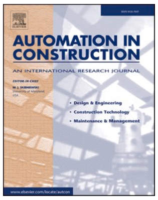
Automation in Construction 150 (2023) 104854

Contents lists available at ScienceDirect

Automation in Construction

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



Multimodal deep learning for point cloud panoptic segmentation of railway environments

Javier Grandio [*], Belen Riveiro, Daniel Lamas, Pedro Arias

CINTECX, Universidade de Vigo, GeoTECH Group, Campus Universitario de Vigo, As Lagoas, Marcosende, 36310 Vigo, Spain

ARTICLE INFO

Keywords:
Deep learning
Point clouds
Multimodal
Panoptic segmentation
Railway infrastructure

ABSTRACT

The demand for transportation asset digitalisation has significantly increased over the years. For this purpose, mobile mapping systems (MMSs) are among the most popular technologies that allow capturing high precision three-dimensional point clouds of the infrastructure. In this paper, a multimodal deep learning methodology is presented for panoptic segmentation of the railway infrastructure. The methodology takes advantage of image rasterisation of the point clouds to perform a rough segmentation and discard more than 80% of points that are not relevant to the infrastructure. With this approach, the computational requirements for processing the remaining point cloud are highly reduced, allowing the process of dense point clouds in short periods of time. A 90 km-long railway scenario was used for training and testing. The proposed methodology is two times faster than the current state-of-the-art for the same point cloud density, and pole-like object segmentation metrics are improved.

1. Introduction

Owing to the current degree of globalisation, transport infrastructure is essential in modern society. In particular, railroads are among the main transportation modes for goods and people [1,2]. Therefore, the preservation of railway infrastructure has a direct impact on citizens’ quality of life. To ensure the proper functioning of this infrastructure, it is necessary to perform maintenance operations that may involve predictive or corrective decision-making [3–6]. However, the large” scale of railway infrastructure and its distance to urban areas are barriers to carrying out maintenance in a cost-effective and efficient manner [7,8].

The emergence of digitalisation technologies can ease predictive maintenance as they enable automated or semi-automated monitoring of infrastructure assets [9]. This automation allows performing the tasks more productively and securely [10,11], preventing possible accidents, and reducing the operation times due to inspection operations.

For the digitalisation of infrastructure, building information modelling (BIM) is among the most widely used solutions, which helps improve the efficiency and integration of the information of large construction projects [12,13]. Furthermore, there are examples in the literature that promote BIM applications in railway infrastructure [14–16]. However, a major difficulty in creating as-is BIM models of the existing infrastructure is obtaining the required data [17,18].

To do this, mobile mapping systems (MMSs) are a technology that allows recording 3D geometric and radiometric data from built transport infrastructures in short periods of time, generating a massive amount of information [19–21]. These systems may be equipped with several types of sensors to capture data from different sources. In particular, light detection and ranging (LiDAR) sensors are useful for recording the geometry of infrastructures. LiDAR technology allows the capture of 3D data of the environment with high accuracy and speed [22–24], presenting the data as unorganised 3D point clouds. A point cloud is defined as a series of points in a 3D coordinate system that represents the surfaces of the objects around the scanning device. The recorded points can also provide information about additional attributes such as the intensity attribute, which represents the reflectance where a point is found, colour of the surface, number of returns of the laser, and other fields related to the capture, such as the sensor angle and timestamp.

The raw 3D point clouds recorded by LiDAR sensors do not include any semantic information; thus, the point clouds must be segmented to identify and characterise the assets under study. Segmentation can be semantic or by instance. While semantic segmentation aims to assign categorical labels to the individual points that constitute a point cloud [25], instance segmentation assigns different labels for separate

[*] Corresponding author.
E-mail addresses: javier.grandio.gonzalez@uvigo.es (J. Grandio), belenriveiro@uvigo.es (B. Riveiro), daniel.lamas.novoa@uvigo.es (D. Lamas), parias@uvigo.es (P. Arias).

https://doi.org/10.1016/j.autcon.2023.104854
Received 29 August 2022; Received in revised form 15 March 2023; Accepted 28 March 2023
Available online 6 April 2023

instances of objects belonging to the same class [26]. Kirillov et al. [27] referred to panoptic segmentation as a task that combines both semantic and instance segmentation.

Segmentation tasks on point clouds are rather heavy [28], and because of the massive nature of railway infrastructure, it is not viable to do it manually. Consequently, automatic algorithms for point cloud segmentation (both semantic and instance) are essential for the digitalisation of the infrastructure.

The use of deep learning methods as a solution for automatic point-cloud processing has grown over the years. These methods have mainly been used for classification, segmentation, and object detection. Regarding segmentation tasks, these methods can be divided into [29] 1) projection-based methods, 2) discretisation-based methods, 3) point-wise methods, and 4) hybrid methods.

Regarding the automatic segmentation of railway infrastructure point clouds, two different trends have been found in the literature.

The first trend relies on heuristic algorithms that are designed to detect the most relevant infrastructure assets. With this objective, Oude et al. [30] presented a method that can detect the rail tracks in the infrastructure. In addition, a more complete work was presented by Arastounia, where the author also segmented cables, masts, and cantilevers from a 550 m long rural railway line [31]. As an improvement, in our previous work [32], we presented a robust methodology applied to a 90 km long railway track. The methodology was used to segment pole-like and linear objects. Signs and masts are examples of such pole-like objects. In contrast, linear objects include rails and cables.

The second trend for semantic segmentation of railway infrastructure is based on the use of deep learning techniques. This is yet to be established, but it has shown promising results. In [33], the authors presented a projection-based method for segmenting railway tracks. A more specific application was presented in [34], where the authors used Pointnet++ [35] for ballast railway fastener inspections. Subsequently, in [36], the neural networks Pointnet [37] and KPConv [38] were used to segment railway tunnels. In our previous work [39], we proposed a modified version of Pointnet++ to segment all relevant objects found in a railway infrastructure semantically. Finally, Eickeler et al. [40] trained KPConv using existing CAD data to improve the semantic segmentation results in railway environments.

Despite the promising results presented in the literature, all these deep learning methods segment the point clouds semantically. However, it would be interesting to achieve instance segmentation for pole-like objects, such as masts, signs, and traffic lights. This improvement would ease the digitalisation of infrastructure. In addition, because of the massive nature of railway infrastructure, point clouds contain millions of points, so they need to be subsampled and cropped to feed the neural networks owing to computational limitations. In addition, the amount of data increases processing runtimes.

As a follow-up to these works, intending to overcome the drawbacks previously presented, this paper proposes a high performance deep learning methodology based on a multimodal approach that merges projection-based methods with point-wise methods to obtain panoptic segmentation of the railway environment. The main contributions of this study are as follows.

1. It offers panoptic segmentation. This is achieved by instance segmentation for pole-like objects while preserving the semantic segmentation of linear assets.
2. It provides results with the original point-cloud density to preserve the details of the objects for further processing. In addition, it processes point clouds with the same point density faster than other methods in the literature.

The remainder of this paper is organised as follows. Section 2 presents the case study, Section 3 describes the methodology and the steps followed to obtain the panoptic segmentation, Section 4 presents the results obtained using the methodology, and Section 5 discusses the results. Finally, the conclusions of this study are presented in Section 6.

2. Case study

The scenario used for this study consists of a 90 km railway track, as shown in Fig. 1. This scenario was presented in [41]. The survey was conducted using a LYNX Mobile Mapper by Optech [42]. Two LiDAR sensors equipped on an MMS provided an average point cloud density of 980 points/m^2 and a range precision of 5 mm. The dataset was divided into 450 georeferenced point clouds, each 200 m in length. The total dataset contained more than 2 billion points. The points were characterised by their Euclidian position, intensity, scan angle, number of returns, and GPS time.

The results obtained in [32] using the heuristic methodology were used as the ground truth for the classification attribute. This methodology can produce misclassifications; however, the misclassifications have been studied in depth, and in most cases, they are small and isolated errors that could also be present in the case of manual classification.

2.1. Assets to be detected

The assets that comprise the railway infrastructure can be divided into pole-like and linear assets. Pole-like assets are placed at discrete locations along the railroad, and those detected using this methodology are as follows:

- **Informative signs**. Small signs are used to show the kilometres of the railway where the signs are found. An example is shown in Fig. 2 (a).
- **Masts**. An example is shown in Fig. 2 (b).
- **Traffic lights**. An example is shown in Fig. 2 (c).
- **Traffic signs**. Speed restriction and related signs, they have high-intensity values. An example of this is shown in Fig. 2 (d).

The linear assets to detect are the following:

- **Cables**. Includes all the cables except for the droppers.
- **Droppers**. These are vertical structural wires that join the catenary with the contact wires. The cables and droppers are shown in Fig. 2 (e).
- **Rails**. Includes all rails. Shown in Fig. 2 (f).

Finally, another label denominated **background** represents all the points that do not belong to any of the classes described above.

3. Methodology

This section presents a methodology designed for the panoptic segmentation of point clouds from the railway infrastructure. The proposed method is end-to-end, having raw point clouds as input, and returning classification values for the individual points of the clouds. The methodology is based on a multimodal approach that relies on image and point cloud data. A summary of these steps is shown in Fig. 3.

This methodology breaks down the task into simpler steps, considerably reducing the computational complexity of the work while maintaining the high quality of the original point cloud.

The method starts with a previous segmentation of the railway tracks. The track includes the ground, rails, ballasts, and sleepers. This topic is not addressed in this work because it has been studied by many researchers in the literature [43–46]; in this case, we follow the approach previously presented by Lamas et al. [32] because it has been tested in the same railway environment.

The track-segmentation approach is based on voxelisation. Considering an adequate voxel size, it can be assumed that voxels that belong to the track do not have neighbours under or over them. Consequently,

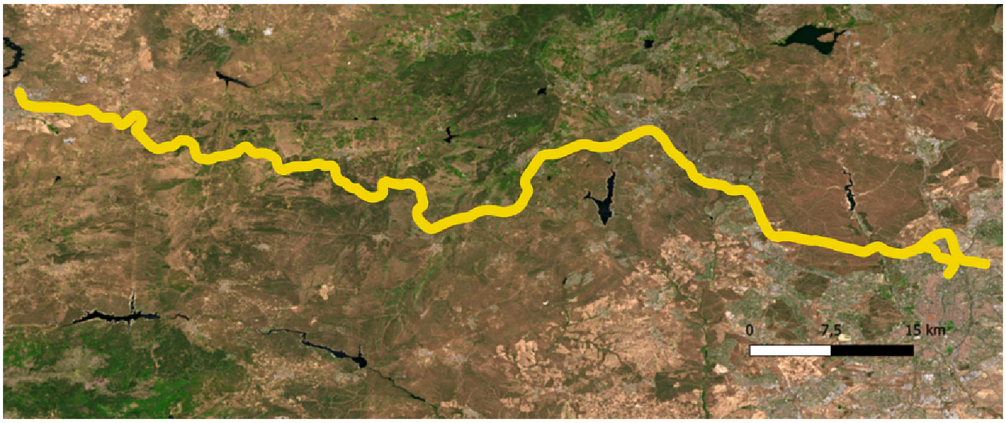


Fig. 1. Railway infrastructure surveyed for the case study.

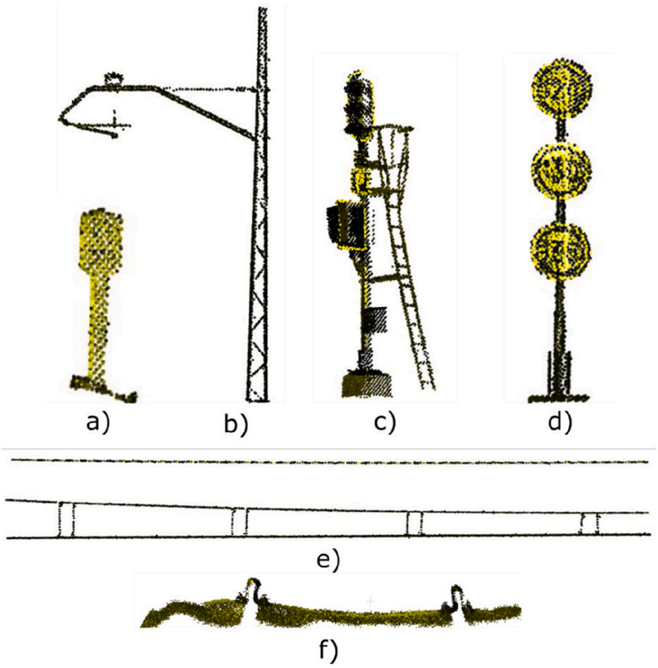


Fig. 2. Assets to be detected.

voxels that fulfil this requirement and are under the trajectory followed by the surveying vehicle are considered track points.

The first step of the methodology involves generating raster images [47] from the original point clouds to reduce the computational complexity of the task. These images are used to segment rails and cables and identify regions of interest (ROIs) where pole-like objects may be found. Image data provide a simpler representation of the point cloud, but in exchange geometrical characteristics can be lost. However, cables and rails were identified in the images, and ROIs for pole-like objects were also generated.

The second part of the methodology separately processes rails, cables, and pole-like assets. First, the ROIs were generated from the raster images. The points belonging to the ROI pixels were loaded, and sub-point clouds were generated for the individual ROIs. Finally, the sub-point clouds were classified using Pointnet++ [35] to identify the type of object to which they belong. Because the ROIs are just a small percentage of the original clouds, the computational complexity was significantly reduced compared with working directly with the entire raw point cloud. Regarding the rails, points of the track that belong to pixels classified as rails were classified as a given asset. Finally, points that did not belong to the track and were identified as cables were retrieved. The points were then segmented to remove noise and separate droppers.

The results obtained for rails, cables, and pole-like assets were combined to obtain the point cloud segmentation.

3.1. Image processing

Each of the individual point clouds that constitute the dataset comprises an average of 4.5 million points. This amount of data is too large to feed directly to a neural network owing to computational limitations; therefore, the point clouds need to be broken down into smaller point clouds and subsampled. In addition, the number of points increases the runtime of the task. However, in the case of railway infrastructure, surroundings irrelevant to the infrastructure accounted for more than

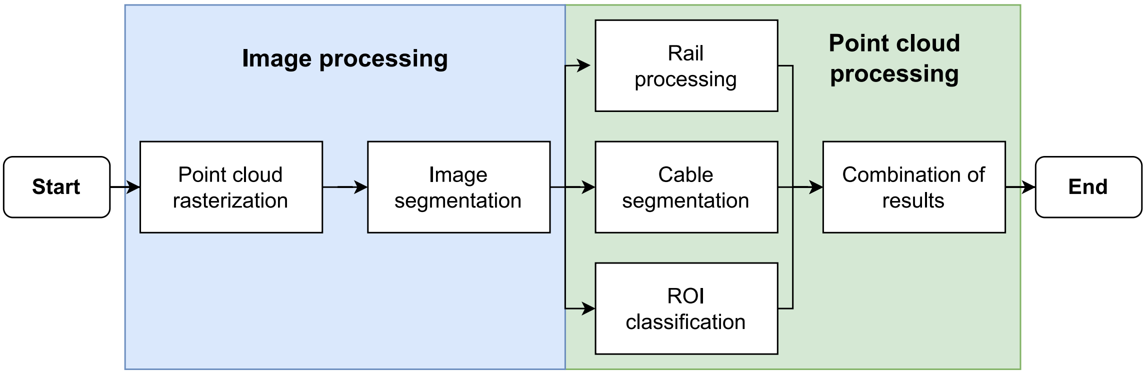


Fig. 3. Summary of the methodology.

80% of the points.

Considering this, the methodology used 2D raster images for the first processing step. By using images instead of the original point clouds, the computational complexity was significantly reduced, and several point clouds could be processed simultaneously. This step removed most of the unnecessary data and achieved a faster processing. Finally, the final segmentation of the rails was obtained.

3.1.1. Raster generation

The input data for the method is a raw point cloud, $\boldsymbol{P}_{Nx4} = [\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{I}]$, where N is the number of input points contained in the point cloud, ($\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$) represents the Euclidian coordinates, and $\boldsymbol{I}$ corresponds to the intensity values of the points. Because the track was segmented in a prior step, the points belonging to the track were denominated $\boldsymbol{P_t}$, and the rest were $\boldsymbol{P_{nt}}$. The point clouds were converted into image data by rastering them. The rasters were generated based on the top perspectives of the point clouds to generate horizontal views. Both the points belonging to the track and others were used to generate two individual images. Two different images were created instead of one because it is a simple operation, and a better representation of the original point cloud was obtained.

To preserve the information of the original point cloud, a new vector $\boldsymbol{R}_{Nx1}$ was created when building raster images. This vector $\boldsymbol{R}$ indicates the pixel of the raster image where each of the original points is found. Thus, the information obtained from the raster images could be applied directly to the original point cloud.

The generated raster images contained three channels, each of which represented certain characteristics of the point cloud.

- The first channel of the image was based on the **intensity** values of the points. The values of pixels I_p were calculated as the average intensity values of all the points contained in the given pixel.

$$I_p = \frac{\sum_{i=0}^{n} I_i}{n}$$

- The second channel provided information regarding the **density of points on the z-axis**. The pixel values were calculated as the number of points from the original point cloud contained in each pixel. Consequently, the pixel values tend to be high when vertical objects are found.

$$\rho_p = n$$

- The last channel provided information regarding the **maximum height on the z-axis**. The pixel values were assigned with the z-coordinate of the highest point found in them. This field helps visualise objects above the ground, such as cables.

$$h_p = z_{max}$$

An example of the raster image generated is presented in Fig. 4. While some objects such as masts, rails, and cables are easily interpretable by humans, smaller objects, such as informative signs, are not.

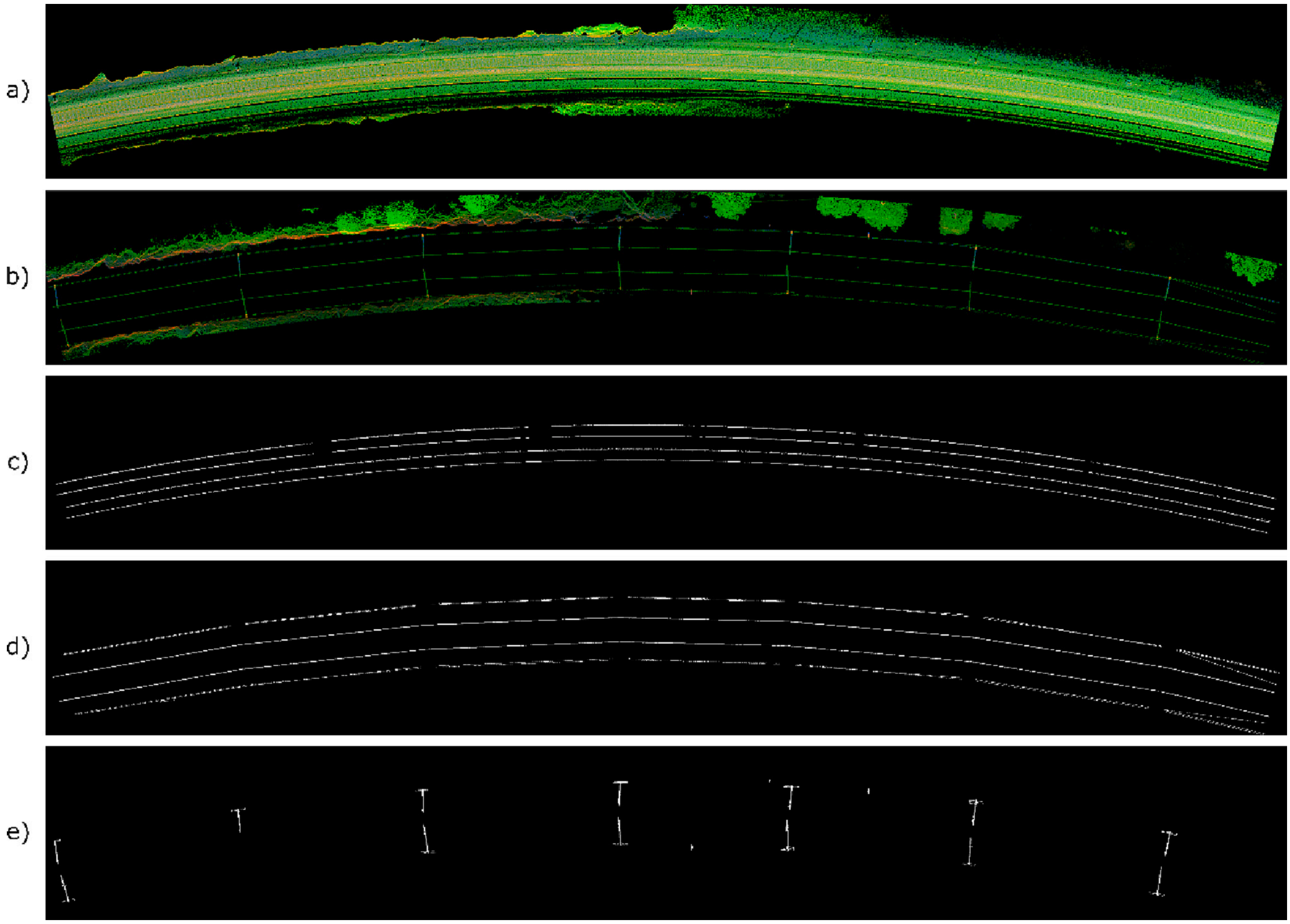


Fig. 4. Images generated from a point cloud rasterised according to the parameters listed in Table 2. (a) Track image raster that is fed to the neural network; (b) no track raster fed to the neural network; (c) labels for rail prediction, (d) labels for cable predictions; and (e) labels for the ROIs.

However, even when not interpretable by humans, convolutional neural networks (CNNs) have proven to work well in detecting them.

These images are useful for segmenting rails, cables, and ROIs where pole-like objects may be found. However, these ROIs may overlap with rails and cables along the **z**-axis. To avoid this issue, the segmentation task was divided into three steps: 1) ROI segmentation, 2) rail segmentation, and 3) cable segmentation. Consequently, along with the raster images, three different mask images were also created as the ground truth, each of which corresponded to an asset to study. These images had a single binary channel, with zeros in the pixels where no assets were present and ones in the pixels that contained points that belonged to any of the assets. An example of a mask generated for a point cloud is found in Fig. 4 (c), (d), and (e).

3.1.2. Image segmentation

Once the images were available, they were segmented to remove their surroundings. It is well known that CNNs achieve state-of-the-art results for computer vision tasks [48]. Consequently, the task of segmenting ROIs, rails, and cables was delegated to neural networks. U-Net [49] is one of the best-known classical CNN architectures for semantic segmentation, and many state-of-the-art semantic segmentation CNNs are based on U-shaped architectures [50]. Consequently, this type of architecture was designed for the task.

The input of the neural network includes raster images of both the track (with its three channels) and the remaining points (with three channels). Consequently, the input of the neural network must have six channels, and thus, the input dimensions are $(Image\ height, image\ width, 6)$. Taking all the information presented into account, the architecture of the neural network designed for the task is shown in Fig. 5.

One raster image was created for each of the 200 m-long point clouds of the case study. Because only 450 point clouds are available and 20% of them are left out for testing, the number of training images is lower than 400, which is a considerably low quantity for CNN image segmentation training. This issue was addressed by performing data augmentation during training. The preprocessing steps applied to the training images were as follows:

- **Rescale values:** The pixel values were rescaled to (0,1). In addition, to eliminate possible outliers in the number of points and height rasters, the maximum values of the pixels were limited to the value of the 99th percentile of each point cloud.
- **Noise:** Each time an image was sampled for training, random noise with a standard deviation of 0.05 was added to each channel of the image.
- **Random Flip:** Each time an image was sampled, it had a 50% probability of being flipped horizontally. This allowed for greater diversity in the training data.

Because most of the image labels belong to the surroundings, the CNN may be biased to label all points as the background. To solve this problem, two approaches were used for the loss function. The first was using Jaccard loss. However, with this loss, informative signs were discarded in most cases. The second solution consisted of using binary cross-entropy loss with weights to increase the loss when points belonging to ROIs were misclassified. The weights were calculated using the following equation:

$$w_i = \frac{1}{log\left(1.2 + \frac{n_i}{N}\right)}$$

where w_i is the weight of class i, n_i is the number of pixels in the image belonging to class i, and N is the total number of pixels in the image.

3.2. Point cloud processing

The prediction masks generated by the trained CNN removed the surroundings of the infrastructure; therefore, these points did not require further processing. However, the segmentation of rails, cables, and ROIs was given on the 2D raster image plane; therefore, further processing was required to provide the final results.

Owing to the differences between the three types of assets, they were processed individually, and the results were combined to achieve the final panoptic.

3.2.1. Rails

In the case of rails, the implementation is rather simple, as shown in

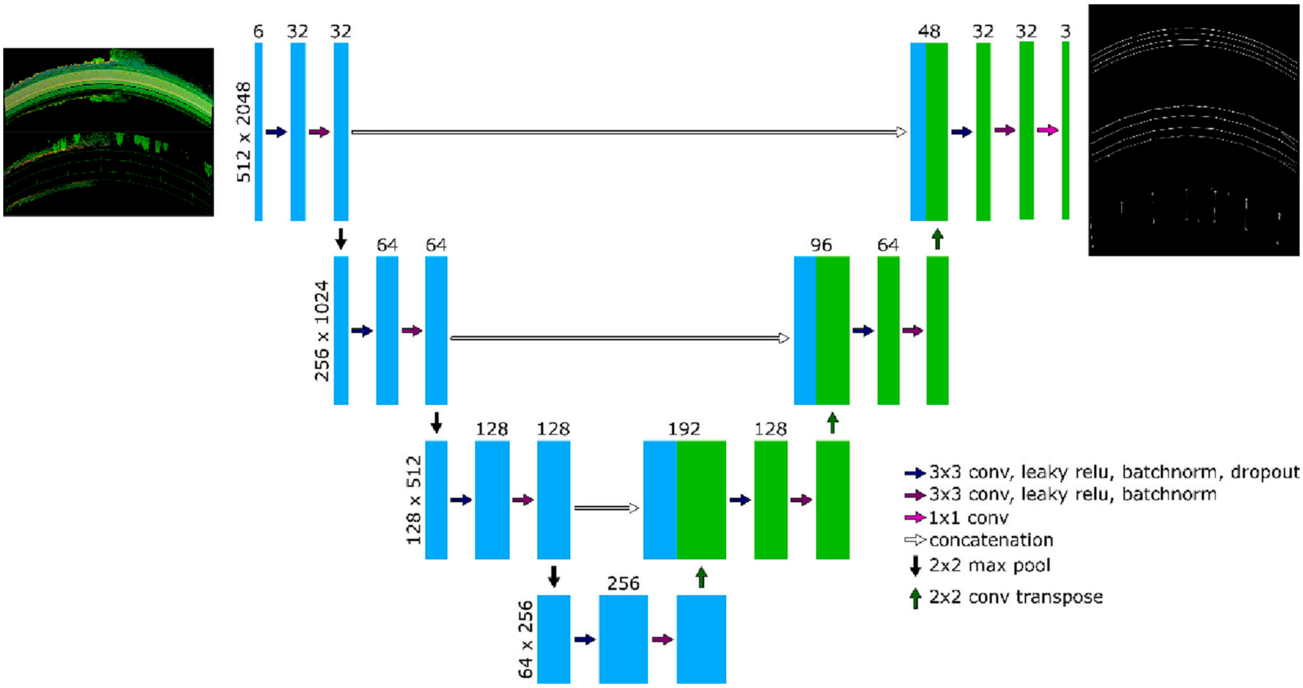


Fig. 5. Neural network architecture used for image segmentation.

Fig. 6.

Because the track of the point clouds had been previously segmented using the methodology of Lamas et al. [32], the points that belonged to rail pixels and track $\boldsymbol{P_t}$ simultaneously were segmented as rails. This condition was used to ensure that only points that belonged to the track were considered, whereas points that overlapped in the **z**-axis with the rails were not incorrectly classified as rails. Fig. 7 shows an example of points retrieved as rails in a random point cloud.

3.2.2. Cables

Cable postprocessing involves two tasks: i) recovering points that have been labelled as cables by the CNN, and ii) separating droppers and noise from the rest of the cables. A summary of the procedure is shown in Fig. 8.

For this process, the first step consisted of taking only the points tagged as cables that were above the ground $\boldsymbol{P_{nt}}$, and an example of this is shown in Fig. 9. As depicted in the image, the resulting point cloud was populated by noisy areas that overlapped with the cables. To remove this noise and segment droppers as their own classes, two approaches have been proposed.

The first approach consists of voxelising the point cloud and calculating voxel dispersion along the **z-axis**. Because droppers are known to have a vertical geometry, voxels with high dispersion values on the **z**-axis are labelled as droppers. This approach provides acceptable results and is simple. However, the results were partially noisy. In addition, some points that may overlap with real cables were not removed.

To overcome this drawback, a semantic segmentation neural network is proposed. The neural network segmented the points into three classes: cables, droppers, and noise. Thus, the droppers were properly identified, and the noise could be removed. An architecture based on Pointnet ++ [35] was adopted to perform segmentation. This architecture was selected over others, such as KPConv [38] or Point Transformer [51] because these provide similar results but longer runtimes.

The number of points classified by the CNN as cables was just a small percentage compared with the original point cloud; therefore, the workload of the neural network was highly reduced, and whole point clouds could be processed simultaneously. In addition, computational complexity can be further reduced by applying voxelisation prior to segmentation. For this study, a voxelisation of 0.11 m was proposed. This value was selected because, with this voxelisation, the point clouds did not lose geometrical information. In addition, 65,536 input points were fed to the network because, in all cases, the total number of remaining points after voxelisation was below that value, and a lower number would leave out points in some cases.

To train the neural network used to segment cables, it is necessary to generate training data. These training data were obtained from the cable mask CNN training dataset. The following preprocessing steps were applied to the data:

- **Scale coordinates**. The coordinates were centred and scaled to values of (0,4). These values were taken because the range on the **z**-axis was smaller than the range on **x** and **y**, and hence other transformations would result in excessively small values on the **z**-axis.
- **Random rotation.** Random rotation along the **z**-axis was applied before feeding the network.
- **Fixed number of points.** For this, the points were resampled until 65,536 input points were reached. The resampled points were ignored when computing the loss during training.

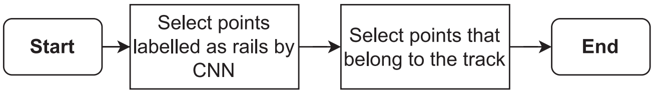

Start
Select points labelled as rails by CNN
Select points that belong to the track
End

Fig. 6. Rail extraction diagram.

In addition, **a** weighted loss function was applied for training, increasing the loss values when the least-populated classes were misclassified. The approach adopted in [35] was used for this purpose.

Finally, the cross-entropy loss function and Adam optimiser were used with the following hyperparameters: learning rate = 0.001 and batch normalisation momentum = 0.1.

3.3. Regions of interest

The ROIs predicted by the CNN were used to generate new subpoint cloud candidates to be pole-like objects. Consequently, these new subpoint clouds must be classified to assign them a pole-like object class or remove them. A summary of ROI processing is shown in Fig. 10.

First, subpoint clouds were created from the ROIs. For this, points that did not belong to track $\boldsymbol{P_{nt}}$ and had been predicted as ROIs were obtained. An example of the subpoint clouds obtained from the ROIs of a full-point cloud is shown in Fig. 11. This figure shows how ROIs are spatially distant and easy to separate. In addition, small sections of cables that overlapped with ROIs were retrieved.

Because the objective is to achieve panoptic segmentation, single objects must be separated. Spatial clustering algorithms are well-fitted for this task. In this case, DBSCAN [52] clustering with a minimum distance of 0.4 m and a minimum number of three points was applied. These parameters were selected considering the size of railway assets and the separation between them. Thus, individual subpoint clouds were generated.

3.3.1. Classification of regions of interest

Once the individual subpoint clouds are available, they must be classified. For this purpose, an artificial neural network with an architecture based on Pointnet++ [35] was proposed. In all cases, these networks operate only with the coordinates of the point cloud, leaving aside the other attributes available in .las data files.

As was done when training the CNN, the labels from the heuristic method [32] were used to generate the training data. Thus, point clouds containing individual objects were generated. These point clouds were also manually verified to remove possible misclassifications from the heuristic method. In addition, a new class, called noise, was added. This new class was created to remove possible errors from the image segmentation. Therefore, in the case of noisy data, such as parts of trees, bushes, or other noise that overlapped with real objects, they were ignored.

Because there were no samples of noise available, the misclassifications from the heuristic methods were labelled as noise. In addition, to add more variability to the dataset, a version of the image-segmentation CNN was applied to the training data, which generated new ROIs for training. The ROIs obtained from the CNN were clustered and labelled according to the heuristic results. Some data were duplicated using the boundaries of the CNN when choosing the ROIs, and objects misclassified by the CNN were added to the noise class.

The data available for the training are listed in Table 1. As can be observed, there was a predominant class, the masts, whereas the rest of the classes contained fewer samples.

The first approach to solve this problem is to use weighted loss. This technique is not sufficient for solving unbalanced data issues. Consequently, a data augmentation approach was adopted. The second approach consists of duplicating less-frequent objects. These point clouds were not only duplicated, but some geometric transformations were also applied.

- **Noise:** Random normal noise with a standard deviation of 0.05 was applied to all the coordinates in the three axes.
- **Random Rotation:** The clouds were randomly rotated on the **z**-axis and rotated at a random angle of up to 15° on the **x** and **y** axes.

Although these transformations are rather simple, they were

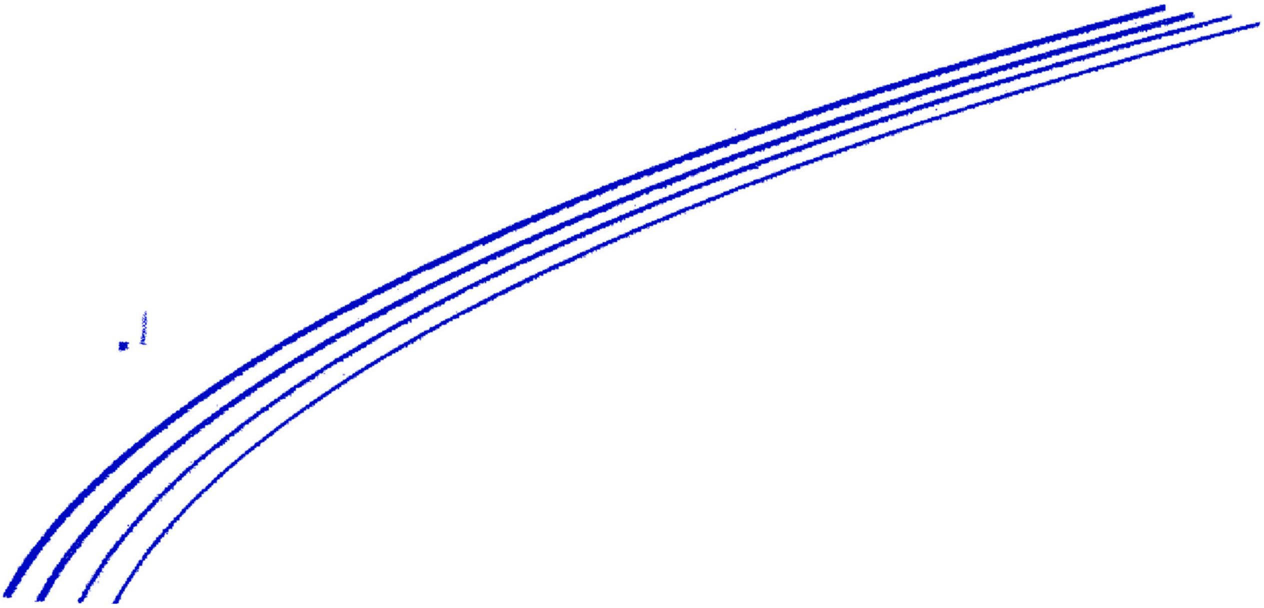


Fig. 7. Rails processed from image segmentation.

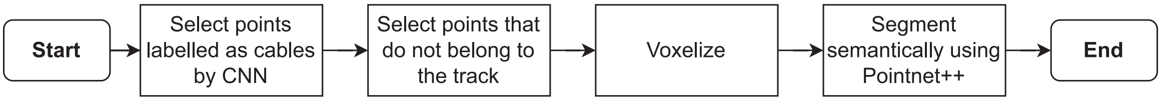

Start
Select points labelled as cables by CNN
Select points that do not belong to the track
Voxelize
Segment semantically using Pointnet++
End

Fig. 8. Cable processing diagram.

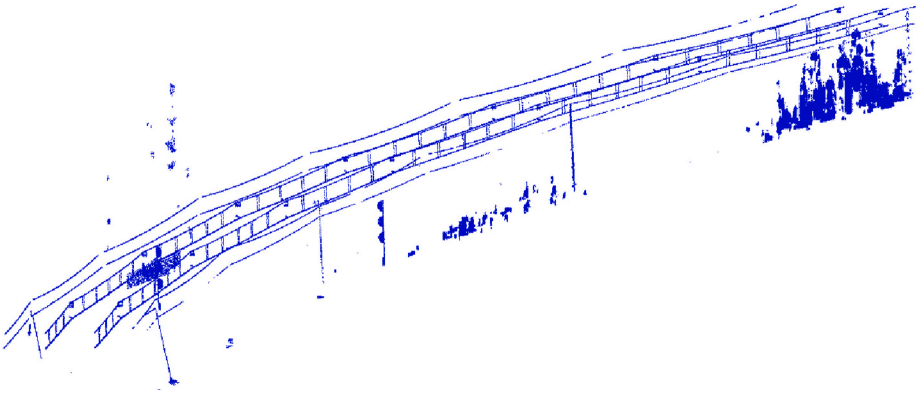


Fig. 9. Raw cables points obtained from image segmentation.

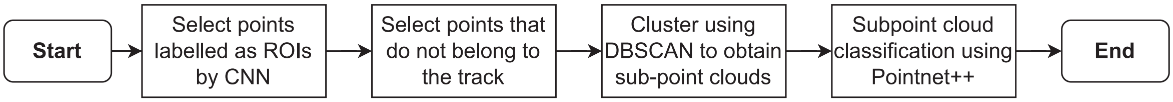

Start
Select points labelled as ROIs by CNN
Select points that do not belong to the track
Cluster using DBSCAN to obtain sub-point clouds
Subpoint cloud classification using Pointnet++
End

Fig. 10. ROI point clouds processing.

sufficient for this case. In other environments with more flexible objects, it would be necessary to consider changes in the pose of the objects; in this case, because objects were rigid, only changes in the orientation were needed.

Finally, the Pointnet++ neural network was trained for 200 epochs using the categorical cross-entropy loss and Adam optimiser with a learning rate of 0.01. In addition, to prevent overfitting, early stopping was used when the validation loss stopped decreasing.

An example of the complete labelled point cloud is shown in Fig. 12.

4. Results

As explained in Section 2, 90 km of railway data were available. These data were first split into training and test datasets, containing 80% of the training dataset. Then, 80% of the data were split into five folds to perform 5-fold cross-validation when training the neural networks, thereby ensuring the robustness of the methodology [53].

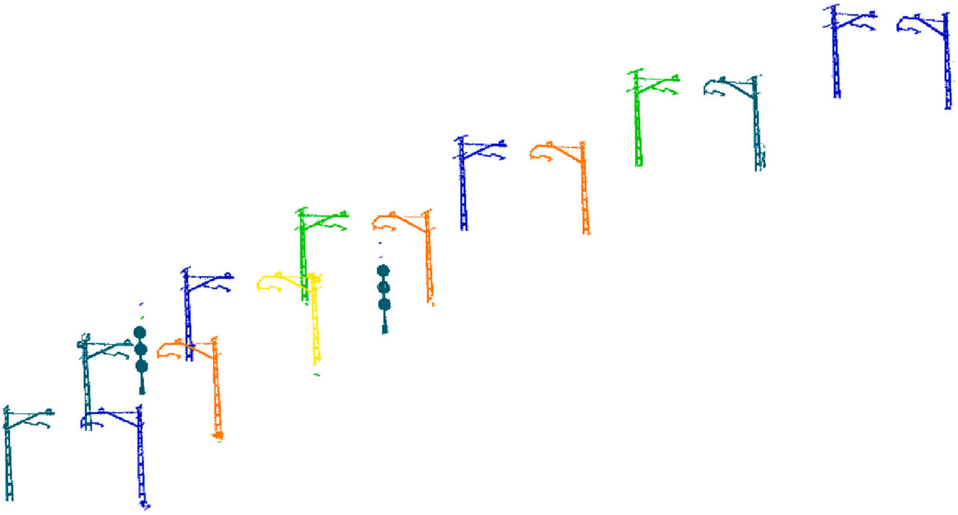


Fig. 11. ROIs obtained from image segmentation.

Table 1
Assets available for ROI classification training.

	Informative Signs	Traffic Lights	Traffic Signs	Masts
Samples	708	158	93	4017

4.1. Image segmentation

First, the performance of the image segmentation step is studied individually. This performance is not only affected by the training of the CNN, but also by the quality of the raster images generated for the task. The parameters used to create the rasters are listed in Table 2. The grid size represents the size (in meters) of the pixels in the raster images. The images had irregular sizes depending on the point cloud; therefore, they were resized using the bilinear interpolation method to a given width and height. Finally, because of interpolation, some pixels of the labelled images have values between (0,1), and they must be rounded or ceiled.

As mentioned earlier, to ensure the robustness of the method used, 5-fold cross-validation was carried out. The metric used to evaluate the quality of the image segmentation CNN was the intersection over union (IoU). The IoUs obtained over the five folds are presented in Table 3. Values of IoU higher than 0.75 were obtained in all the cases. The 5-fold-cross validation metrics were consistent, which proved that the trained

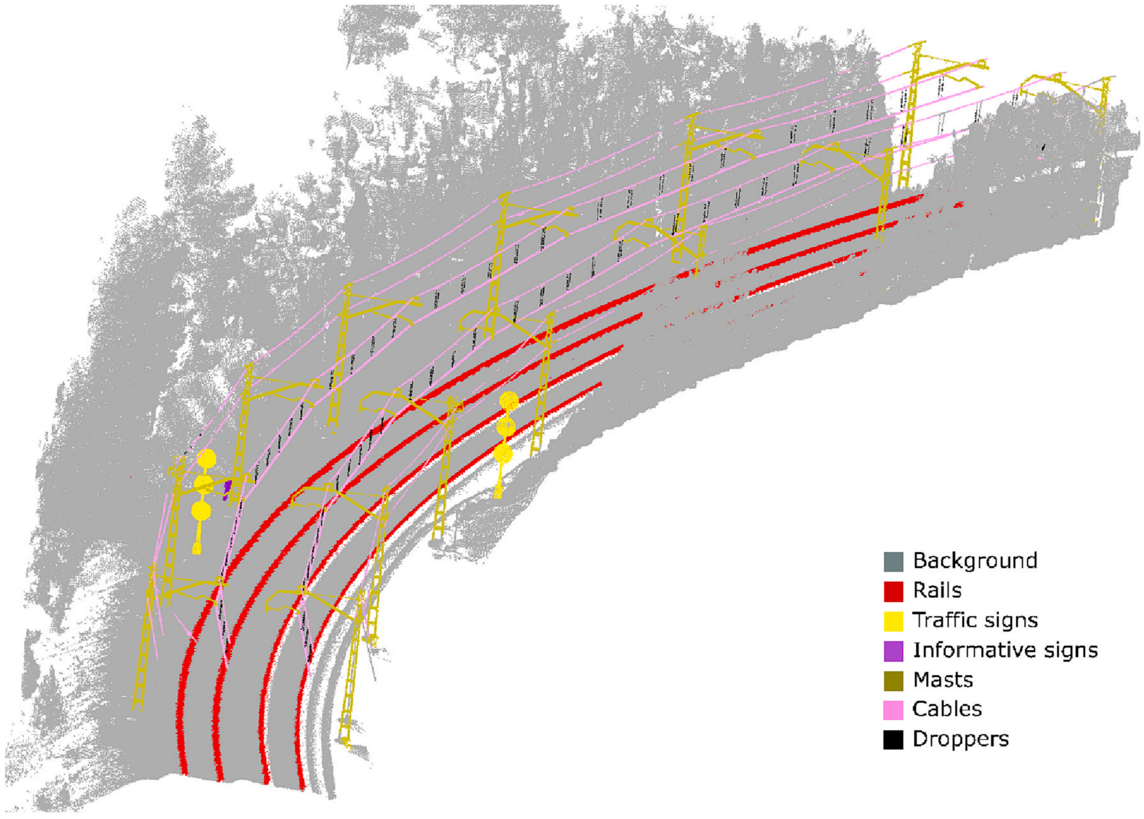


Fig. 12. Point cloud segmented using the proposed methodology.

Table 2
Raster images parameters.

Grid size (m)	Image height (pixels)	Image width (pixels)	Interpolation	Labels pixels
0.05	512	2048	Bilinear	Round

model was robust, and the higher value obtained for the test set ensured that there was no overfitting during training.

The segmentation results obtained over the test set were used in the following steps to obtain the results regarding individual assets.

4.2. Cable segmentation

The Pointnet ++ neural network used to segment the cables was trained using 5-fold cross-validation. In all the cases, the folds used were the same as those used for training the CNN. Table 4 lists the results obtained during training of the validation folds. The accuracy of the droppers was lower than that of the other assets, and the network trained on Fold 5 was used for testing. The testing accuracy is not included in the table because it is presented in Table 6 with the global results.

4.3. Classification of regions of interest

The Pointnet ++ neural network used to classify the ROIs was trained using 5-fold cross-validation. Table 5 lists the accuracy values obtained for each fold on the validation sets. In most cases, the values were greater than 90%. Thus, it can be inferred that the method adopted for the task was sufficiently robust. The network trained with Fold 3, which provided the highest mean accuracy, was used to classify the ROIs from the test set retrieved by the CNN.

4.4. Global results

Finally, once all assets were processed, the results were combined to obtain a panoptic segmentation. Using these results, a study on the global performance of the methodology was conducted. For this purpose, both the metrics and prediction runtimes were studied.

The metrics studied for the task were the precision, recall, and F1 scores. Although some of the assets to study were individual objects, others such as cables and rails were continuous and could not be studied object-wise. Consequently, the metrics presented were studied in a point-wise manner. Table 6 presents the results obtained with the proposed methodology, comparing them with those obtained by applying Pointnet++ in [39], which are discussed in the following section. This is the only comparison carried out because of the lack of implementation for railway scenarios.

Further, the runtime of the methodology predicting new point clouds was also studied. Because several steps were applied in the cascade, different sections are presented in Fig. 13. First, the **image processing runtime** included the time spent reading the point cloud, creating rasters, and performing image segmentation with the CNN. Second, the **point cloud processing runtime** included the time required for rail processing, cable segmentation, and ROI classification. At this point, all the classification values were available in the arrays. Although it is not a step from the proposed methodology, a final step in which the classification values were applied to the original point cloud, which was saved in memory, was also considered for the runtime, as is the usual procedure. This step is referred to as **cloud storage**. Finally, the **track segmentation runtime** was considered. As explained earlier, this track segmentation was not developed in this study, as several valid approaches are available in the literature.

Table 3
Image segmentation intersection over union (IoU).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Test
IoU (%)	0.755	0.755	0.795	0.787	0.794	0.777	**0.803**

Table 4
Cable segmentation accuracies.

	Droppers Acc (%)	Cables Acc (%)	Noise Acc (%)	Overall Acc (%)
Fold 1	67.45%	79.98%	97.92%	95.33%
Fold 2	**69.40%**	77.31%	97.35%	94.42%
Fold 3	63.31%	84.36%	97.14%	94.75%
Fold 4	66.23%	**84.70%**	97.47%	95.33%
Fold 5	66.51%	83.11%	**98.01%**	**95.34%**

Table 5
ROI classification accuracies.

	Marks Acc (%)	Masts Acc (%)	Noise Acc (%)	Signs Acc (%)	Traffic Lights Acc (%)
Fold 1	**97.61%**	97.13%	96.83%	94.92%	94.83%
Fold 2	95.73%	**98.18%**	96.68%	84.09%	90.57%
Fold 3	97.13%	97.55%	**97.06%**	**100.00%**	93.02%
Fold 4	94.74%	94.88%	95.61%	95.24%	97.78%
Fold 5	97.49%	96.56%	96.24%	94.74%	**98.21%**

The runtime for track segmentation is that presented in [32], and it accounts for 71.64% of the total runtime of the proposed methodology. Consequently, the real performance of the developed method is reflected by the total runtime without track segmentation, which is **0.0450 s/m**. Table 7 presents the runtimes in seconds per meter, dividing the total runtime into the subtasks carried out, and comparing them with runtimes using Pointnet++ and the heuristic method presented in [32].

Finally, Table 7 lists the point density used by the methods, and presents the runtime of Pointnet ++ working with the original point density and a reduced one. Point density is one of the parameters that define the quality of the results, as these results can be used for other tasks such as object modelling. In relation to this, the proposed methodology allows working with high-density point clouds with high-performance runtimes and maintains the original point cloud density, while Pointnet++ runtime is punished when working with the original point density of the point clouds.

5. Discussion

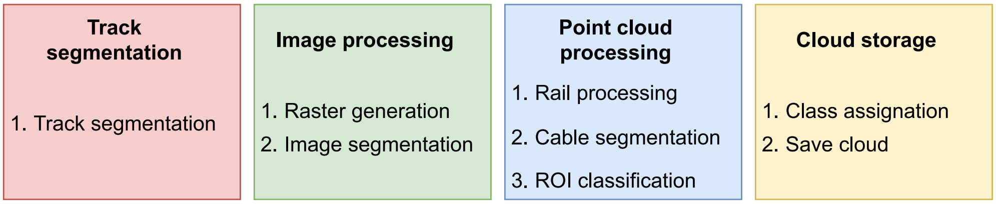
This section discusses the results obtained using the proposed method. Intermediate results, such as image segmentation, cannot be compared to any existing methodology, and they are not relevant to the task; therefore, they are obviated in this section.

In this study, a multimodal methodology is proposed for the automatic panoptic segmentation of relevant assets from railway infrastructure. The methodology starts by generating images from raw point clouds to perform fast rough segmentation. Then, the point clouds obtained from the segmentation are processed individually to obtain the final point-wise panoptic segmentation results. This methodology has the following characteristics: 1. It offers panoptic segmentation. This is an improvement over other deep learning methods that only achieve semantic segmentation: 2. It returns the results with the original point-cloud density while maintaining a high-performance runtime. Although other methodologies are punished by the output point density, this method preserves the original point density without a significant effect on the processing runtime.

The results in Table 6 can be divided into two parts. On the one hand, pole-like objects show better performance for the proposed

Table 6
General results with the proposed methodology and Pointnet++.

	Proposed methodology			Pointnet ++		
	Precision (%)	Recall (%)	F1 score (%)	Precision (%)	Recall (%)	F1 score (%)
Informative signs	**69.65%**	71.94%	**70.78%**	56.64%	**88.65%**	69.12%
Masts	86.88%	**87.22%**	87.05%	**88.79%**	87.09%	**87.93%**
Background	**99.70%**	92.54%	95.98%	99.68%	**99.04%**	**99.36%**
Traffic Signs	**60.04%**	**97.89%**	**74.43%**	52.07%	39.01%	44.60%
Traffic lights	**88.22%**	88.09%	**88.16%**	57.12%	**95.98%**	71.62%
Rails	31.38%	**99.33%**	47.69%	**81.63%**	95.75%	**88.12%**
Cables	81.78%	89.72%	85.56%	**91.13%**	**99.10%**	**94.95%**
Droppers	31.83%	65.29%	42.79%	**81.19%**	**70.93%**	**75.72%**



Track segmentation
1. Track segmentation

Image processing
1. Raster generation
2. Image segmentation

Point cloud processing
1. Rail processing
2. Cable segmentation
3. ROI classification

Cloud storage
1. Class assignation
2. Save cloud

Fig. 13. Runtime divisions.

Table 7
Prediction runtime and point density comparison.

Methodology	Section	Time (s/m)	Total time (s/m)	Point density (points/m^2)
Proposed Methodology	Track segmentation	0.1137	0.1587	**980**
	Image processing	0.0122		
	Asset processing	0.0201		
	Cloud storage	0.0127		
Heuristic	Track segmentation	0.1137	1.9600	**980**
	Segmentation	1.8463		
Pointnet ++	Point cloud processing	0.1417	**0.1417**	327
Pointnet ++	Point cloud processing	0.3265	**0.3265**	**980**

methodology than Pointnet++. While it performs similarly regarding informative signs and masts, there is a clear improvement when working with traffic signs and lights. It is also relevant to highlight that this methodology achieves panoptic segmentation, whereas the methodology presented in [39] provides only semantic segmentation. This is key because the separation of instances is required for further steps in the digitalisation of infrastructure.

On the other hand, linear objects exhibit lower performance. The rails showed a very high recall; however, the precision was relatively low. This is because the labels used as ground truth consider only the top of the rails, whereas the proposed methodology considers the rails from the top to the ground. In summary, despite the low precision value, the quality of the rail results is good, and the low value originates from the way that the ground truth is labelled. Finally, the cable and dropper performance metrics are lower.

Evaluating metrics only, the proposed methodology shows an improvement in pole-like objects, lower metrics but good quality in rails, and worse performance with cables. These results make sense because the methodology focuses on pole-like objects, whereas it is also applicable to linear assets that achieve acceptable performance.

As for the runtimes, the results in Table 7 indicate that the methodology developed in this study outperforms the other two approaches. When working with the same point density, the proposed methodology was twice as fast as Pointnet++. It is also necessary to highlight that track segmentation, which was not developed in this work, accounts for 71% of the runtime of the methodology. Finally, Table 7 also indicates how the point density used with the proposed methodology is three times higher than that used by Grandio et al. [39] to achieve similar runtimes.

In summary, by comparing our approach with the existing methodologies for point clouds in railway environments, our proposal improves the direct application of a neural network by providing i) panoptic segmentation, ii) faster runtime for similar results, and iii) higher metrics for pole-like objects.

6. Conclusions

This paper presents a multimodal deep learning methodology for the panoptic segmentation of assets found in 3D point clouds from railway infrastructure. An end-to-end pipeline was developed. The input data for the methodology consists of raw point clouds, and the point clouds are processed to obtain a point-wise classification while maintaining the original full point density for the result. Finally, full-instance segmentation is achieved for pole-like objects, and semantic segmentation is obtained for linear objects to expand the methodology for cable and rail segmentation.

This methodology shows how the instance segmentation of objects with certain characteristics can be divided into smaller steps that provide better quality results while reducing the computational complexity of the task.

The results show that the method can segment both linear and pole-like objects for point clouds collected in a railway infrastructure. This method outperforms the current state-of-the-art deep-learning semantic segmentation in railway environments when working with pole-like objects. This improvement can be divided into three categories: (1) higher performance metrics, (2) panoptic segmentation over semantic segmentation, and (3) faster runtime. In addition, a relevant improvement was achieved over heuristic methods by reducing runtimes by 72%.

This line of research achieved a significant improvement in the broader objective of digitalising railway infrastructure. The results obtained can be used to build BIM infrastructure models. Future research may include the extraction of specific geometric details of the pole-like objects to be included in the model.

Funding

This project received funding from the European Union's Horizon 2020 Research and Innovation Program under grant agreement No. 955337. This work was supported by a 2021 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation. This work was partially supported by the Spanish Ministry of Science, Innovation, and Universities through project PID2019-108816RB-I00. The support of the Spanish Ministry of Science, Innovation, and Universities through grant FPU20/01024 is acknowledged. The authors would also like to thank CESGA for allowing the use of their servers.

The BBVA Foundation takes no responsibility for the opinions, statements, or contents of this project, which are entirely the responsibility of its authors.

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.

Data availability

The authors do not have permission to share data.

References

[1] Railway passenger transport statistics - quarterly and annual data - Statistics Explained. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Railway_passenger_transport_statistics_-_quarterly_and_annual_data, 2023 accessed May 18, 2022.

[2] Railway freight transport statistics - Statistics Explained. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Railway_freight_transport_statistics, 2023 accessed May 18, 2022.

[3] Z. Allah Bukhsh, A. Saeed, I. Stipanovic, A.G. Doree, Predictive maintenance using tree-based classification techniques: a case of railway switches, Transport. Res. Part C: Emerg. Technolog. 101 (2019) 35–54, https://doi.org/10.1016/J.TRC.2019.02.001.

[4] J. Xie, J. Huang, C. Zeng, S.H. Jiang, N. Podlich, Systematic literature review on data-driven models for predictive maintenance of railway track: implications in geotechnical engineering, Geosciences 10 (2020) 425, https://doi.org/10.3390/GEOSCIENCES10110425.

[5] T. Farrington-Darby, L. Pickup, J.R. Wilson, Safety culture in railway maintenance, Saf. Sci. 43 (2005) 39–60, https://doi.org/10.1016/J.SSCI.2004.09.003.

[6] M. Kans, D. Galar, A. Thaduri, Maintenance 4.0 in railway transportation industry, lecture notes, Mech. Eng. PartF4 (2016) 317–331, https://doi.org/10.1007/978-3-319-27064-7_30/FIGURES/5.

[7] A. D'Ariano, L. Meng, G. Centulio, F. Corman, Integrated stochastic optimization approaches for tactical scheduling of trains and railway infrastructure maintenance, Comput. Ind. Eng. 127 (2019) 1315–1335, https://doi.org/10.1016/j.cie.2017.12.010.

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

[9] A.Q. Gbadamosi, L.O. Oyedele, J.M.D. Delgado, H. Kusimo, L. Akanbi, O. Olawale, N. Muhammed-yakubu, IoT for predictive assets monitoring and maintenance: an implementation strategy for the UK rail industry, Autom. Constr. 122 (2021), 103486, https://doi.org/10.1016/J.AUTCON.2020.103486.

[10] J.M. Sanne, Framing Risks in a Safety-Critical and Hazardous Job: Risk-Taking as Responsibility in Railway Maintenance, https://doi.org/10.1080/13669870701715550. 11 (2008) pp. 645–658. doi:https://doi.org/10.1080/13669870701715550.

[11] H. Feng, Z. Jiang, F. Xie, P. Yang, J. Shi, L. Chen, Automatic fastener classification and defect detection in vision-based railway inspection systems, IEEE Trans. Instrum. Meas. 63 (2014) 877–888, https://doi.org/10.1109/TIM.2013.2283741.

[12] 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.

[13] J.J. McArthur, A building information management (BIM) framework and supporting case study for existing building operations, maintenance and sustainability, Procedia Eng. 118 (2015) 1104–1111, https://doi.org/10.1016/J.PROENG.2015.08.450.

[14] J. Neves, Z. Sampaio, M. Vilela, A case study of BIM implementation in rail track rehabilitation, Infrastructures 4 (2019) 8, https://doi.org/10.3390/INFRASTRUCTURES4010008.

[15] M. Bensalah, A. Elouadi, H. Mharzi, Overview: the opportunity of BIM in railway, smart and sustainable, Built Environ. 8 (2019) 103–116, https://doi.org/10.1108/SASBE-11-2017-0060/FULL/PDF.

[16] S. Kurwi, P. Demian, T.M. Hassan, Integrating BIM and GIS in railway projects: A critical review, in: Assoc. Res. Constr. Manag. ARCOM - 33rd Annu. Conf. 2017, Proceeding, Association of Researchers in Construction Management, 2017, pp. 45–53.

[17] A. Justo, M. Soilán, A. Sánchez-Rodríguez, B. Riveiro, Scan-to-BIM for the infrastructure domain: generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data, Autom. Constr. 127 (2021), 103703, https://doi.org/10.1016/J.AUTCON.2021.103703.

[18] Y.J. Cheng, W.G. Qiu, D.Y. Duan, Automatic creation of as-is building information model from single-track railway tunnel point clouds, Autom. Constr. 106 (2019), 102911, https://doi.org/10.1016/j.autcon.2019.102911.

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

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

[21] G. Petrie, An introduction to the technology: mobile mapping systems. Geoinformatics 13, 2010, p. 32. Accessed date: 29 August 2022, https://www.proquest.com/openview/b6b8c2a2ef8cf9a354184357512977fb/1?pq-origsite=gscholar&cbl=178200.

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

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

[24] Y. Santur, M. Karaköse, E. Akin, A new rail inspection method based on deep learning using laser cameras, in: IDAP 2017 - Int. Artif. Intell. Data Process, Symp., Institute of Electrical and Electronics Engineers Inc., 2017, https://doi.org/10.1109/IDAP.2017.8090245.

[25] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, G. Cottrell, Understanding convolution for semantic segmentation, in: Proc. - 2018 IEEE Winter Conf. Appl. Comput. Vision, WACV 2018, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1451–1460, https://doi.org/10.1109/WACV.2018.00163.

[26] A.M. Hafiz, G.M. Bhat, A survey on instance segmentation: state of the art, international journal of multimedia, Inf. Retr. 9 (2020) 171–189, https://doi.org/10.1007/s13735-020-00195-x.

[27] A. Kirillov, K. He, R. Girshick, C. Rother, P. Dollar, Panoptic segmentation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, pp. 9396–9405, https://doi.org/10.1109/CVPR.2019.00963, 2019-June.

[28] T. Hackel, N. Savinov, L. Ladicky, J.D. Wegner, K. Schindler, M. Pollefeys, SEMANTIC3D.Net: a new large-scale point cloud classification benchmark, in: ISPRS Ann. Photogramm. Remote Sens, Spat. Inf. Sci., Copernicus GmbH, 2017, pp. 91–98, https://doi.org/10.5194/isprs-annals-IV-1-W1-91-2017.

[29] 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 (2020) 4338–4364, https://doi.org/10.1109/tpami.2020.3005434.

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

[31] M. Arastounia, Automated recognition of railroad infrastructure in rural areas from LIDAR data, Remote Sens. 7 (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 (2021) 2332, https://doi.org/10.3390/RS13122332.

[33] Z. Wang, G. Yu, P. Chen, B. Zhou, S. Yang, FarNet: an attention-aggregation network for long-range rail track point cloud segmentation, IEEE Trans. Intell. Transp. Syst. (2021), https://doi.org/10.1109/TITS.2021.3119900.

[34] H. Cui, J. Li, Q. Hu, Q. Mao, Real-time inspection system for ballast railway fasteners based on point cloud deep learning, IEEE Access. 8 (2020) 61604–61614, https://doi.org/10.1109/ACCESS.2019.2961686.

[35] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, in: Adv. Neural Inf. Process. Syst., Neural Information Processing Systems Foundation, 2017, pp. 5100–5109.

[36] 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, ISPRS annals of photogrammetry, in: Remote Sensing and Spatial Information Sciences. V-2–2020, 2020, pp. 281–288, https://doi.org/10.5194/isprs-annals-V-2-2020-281-2020.

[37] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 77–85, https://doi.org/10.1109/CVPR.2017.16.

[38] H. Thomas, C.R. Qi, J.E. Deschaud, B. Marcotegui, F. Goulette, L. Guibas, KPConv: flexible and deformable convolution for point clouds, in: Proc. IEEE Int. Conf. Comput, Vis., Institute of Electrical and Electronics Engineers Inc., 2019, pp. 6410–6419, https://doi.org/10.1109/ICCV.2019.00651.

[39] J. Grandio, B. Riveiro, M. Soilán, P. Arias, Point cloud semantic segmentation of complex railway environments using deep learning, Autom. Constr. 141 (2022), 104425, https://doi.org/10.1016/J.AUTCON.2022.104425.

[40] F. Eickeler, A. Borrmann, Enhancing railway detection by priming neural networks with project Exaptations, Remote Sens. 14 (2022) 5482, https://doi.org/10.3390/rs14215482.

[41] M. Soilán, A. Nóvoa, A. Sánchez-Rodríguez, A. Justo, B. Riveiro, Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data, Autom. Constr. 126 (2021), 103684, https://doi.org/10.1016/J.AUTCON.2021.103684.

[42] Home | Teledyne Geospatial. https://www.teledyneoptech.com/en/home/, 2023 accessed May 16, 2022.

[43] P. Chu, S. Cho, S. Sim, K. Kwak, K. Cho, A fast ground segmentation method for 3D point cloud, J. Informat. Process. Syst. 13 (2017) 491–499, https://doi.org/10.3745/JIPS.02.0061.

[44] W. Huang, H. Liang, L. Lin, Z. Wang, S. Wang, B. Yu, R. Niu, A fast point cloud ground segmentation approach based on coarse-to-fine Markov random field, IEEE Trans. Intell. Transp. Syst. (2021), https://doi.org/10.1109/TITS.2021.3073151.

[45] P. Narksri, E. Takeuchi, Y. Ninomiya, Y. Morales, N. Akai, N. Kawaguchi, A slope-robust cascaded ground segmentation in 3D point cloud for autonomous vehicles, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 2018-November, 2018, pp. 497–504, https://doi.org/10.1109/ITSC.2018.8569534.

[46] M. Velas, M. Spanel, M. Hradis, A. Herout, CNN for very fast ground segmentation in velodyne LiDAR data, in: 18th IEEE International Conference on Autonomous Robot Systems and Competitions 2018, ICARSC, 2018, pp. 97–103, https://doi.org/10.1109/ICARSC.2018.8374167.

[47] N. El-Ashmawy, A. Shaker, Raster vs. point cloud lidar data classification, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 40, 2014, pp. 79–83, https://doi.org/10.5194/ISPRSARCHIVES-XL-7-79-2014.

[48] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: a brief review, Computat. Intellig. Neurosci. 2018 (2018), https://doi.org/10.1155/2018/7068349.

[49] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9351, 2015, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.

[50] Y. Mo, Y. Wu, X. Yang, F. Liu, Y. Liao, Review the state-of-the-art technologies of semantic segmentation based on deep learning, Neurocomputing. 493 (2022) 626–646, https://doi.org/10.1016/J.NEUCOM.2022.01.005.

[51] H. Zhao, L. Jiang, J. Jia, P. Torr, V. Koltun, Point transformer, in: Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 16239–16248, https://doi.org/10.1109/ICCV48922.2021.01595.

[52] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, 1996.

[53] P. Refaeilzadeh, L. Tang, H. Liu, Cross-Validation, Encyclopedia of Database Systems, 2016, pp. 1–7, https://doi.org/10.1007/978-1-4899-7993-3_565–2.