset, having small curvature. Cloud (d) shows the beginning of a tunnel, and how it is also correctly segmented. Track (e) is surrounded by higher terrain.

Table 6
Manually calculated object-wise metrics.

	Precision	Recall	F1-Score
Rails	100.00%	100.00%	100.00%
Cables	98.90%	99.11%	99.01%
Droppers	99.91%	89.91%	94.65%
Masts	97.06%	100.00%	98.51%
Traffic signs	87.50%	100.00%	93.33%
Informative signs	90.48%	95.00%	92.68%
Traffic lights	100.00%	100.00%	100.00%

evaluation is done object-wise for punctual objects and measured by meters for linear objects, ignoring individual points of noise that may be present using this methodology. The results obtained are shown in Table 6.

4.8. Qualitative evaluation of generalization capability

The scenarios that do not have available the labels are relatively small, so qualitative results can be studied manually.

In the first place, the second scenario is tested. Fig. 11 shows the results. The main difference between this point cloud and the ones used for training is that this one has much more noise, in particular, the cables show noise around them, and the shape of the masts is not as well defined as in the other clouds. Regarding the results obtained, they are acceptable, the major problem is the lack of precision in the rails, showing some noise and having some small sections that have not been segmented as rails. It also presents some noise in the surroundings that could be easily postprocessed.

The results obtained for the third scenario are shown in Fig. 12. These data have been recorded in a different country than the training data, which causes that the geometries of the railway objects to be different from the training clouds. Regarding the posts, their geometry is considerably different, and even though, the network is able to segment correctly most of them, but showing some noise. Regarding the rails, most of them are correctly segmented. However, Fig. 12 (a) shows a lot of noise, segmenting some objects that are present in the interior of the rails, as rails. This issue has been studied, and the reason is that there are parts of old rails that have been temporally placed there. Another particularity of point cloud shown in Fig. 12 (b) is that is has two traffic lights, the first one is vertical, and it is correctly segmented, but the other one, shown in the zoomed section, hangs from a mast. This kind of traffic lights are not present in the training data, but the neural network is able to generalize the geometry and it partially segments it as a traffic light.

Finally, results obtained in the fourth scenario are shown in Fig. 13. These results show how the network has trouble segmenting the surroundings of the railway track. This is because the training data do not contain the surroundings of the track, so it is a new scenario for the neural network. The rails segmented have empty spots, which is mainly due to the low quality of the input point cloud, being even hard to recognize by humans.

5. Discussion

This section discusses the results obtained for the segmentation, studying the strengths and weaknesses found in the methodology.

In the first place, it was studied why when using intensity values as input, the network is not able to segment the third scenario. It was seen that density distributions of the intensity values are different depending on the scenario. As shown in Fig. 14, intensity values from the first scenario (a) follow a log-normal distribution, while the intensity values from the third scenario (b) follow a normal distribution. Obviously, this makes the neural network unable to generalize for the given case.

Regarding the testing results show in Table 4, while most of the approaches have mIoU values around 75%, this metric drops for the two architectures that have not been designed for the task, and the approach with reduced training data. This fact supports the work carried out to adapt the architecture for the given task.

As for as the other approaches studied, the best results are obtained by the one that uses 16,384 points as input. However, the difference is low with respect to the other cases. And this small difference could by caused by the randomness of the training. In general, the results show how different approaches perform better for different objects, but, in most cases, the difference is not significative.

With respect to the IoU values obtained for each object, it is clear than the main issue are traffic signs and traffic lights. As is has been highlighted earlier, the presence of these objects in the dataset is low, so the network does not have enough training data to perform as well as it does with the other classes. And, since the number of objects belonging to those classes is low, small errors reduce more significantly the IoU. Also, it is interesting how traffic and informative signs perform better with the neural network architectures that provide the worst general results.

Taking into consideration the results obtained, the architecture that

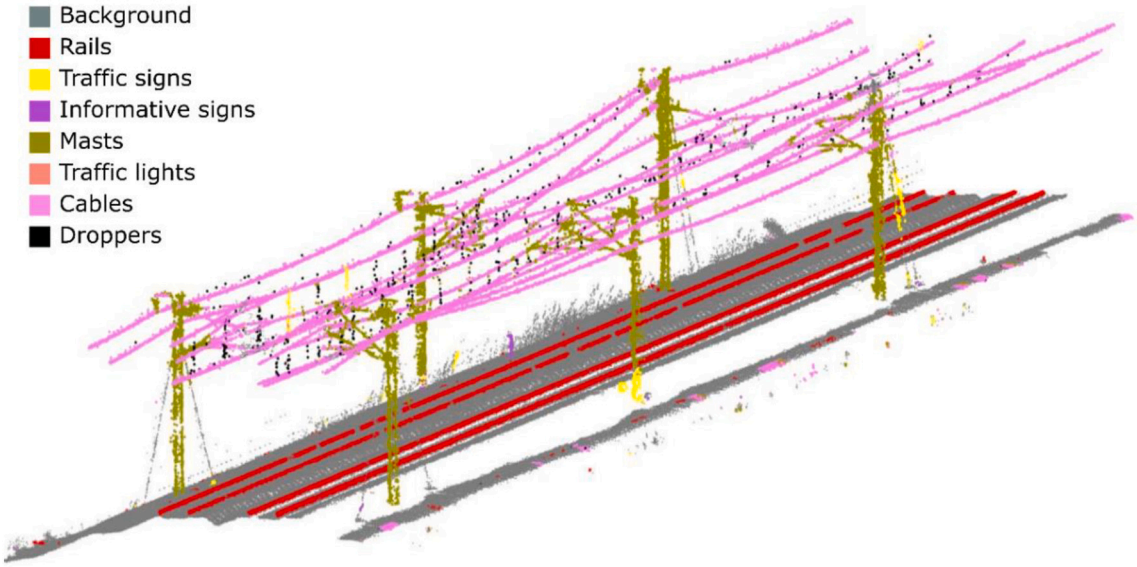


Fig. 11. Second scenario results. The point cloud is noisy and contains only masts as punctual objects.

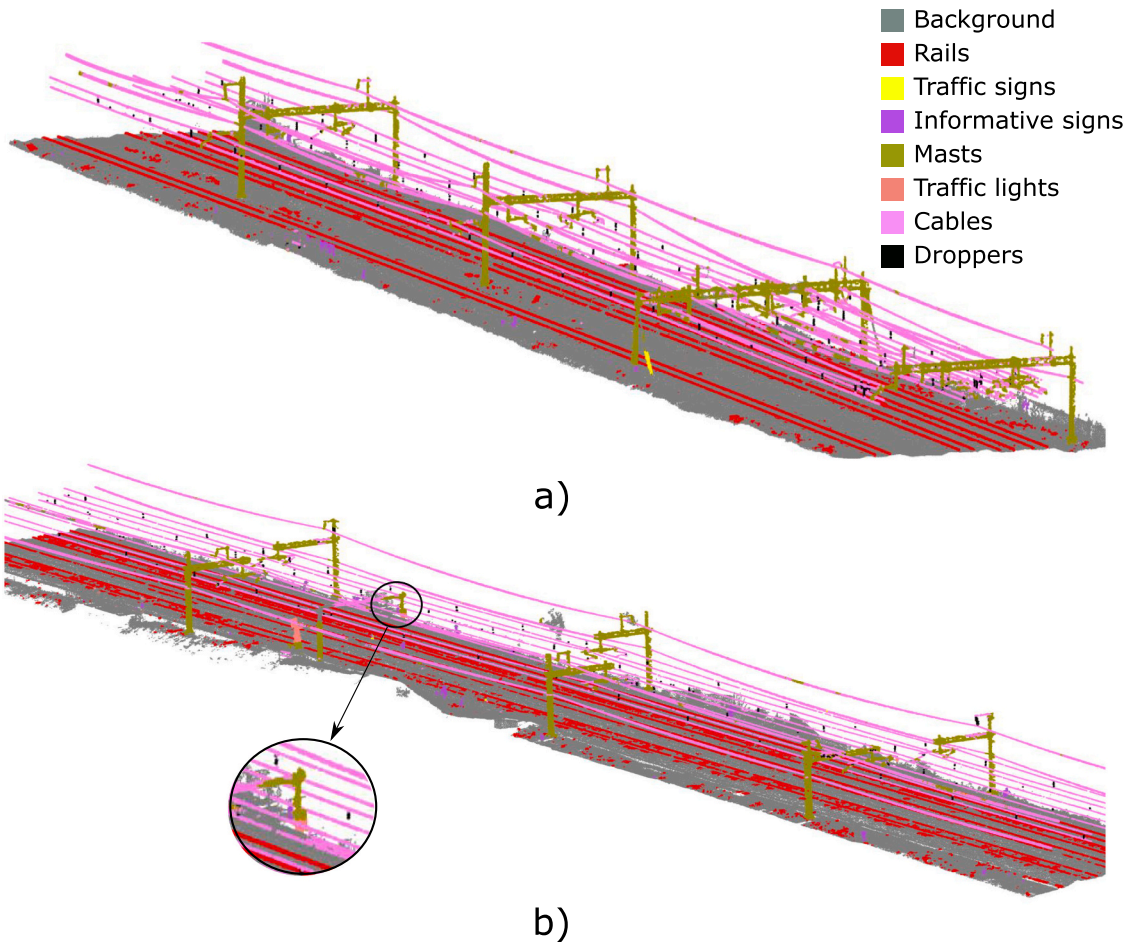


Fig. 12. Third scenarios results. These clouds belong to urban areas in a different country. Also, the geometry of the masts in (a) is different than most of the masts found in the other datasets.

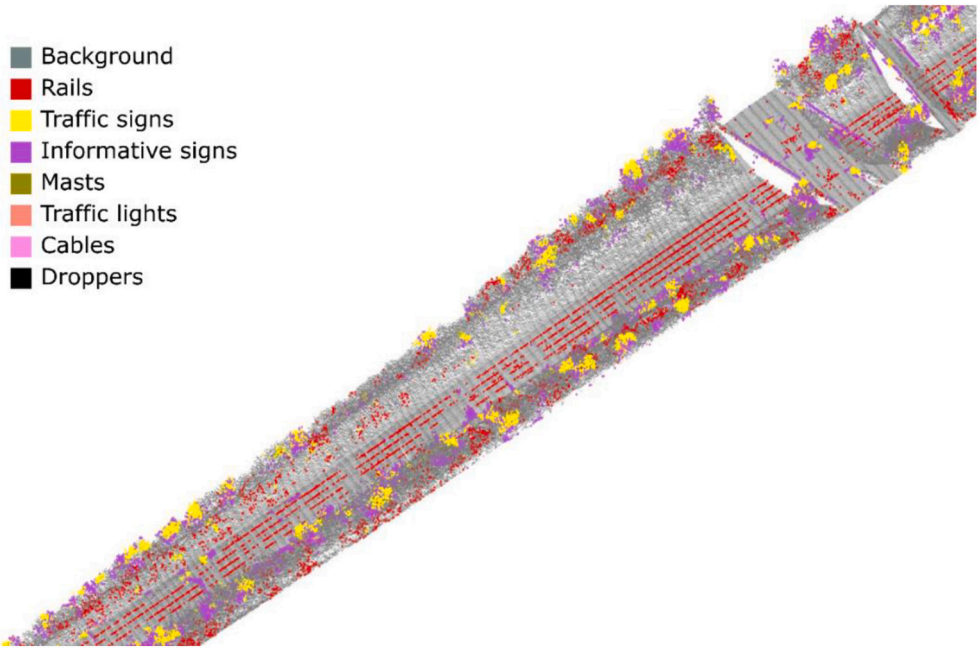


Fig. 13. Fourth scenario results. This is the point cloud with the lowest quality, being hard to segment manually. The only assets present in the point cloud are rails.

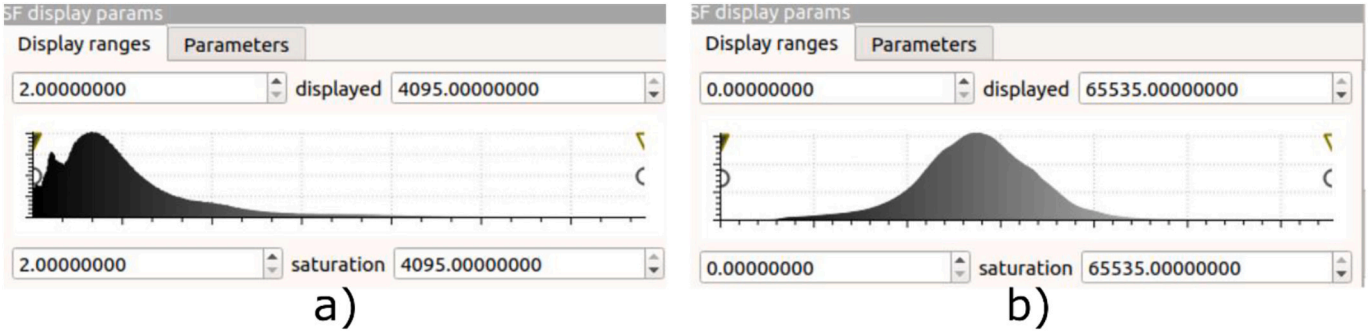


Fig. 14. Samples of intensity values distributions from first (a) and third (b) scenarios. The intensity values from the fist scenario follow log-normal distributions while the third scenario shows normal distributions.

is considered to perform best has the following characteristics: i) 32,768 input points, it provides results with better quality than the network that uses 16,384 input points, since the density of the output is twice as dense as the other one. ii) Simple architecture, because it provides better computational performance than the complex one, while maintaining similar metrics. iii) No intensity as input feature, although for the first scenario the intensity helps to improve the results, by not using the intensity, the network can be used also for the third scenario.

The mIoU obtained with the network categorized as the best is 74.89%. This value cannot be compared with railway segmentation works found in the literature, since in all cases, the metrics are calculated object-wise, instead of point-wise, like it is done in this work. However, compared to the mIoU presented in [28] for different architectures and benchmarks, the result is above the average.

Regarding object-wise metrics from Table 6, they show how the real performance of the methodology is similar to the heuristic method using to generate the training data, having better results with some assets such as rails, cables and traffic lights, while maintaining F1-Score above 90% for all cases.

As for as the training runtimes shown in Table 5, it can be observed that the difference between the simple and complex architecture is significative. And it can be even faster by reducing the input points of the network. It is also interesting that KPConv architecture runs almost as slow as the complex architecture, but providing worse results.

The well-functioning of the methodology for the first scenario is confirmed by the qualitative results studied. Point clouds with complex scenarios are tested, and all the objects present in them are correctly segmented. The decrease in the mIoU value is due to small noisy zones and point-wise discrepancies in object-background boundaries that affect to the metrics, but not to the quality of the results. In short, the methodology shows good performance in the test clouds of the first scenario.

Regarding the capability to generalize to new scenarios and sensors, the performance of the network decreases depending on the case. The results obtained in the second scenario show some noise that can be easily cleaned, so the methodology could be used to replace a manual segmentation. The third scenario shows more noise than the second, but, in general, the network is able to segment correctly most of the scene. Thus, it could be used as a semi-automatic method. Finally, the fourth scenario, due to the low quality of the point cloud, show very noisy results, but the rails are mostly segmented.

In summary, the methodology can be used for automated and semi-automated segmentation of new scenarios, since it shows very good performance, but some noise should be manually cleaned. As an alternative, increasing the training with more diverse scenarios would also improve the generalization capability, so the post-processing step could be eliminated.

6. Conclusions

This paper presents a methodology based in deep learning for automatic segmentation of the relevant assets from 3D point clouds in railway infrastructure. Different approaches are tested and compared to present a full pipeline that prepares the 3D point cloud to feed a neural network that outputs per point classification, so finally, fully segmented point clouds are obtained.

The results demonstrate how the method can correctly segment data captured in the same conditions as the data used to train the neural network, and it can also segment point clouds that have been captured in new environments, and with sensors of different characteristics. This method outperforms the current state of the art of semantic segmentation in railway environments by its generalization capability. These methods are mainly based on heuristics, and they have a high dependence on the homogeneity of the data, so it is hard to apply them in new environments, while deep learning methods allow to generalize to new environments that have not been seen during the training process of the neural network. Also, to the best of the authors' knowledge, this is the first approach where deep learning techniques are used in general railway environments working with point clouds.

Finally, the line of research presented in this work is promising for further study. The results obtained could serve as a basis to build as-is BIM models of the railway infrastructure. And also, the segmentation could be used to improve the maintenance of the infrastructure, comparing the 3D geometries of the same object at different time frames. Finally, it shows the utility of deep learning methods for railway infrastructure point clouds, that could be further exploited by designing specific methods to work in this particular environment.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 769255. This work has been partially supported by the Spanish Ministry of Science, Innovation and Universities through the project Ref. PID2019-108816RB-I00. The support of the Spanish Ministry of Science, Innovation, and Universities through the grant FPU20/01024 is acknowledged. Funding for open access charge: Universidade de Vigo/CISUG.

Declaration of Competing Interest

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

References

[1] Railway passenger transport statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php/Railway_passenger_transport_statistics_-_quarterly_and_annual_data, 2021.

[2] Passenger transport statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Passenger_transport_statistics&oldid=499254, 2021.

[3] Freight transport statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Freight_transport_statistics&oldid=496663, 2021.

[4] M.T. Baysari, A.S. McIntosh, J.R. Wilson, Understanding the human factors contribution to railway accidents and incidents in Australia, Accid. Anal. Prev. 40 (5) (2008) 1750–1757, https://doi.org/10.1016/J.AAP.2008.06.013.

[5] Q. Zhan, W. Zheng, B. Zhao, A hybrid human and organizational analysis method for railway accidents based on HFACS-railway accidents (HFACS-RAs), Saf. Sci. 91 (2017) 232–250, https://doi.org/10.1016/J.SSCI.2016.08.017.

[6] R. Gasparini, S. Pini, G. Borghi, G. Scaglione, S. Calderara, E. Fedeli, R. Cucchiara, Anomaly detection for vision-based railway inspection, in: Communications in Computer and Information Science, *1279 CCIS*, 2020, pp. 56–67, https://doi.org/10.1007/978-3-030-58462-7_5.

[7] X. Gibert, V.M. Patel, R. Chellappa, Deep multitask learning for railway track inspection, IEEE Trans. Intell. Transp. Syst. 18 (1) (2017) 153–164, https://doi.org/10.1109/TITS.2016.2568758.

[8] T. Lidén, Railway infrastructure maintenance - a survey of planning problems and conducted research, Transportat. Res. Procedia 10 (2015) 574–583, https://doi.org/10.1016/j.trpro.2015.09.011.

[9] O. Al-Bayari, Mobile mapping systems in civil engineering projects (case studies), Appl. Geomat. 11 (1) (2018) 1–13, https://doi.org/10.1007/S12518-018-0222-6.

[10] G.H. Kim, H.G. Sohn, Y.S. Song, Road infrastructure data acquisition using a vehicle-based mobile mapping system, Computer-Aided Civil Infrastruct. Eng. 21 (5) (2006) 346–356, https://doi.org/10.1111/j.1467-8667.2006.00441.x.

[11] G. Petrie, An introduction to the technology: Mobile mapping systems, Geoinformatics 13 (1) (2010) 32.

[12] P. Smith, BIM implementation – global strategies, Procedia Eng. 85 (2014) 482–492, https://doi.org/10.1016/J.PROENG.2014.10.575.

[13] A. Bradley, H. Li, R. Lark, S. Dunn, BIM for infrastructure: an overall review and constructor perspective, Autom. Constr. 71 (2016) 139–152, https://doi.org/10.1016/j.autcon.2016.08.019.

[14] R. Samimpay, E. Saghatforoush, Benefits of implementing building information modeling (BIM) in infrastructure projects, J. Eng. Proje. Product. Manag. 10 (2) (2020) 123–140, https://doi.org/10.2478/JEPPM-2020-0015.

[15] V. Vignali, E.M. Acerra, C. Lantieri, F. Di Vincenzo, G. Piacentini, S. Pancaldi, Building information modelling (BIM) application for an existing road infrastructure, Autom. Constr. 128 (2021), 103752, https://doi.org/10.1016/J.AUTCON.2021.103752.

[16] I. Puente, H. González-Jorge, J. Martínez-Sánchez, P. Arias, Review of mobile mapping and surveying technologies, Measurement 46 (7) (2013) 2127–2145, https://doi.org/10.1016/J.MEASUREMENT.2013.03.006.

[17] Ç. Aytekin, Y. Rezaeitabar, S. Dogru, I. Ulusoy, Railway fastener inspection by real-time machine vision, IEEE Transact. Syst. Man,Cybernet. Syst. 45 (7) (2015) 1101–1107, https://doi.org/10.1109/TSMC.2014.2388435.

[18] Y. Santur, M. Karaköse, E. Akin, A new rail inspection method based on deep learning using laser cameras, in: IDAP 2017 - International Artificial Intelligence and Data Processing Symposium, 2017, October 30, https://doi.org/10.1109/IDAP.2017.8090245.

[19] J. Zhong, Z. Liu, Z. Han, Y. Han, W. Zhang, A CNN-based defect inspection method for catenary Split pins in high-speed railway, IEEE Trans. Instrum. Meas. 68 (8) (2019) 2849–2860, https://doi.org/10.1109/TIM.2018.2871353.

[20] Q. Wang, M.K. Kim, Applications of 3D point cloud data in the construction industry: a fifteen-year review from 2004 to 2018, Adv. Eng. Inform. 39 (2019) 306–319, https://doi.org/10.1016/J.AEI.2019.02.007.

[21] V. Pətrəucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, C. Haas, State of research in automatic as-built modelling, Adv. Eng. Inform. 29 (2) (2015) 162–171, https://doi.org/10.1016/J.AEI.2015.01.001.

[22] P. Tang, D. Huber, B. Akinci, R. Lipman, A. Lytle, Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques, Autom. Constr. 19 (7) (2010) 829–843, https://doi.org/10.1016/J.AUTCON.2010.06.007.

[23] S. Chen, H. Liu, Z. Feng, C. Shen, P. Chen, Applicability of personal laser scanning in forestry inventory, PLoS One 14 (2) (2019), e0211392, https://doi.org/10.1371/JOURNAL.PONE.0211392.

[24] G.D. Pearse, J.P. Dash, H.J. Persson, M.S. Watt, Comparison of high-density LiDAR and satellite photogrammetry for forest inventory, ISPRS J. Photogramm. Remote Sens. 142 (2018) 257–267, https://doi.org/10.1016/J.ISPRSJPRS.2018.06.006.

[25] S. Gargoum, K. El-Basyouny, Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation, in: 2017 4th International Conference on Transportation Information and Safety, *ICTIS 2017 - Proceedings*, 2017, pp. 563–574, https://doi.org/10.1109/ICTIS.2017.8047822.

[26] H. Guan, J. Li, S. Cao, Y. Yu, Use of mobile LiDAR in road information inventory: a review, Int. J. Image Data Fusion Vol. 7 (3) (2016) 219–242. Taylor and Francis ltd, https://doi.org/10.1080/19479832.2016.1188860.

[27] M. Soilán, A. Sánchez-Rodríguez, P. del Río-Barral, C. Perez-Collazo, P. Arias, B. Riveiro, Review of laser scanning technologies and their applications for road and railway infrastructure monitoring, Infrastructures 4 (4) (2019) 58, https://doi.org/10.3390/infrastructures4040058.

[28] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, Deep learning for 3D point clouds: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2020) 4338–4364, https://doi.org/10.1109/tpami.2020.3005434.

[29] S. Oude Elberink, K. Khoshelham, M. Arastounia, D. Diaz Benito, Rail track detection and modelling in Mobile laser scanner data, ISPRS Annals Photogr. Remote Sens. Spatial Informat. Sci. II-5/W2(5W2) (2013) 223–228, https://doi.org/10.5194/isprsannals-II-5-W2-223-2013.

[30] M. Soilán, A. Justo, A. Sánchez-Rodríguez, B. Riveiro, 3D point cloud to BIM: semi-automated framework to define IFC alignment entities from MLS-acquired LiDAR data of highway roads, Remote Sens. 12 (14) (2020) 2301, https://doi.org/10.3390/rs12142301.

[31] M. Arastounia, Automated recognition of railroad infrastructure in rural areas from LIDAR data, Remote Sens. 7 (11) (2015) 14916–14938, https://doi.org/10.3390/rs71114916.

[32] D. Lamas, M. Soilán, J. Grandío, B. Riveiro, Automatic point cloud semantic segmentation of complex railway environments, Remote Sens. 13 (12) (2021) 2332, https://doi.org/10.3390/RS13122332.

[33] A. Sánchez-Rodríguez, B. Riveiro, M. Soilán, L.M. González-deSantos, Automated detection and decomposition of railway tunnels from Mobile laser scanning datasets, Autom. Constr. 96 (2018) 171–179, https://doi.org/10.1016/j.autcon.2018.09.014.

[34] M. Soilán, A. Nóvoa, A. Sánchez-Rodríguez, B. Riveiro, P. Arias, Semantic segmentation of point clouds with pointnet and kpconv architectures applied to railway tunnels, in: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial* Information Sciences, V-2–2020(2), 2020, pp. 281–288, https://doi.org/10.5194/isprs-annals-V-2-2020-281-2020.

[35] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, *CVPR 2017*, *2017-Janua*, 2017, pp. 77–85, https://doi.org/10.1109/CVPR.2017.16.

[36] H. Thomas, C.R. Qi, J.E. Deschaud, B. Marcotegui, F. Goulette, L. Guibas, KPConv: Flexible and deformable convolution for point clouds, in: Proceedings of the IEEE International Conference on Computer Vision, 2019-Octob, 2019, pp. 6410–6419, https://doi.org/10.1109/ICCV.2019.00651.

[37] X. Giben, V.M. Patel, R. Chellappa, Material classification and semantic segmentation of railway track images with deep convolutional neural networks, in: Proceedings - International Conference on Image Processing, *ICIP*, *2015-December*, 2015, pp. 621–625, https://doi.org/10.1109/ICIP.2015.7350873.

[38] F.J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F.S. Khan, M. Felsberg, Deep projective 3D semantic segmentation, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), *10424 LNCS*, 2017, pp. 95–107, https://doi.org/10.1007/978-3-319-64689-3_8.

[39] H. Jing, S. You, Point cloud labeling using 3D convolutional neural network, in: Proceedings - International Conference on Pattern Recognition 0, 2016, pp. 2670–2675, https://doi.org/10.1109/ICPR.2016.7900038.

[40] C. Choy, J. Gwak, S. Savarese, 4D Spatio-temporal ConvNets: Minkowski convolutional neural networks. https://github.com/StanfordVL/MinkowskiEngine, 2019.

[41] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, in: *Advances in Neural Information Processing Systems*, *2017-Decem*, 2017, pp. 5100–5109.

[42] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, Randla-net: efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 11105–11114, https://doi.org/10.1109/CVPR42600.2020.01112.

[43] S. Wang, S. Suo, W.C. Ma, A. Pokrovsky, R. Urtasun, Deep parametric continuous convolutional neural networks, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 2589–2597, https://doi.org/10.1109/CVPR.2018.00274.

[44] F. Engelmann, T. Kontogianni, A. Hermans, B. Leibe, Exploring spatial context for 3D semantic segmentation of point clouds, in: Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, *ICCVW 2017*, *2018-January*, 2017, pp. 716–724, https://doi.org/10.1109/ICCVW.2017.90.

[45] Q. Huang, W. Wang, U. Neumann, Recurrent slice networks for 3D segmentation of point clouds, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 2626–2635, https://doi.org/10.1109/CVPR.2018.00278.

[46] L. Landrieu, M. Simonovsky, Large-scale point cloud semantic segmentation with Superpoint graphs, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 4558–4567, https://doi.org/10.1109/CVPR.2018.00479.

[47] L. Wang, Y. Huang, Y. Hou, S. Zhang, J. Shan, Graph attention convolution for point cloud semantic segmentation, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 2019, pp. 10288–10297, https://doi.org/10.1109/CVPR.2019.01054.

[48] W. Sun, Z. Zhang, J. Huang, RobNet: real-time road-object 3D point cloud segmentation based on SqueezeNet and cyclic CRF, Soft. Comput. 24 (8) (2020) 5805–5818, https://doi.org/10.1007/S00500-019-04355-Y/FIGURES/15.

[49] B. Wu, A. Wan, X. Yue, K. Keutzer, SqueezeSeg: convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud, in: Proceedings - IEEE International Conference on Robotics and Automation, 2018, pp. 1887–1893, https://doi.org/10.1109/ICRA.2018.8462926.

[50] B. Wu, X. Zhou, S. Zhao, X. Yue, K. Keutzer, SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a

LiDAR point cloud, in: Proceedings - IEEE International Conference on Robotics and Automation, 2019-May, 2019, pp. 4376–4382, https://doi.org/10.1109/ICRA.2019.8793495.
[51] CloudCompare (version 2.12.2) [GPL software]. http://www.cloudcompare.org/, 2022.
[52] Teledyne Optech. https://www.teledyneoptech.com/en/home/, 2021.
[53] Ingenieria Insitu. https://ingenieriainsitu.com/, 2021.
[54] RIEGL, RIEGL Laser Measurement Systems. http://www.riegl.com/, 2022.
[55] S. Ghosh, N. Das, I. Das, U. Maulik, Understanding deep learning techniques for image segmentation, ACM Comput. Surv. 52 (4) (2019), https://doi.org/10.1145/3329784.
[56] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, M. Nießner, ScanNet: Richly-annotated 3D reconstructions of indoor scenes, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, *CVPR 2017*, *2017-January*, 2017, pp. 2432–2443, https://doi.org/10.1109/CVPR.2017.261.
[57] T. Chaton, N. Chaulet, S. Horache, L. Landrieu, Torch-Points3D: A modular multi-task framework for reproducible deep learning on 3D point clouds, in: 2020 International Conference on 3D Vision (3DV), 2020, pp. 1–10, https://doi.org/10.1109/3DV50981.2020.00029.
[58] R. Indraswari, T. Kurita, A.Z. Arifin, N. Suciati, E.R. Astuti, Multi-projection deep learning network for segmentation of 3D medical images, Pattern Recogn. Lett. 125 (2019) 791–797, https://doi.org/10.1016/j.patrec.2019.08.003.
[59] M. Everingham, S.M.A. Eslami, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The Pascal visual object classes challenge: a retrospective, Int. J. Comput. Vis. 111 (1) (2015) 98–136, https://doi.org/10.1007/S11263-014-0733-5/FIGURES/27.