

SEMANTIC SEGMENTATION OF POINT CLOUDS WITH POINTNET AND KPConv ARCHITECTURES APPLIED TO RAILWAY TUNNELS

M. Soilán^{1,*}, A. Nóvoa¹, A. Sánchez-Rodríguez¹, B. Riveiro¹, P. Arias²

¹ Dept. of Materials Engineering, Applied Mechanics and Construction, School of Industrial Engineering, University of Vigo, 36310, Vigo, Spain – (msoilan, andrea.novoa.martinez, anasanchez, belenriveiro)@uvigo.es

² Dept. of Natural Resources and Environmental Engineering, School of Mining Engineering, University of Vigo, 36310, Vigo, Spain - parias@uvigo.es

Commission II, WG II/3

KEY WORDS: Deep Learning, Point Cloud Processing, Railway tunnel, Semantic Segmentation, Infrastructure Monitoring.

ABSTRACT:

Transport infrastructure monitoring has lately attracted increasing attention due to the rise in extreme natural hazards posed by climate change. Mobile Mapping Systems gather information regarding the state of the assets, which allows for more efficient decision-making. These systems provide information in the form of three-dimensional point clouds. Point cloud analysis through deep learning has emerged as a focal research area due to its wide application in areas such as autonomous driving. This paper aims to apply the pioneering PointNet, and the current state-of-the-art KPConv architectures to perform scene segmentation of railway tunnels, in order to validate their employability over heuristic classification methods. The approach is to perform a multi-class classification that classifies the most relevant components of tunnels: ground, lining, wiring and rails. Both architectures are trained from scratch with heuristically classified point clouds of two different railway tunnels. Results show that, while both architectures are suitable for the proposed classification task, KPConv outperforms PointNet with F1-scores over 97% for ground, lining and wiring classes, and over 90% for rails. In addition, KPConv is tested using transfer learning, which gives F1-scores slightly lower than for the model training from scratch but shows better generalization capabilities.

1. INTRODUCTION

The modern society is increasingly dependent on transportation networks in its daily activities. In a context of increasing risk of extreme natural hazards related to climate change, and a lack of maintenance on road and rail infrastructure, with maintenance budgets that do not evolve with the length and age of the infrastructure (European Commission, 2019), higher risks of accidents, congestions and serviceability reduction are expected. The infrastructure requires relevant adaptations in order to improve its resilience, that is, to improve its ability to resist, adapt to, and recover from the effects of a hazard, considering all modes of disaster cycle: preparation, response and recovery, and mitigation (Erdelj et al., 2017). Focusing on the preparation, it is important to highlight infrastructure monitoring as an essential component within an infrastructure management system, as it offers data that allows to understand and quantify the state of the infrastructure, as well as to make decisions according to it. In a context of limited budget for infrastructures, there is a need of research on new technologies that improve infrastructure monitoring, which allows a more efficient and automated data collection and decision-making.

Mobile Mapping Systems (MMS) have evolved rapidly during the last few years, as they have proven to be a valid technology for infrastructure monitoring applications. They are mobile platforms equipped with different monitoring sensors, such as laser scanners, which are able to collect 3-dimensional (3D) representations of the environment in the form of point clouds on an accurate and efficient manner. MMS typically include other remote sensing components such as positioning sensors or RGB cameras, which provide additional data that is coupled to the point cloud. Such a point cloud is a sparse non-grid structured

data and hence its processing is more challenging than that of 2-dimensional grids. Research regarding the treatment of the data collected by MMS systems has been constantly increasing year by year, so as the number of applications related to infrastructure monitoring, as seen in different recently published reviews (Che et al., 2019; Guan et al., 2016; Ma et al., 2018; Soilán et al., 2019c). Some of the most common applications address the automatic processing of 3D point cloud data to obtain information about the road surface (e.g. pavement, road markings, driving lanes, road cracks) and relevant assets of the infrastructure (e.g. traffic signs, poles, vegetation, power lines).

From the aforementioned research reviews, it can be seen that while the research on road infrastructure is relatively prolific, the number of publications that study railway infrastructure is considerably lower. This is motivated by the complexity of installing the equipment of the MMS on the railway, which requires special auxiliary vehicles such as a draisine; the correspondent permissions by the infrastructure operators; and a very specific and typically constrained window of time when the inspection can be carried out. Nonetheless, there have been relevant research projects on railway infrastructure. Arastounia (2015) developed a heuristic process that recognizes key elements of the railroad infrastructure (rails, cables, masts and cantilevers) with very high precision at object level using a MMS mounted on a train, by using the geometric properties and topological relationship of the 3D point cloud data. This work has been progressively enhanced, using a template matching for clustering the railroad into rail track, contact cable and catenary cable classes (Arastounia and Oude Elberink, 2016).

* Corresponding author: msoilan@uvigo.es

More recently, heuristic methods have been upgraded by making them applicable to railroads with any slope, and more independent to the rails' dimensions and point cloud density (Arastounia, 2017). Lou et al. (2018) proposed a real-time algorithm that detects rails based on their geometric and radiometric properties, using a Velodyne scanner. Other works apply supervised classification algorithms, instead of relying only on heuristics, in order to reduce the dependence on manual thresholds and conditions. Sánchez-Rodríguez et al. (2018) combine a first heuristic step with a subsequent application of a linear Support Vector Machine (SVM) to classify rails. This type of classifier is also employed in Hackel et al. (2015) to classify rail frogs with features such as rail alignment, normal orientation or the distance of nearest neighbours in three different rails.

Alternatively, Luo et al., (2014) proposed a context-based classification method to automatically recognise railway objects from point cloud data acquired with an Optech Lynx M1 scanner. This work enhanced the predictions from a local classifier of Gaussian Mixture Model-Expectation Maximization (GMM-EM) by incorporating contextual information with a Conditional Random Field (CRF) model. Afterwards, this CRF model was used to develop supervised learning classification methods using a SVM local classifier for the classification of the different components of railway electrification systems (Jung et al., 2016; Chen et al., 2019).

Nowadays, the usage of new deep learning architectures for supervised classification or semantic segmentation of 3D point cloud data is a very relevant research topic. This research aims to develop end-to-end processes where the heuristics or feature handcrafting processes are minimized, while the classification models learn by themselves the relevant features that allow to classify the input data. While the state of the art on 2D image classification with Deep Learning models is large, it is still an emergent field on 3D data. A recent review of this research can be found in Liu et al. (2019). It highlights the models of PointNet (Qi et al., 2016) and PointNet++ (Qi et al., 2017) as pioneers working with raw point clouds for classification and semantic segmentation, addressing unstructured point cloud data. There also exist networks inspired by Convolutional Neural Networks (CNN) such as PointCNN (Li et al., 2018) or KPConv (Thomas et al., 2019), or by Recurrent Neural Networks (RNN) such as 3P-RNN (Ye et al., 2018) among others.

As the application of these deep learning architectures is very limited in railway environments, this paper proposes the application of two different architectures for the classification of railway tunnels: PointNet and KPConv. While the first is a pioneering architecture that has been successfully employed in different applications, the latter is recent and shows great potential. Summarizing, the aim and contributions of this work are the following:

- 1) To analyse and compare the performance of the two proposed deep learning-based models through their application in 3D point clouds of railway tunnels, classifying the most relevant parts of the infrastructure.
- 2) To draw conclusions on the employability of these methods over heuristic-based methods, to foresee future work related with infrastructure monitoring and generation of information models.

This paper is structured as follows: Section 2 presents the data employed in this work. Section 3 describes the methodology and classification models, and details how the data is managed to get

the results, which are presented and discussed in Section 4. Finally, Section 5 outlines the conclusions and future work.

2. CASE STUDY DATA

This study employs 3D point cloud data acquired with a LYNX Mobile Mapper system (Optech, 2019). A complete description and an accuracy study of this system can be found in (Puente et al., 2013). The acquisition vehicle's average speed was 5 km/h. Data from two railway tunnels (named Tunnel A and Tunnel B in this work) are used to develop and validate the proposed methodology. Both tunnels have circular shape and two power lines on top of the rails. In order to make the point clouds manageable, they were divided in sections of about 20 meters of length each in the direction of the vehicle trajectory. Point clouds at the entrances that include data from outside the tunnel were discarded. Then, following a proportion of approximately 80/20%, data from each tunnel were split for training and testing the models resulting from this work (Table 1).

Dataset	Length (m)	Number of sections	Training sections	Test sections
Tunnel A	850	41	32	9
Tunnel B	1 100	54	44	10

Table 1. Parameters of the case study data.

3. METHODOLOGY

This work validates the capability of two different deep learning architectures, KPConv and PointNet, for semantic segmentation of 3D point clouds of railway tunnels. In this section, the data labelling and preprocessing are firstly presented. Then, the main characteristics of the proposed architectures are summarized, along with the most relevant parameters that were used for training the classification models.

3.1 Data labelling and preprocessing

Data labelling is an indispensable step to carry out when training supervised classification models. Obtaining labelled data is typically a manual and tedious process. While there already exist labelled datasets for a wide range of image classification applications, the number of 3D point cloud labelled datasets is limited. In this work, the data introduced in Section 2 are not initially labelled, and to the best knowledge of the authors there are not public datasets of 3D point clouds of railway tunnels with labels to assist supervised classification. Therefore, the first step of the proposed methodology consists of assigning a single label to each point of the case study data.

In order to avoid an extensive and manual labelling process, this method takes advantage of previous work. Sánchez-Rodríguez et al. (2018) present a heuristic-based process to automatically detect different parts of railway tunnels such as lining, ground, rails, power line wires and cantilever arms. Following this work, Sánchez-Rodríguez et al. (2019) performed an automated inspection of power line wires, classifying contact wires and suspension wires. By combining the results of these works, the dataset is initially labelled considering seven classes: ground, lining, cantilever arm, contact wire, suspension wire, rail, and unclassified points. Although the quality of this automatic labelling is high, a further manual refinement is done. First, wires and cantilever arms are merged in the same class, 'wiring'. Then, unclassified points are manually checked and assigned to the most appropriate class, ending up with a four-class classification (ground, lining, wiring and rails; Figure 1a). Nevertheless,

mislabelling may occur specially in the boundaries of different classes due to subjectivity on the manual refinement or unchecked areas (Figure 1b).

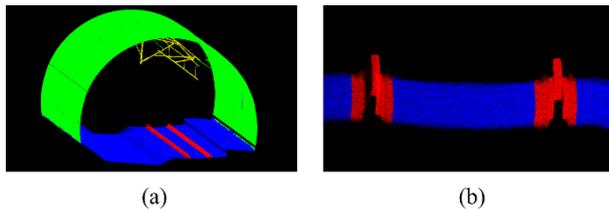


Figure 1. Point cloud labelling. (a) Each point cloud is labelled defining four classes, namely ground (blue), lining (green), rails (red) and wiring (yellow). (b) Mislabelling may occur in their boundaries. In this case, between rails and ground classes.

3.2 Deep learning architectures

3.2.1 KPConv

Thomas et al. (2019) introduced the Kernel Point Convolution (KPConv) operator. KPConv is a point convolution design that operates directly on the points and consists of a set of local 3D filters. This is the current state-of-the-art approach for semantic segmentation. In contrast to multilayer perceptrons (MLP) methods such as PointNet (Qi et al., 2016), this method is inspired in 2D image convolutions and explicitly defines a set of learnable convolution kernels by using the spatial localisation of the point cloud. These kernel points determine the areas for applying the kernel weights, which are computed as the Euclidean distance to the kernel points. KPConv allows any number of kernel points, which makes this design highly flexible. The positions of the kernel points are formulated as an optimization problem of best coverage in the sphere space.

Specifically, this work considers the Kernel Point Fully Convolutional Neural Network (KP-FCNN) architecture proposed by Thomas et al. (2019) for 3D scene segmentation. The KP-FCNN segmentation starts with a subsampling process. This procedure divides the point cloud into smaller clouds contained in spheres, which are chosen randomly during training and regularly during testing. In addition, KPConv operator uses the radius neighbourhood technique, which allows to keep a consistent receptive field. The grid subsampling and the radius neighbourhood ensure that the algorithm is robust in non-uniform sampling areas and reduce the computational cost required, when compared to other point convolution networks such as Pointwise CNN (Thomas et al. 2019).

The first approach to apply KPConv to the dataset described in Section 2 is to apply transfer learning to take advantage of the learned features in a more complex dataset. This work considers the 3D dataset Semantic3D (Hackel et al., 2017), which Thomas et al. (2019) used for the application of KPConv for scene segmentation. This dataset consists of a large set of point clouds obtained from outdoor fixed scans, with over four billion manually labelled points divided into eight classes. The process of applying transfer learning starts by training the KPConv in the Semantic3D dataset and the resulting model is used for feature extraction. This pretrained model is adapted for its application to the railway tunnel dataset by freezing every layer in the network except the input and output layers. Then, the modified pretrained model is trained with the 80% of the data and tested in the remaining 20%, as mentioned in Section 2. The reason to restrict training to the external layers and not updating the weights across

the whole network is to avoid overfitting, as the tunnel data is smaller and more repetitive than that of Semantic3D.

The second approach to apply KPConv consists of retraining a model from scratch. The parameters used for this training are the default parameters defined by Thomas et al. (2019) for the Semantic3D scene segmentation.

Both KPConv models are trained defining the same parameters. Initially, all the main parameters are maintained as set out by Thomas et al. (2019). This method uses momentum gradient descent to minimise a point-wise cross-entropy loss using a batch size of 10, 0.98 momentum and a learning rate of 10^{-2} . The number of kernels is first set to $K = 15$, the initial subsampling parameter to $dl_0 = 0.06$ m, and the radius of influence to $R = 3.0$ m. The subsampling parameter is also tested at $dl_0 = 3$ cm, however, this increases the computation cost by a factor of 5 and thus its application is not practical.

In order to ease the convergence, the point clouds are first centred in the XY-plane such that:

$$(\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c) = (\mathbf{x}, \mathbf{y}, \mathbf{z}) - (x_{avg}, y_{avg}, 0) \quad (1)$$

where x_{avg} and y_{avg} are the mean x and y coordinates of the point cloud, respectively. Moreover, the point clouds are aligned to the rails direction by multiplying the point cloud with the principal components of the points categorised as ‘rails’, which are determined with the MATLAB built-in function *pca*.

3.2.2 PointNet

Along with KPConv, this work employs PointNet. This pioneering architecture is considered as a milestone in point cloud deep learning due to its capability of directly processing 3D point clouds. PointNet uses multi-layer perceptrons (MLPs) to learn per-point features and it can be applied for semantic segmentation applications. A complete description of the architecture can be found in (Qi et al., 2016). Originally, this network employed indoor point clouds, but it has proven to be valid for semantic segmentation of aerial (Soilán et al., 2019a, 2019b) and terrestrial (Balado et al., 2019) point clouds. There, different parameters are mentioned to have an influence on the preparation of the data. First, the point cloud is divided in squared blocks defined by a block size and a block stride. Furthermore, the number of points to be sampled per block is defined (if the number of points within a block is smaller, points are duplicated randomly to match the required number of points).

In terms of point cloud processing, it is necessary to normalize the point cloud coordinates in a first place to ease the convergence during the network training process. For that purpose, each point cloud is translated. Defining $\mathcal{P} = (\mathbf{x}, \mathbf{y}, \mathbf{z})$ as the raw data, the translated point cloud is defined in Equation (2):

$$(\mathbf{x}_t, \mathbf{y}_t, \mathbf{z}_t) = (\mathbf{x}, \mathbf{y}, \mathbf{z}) - (x_{min}, y_{min}, z_{min}) \quad (2)$$

where $(x_{min}, y_{min}, z_{min})$ are the minimum values of each coordinate. This way, every point cloud has a point at $(0, 0, 0)$ and there are not negative coordinates.

Then, the point cloud is divided in blocks using the aforementioned parameters. For this work, they were defined as $block_{size} = 5$ m, $block_{stride} = 2$ m, $num_{points} = 8192$. The

coordinates within each block are normalized (x_n, y_n, z_n) such that they are in the range $[0, 1]$. Finally, the network is fed with a $N \times 8192 \times 6$ array, where N is the number of blocks or data batches. For each point, a 6-dimensional feature is defined as:

$$feature_pointnet = (x_t, y_t, z_t, x_n, y_n, z_n) \quad (3)$$

That is, only the geometric coordinates are used to define the feature. Then, a PointNet model is trained for 30 epochs, a batch size of 8, with Adam (Kingma and Ba, 2015) as optimizer and with a learning rate of 0.001 with decay each 300 000 training samples.

3.3 Data evaluation and comparison

In order to evaluate and compare the multiclass classification results obtained from both architectures, in this work the following classification metrics are used:

- Precision, (P), also known as positive predictive value, evaluates the proportion of the predicted points of a class that truly belong to it. For each class the precision is computed as:

$$P = \frac{TP}{TP + FP} \quad (4)$$

- Recall, (R), or true positive rate, evaluates fraction of the amount of predicted points of a class that are successfully classified. The recall of each class is determined as:

$$R = \frac{TP}{TP + FN} \quad (5)$$

- F1-score, (F_1), combines precision and recall into a single metric by computing their harmonic mean, which gives a larger weight to small values:

$$F_1 = 2 \frac{P \cdot R}{P + R} \quad (6)$$

where TP is the number of True Positives, FP is the number of False Positives, and FN is the number of False Negatives for a given class. In general, the higher P , R , and F_1 values, the better performance of the deep learning models.

4. RESULTS AND DISCUSSION

In order to train and validate the proposed architectures, different resources from the Supercomputing Centre of Galicia (CESGA) were used. Specifically, the GPUs employed were a NVIDIA Tesla V100 PCIe, and a NVIDIA Tesla K80.

4.1 KPConv architecture

The KP-FCNN architecture introduced in Section 3.2.1 is first trained with transfer learning using the pretrained model with Semantic3D. The input and output layers are retrained with the dataset described in Section 2 using the default parameters.

Figure 2 shows how the cross-entropy loss continuously decreases with the number of epochs to eventually stabilise at around 300 epochs. Similarly, the overall training accuracy increases rapidly and converges, reaching values over 91%. Therefore, the KPConv model pretrained in the Semantic3D dataset is successfully fine-tuned and takes advantage of the features learned with these complex point clouds to identify the different classes in the railway tunnel dataset.

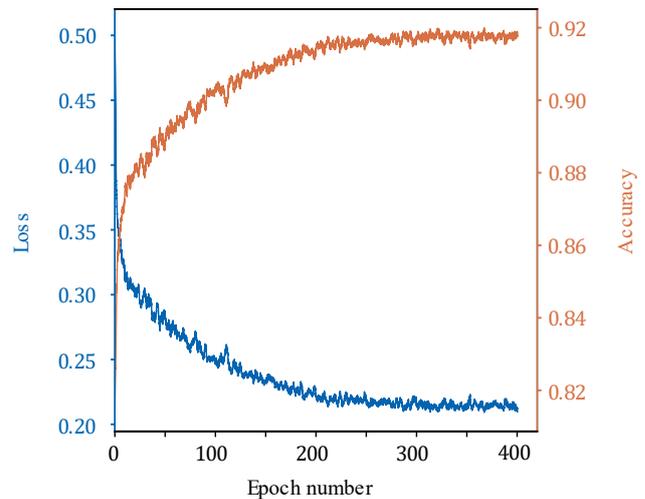


Figure 2. Evolution of the point-wise cross-entropy loss and the overall training accuracy with the epoch number for the training of KPConv using transfer learning.

This model is tested using the railway tunnel sections, as defined in Section 3.1. Table 2 provides the confusion matrix of the semantic segmentation results of KPConv using transfer learning with the default parameters. This table shows that, overall, most of the points are correctly classified. The rails category shows the greatest discrepancies with 31% of the ground truth (GT) rail points being classified as ground.

GT \ Prediction	Ground	Lining	Wiring	Rails
Ground	27 055 384	542 486	710	130 434
Lining	793 413	73 516 588	20 200	815
Wiring	12 378	123 541	628 003	0
Rails	768 771	4 372	0	1 700 967

Table 2. Confusion matrix for the results obtained from the KPConv model with transfer learning

In order to provide more insight into the performance of the test, Table 3 includes the main classification metrics, including the precision and recall of the predictions as well as the F1-score.

Metric \ Class	Ground	Lining	Wiring	Rails
Precision	94.50%	99.10%	96.78%	92.84%
Recall	97.57%	98.90%	82.21%	68.75%
F1-score	96.01%	99.00%	88.90%	79.00%

Table 3. Classification metrics for the KPConv transfer learning results including precision, recall and F1-score.

Figure 3 includes the prediction, ground truth and misclassified points for one section in Tunnel A and another in Tunnel B. These figures show that the point clouds are properly segmented overall, with discrepancies mostly around the boundaries of the different elements.

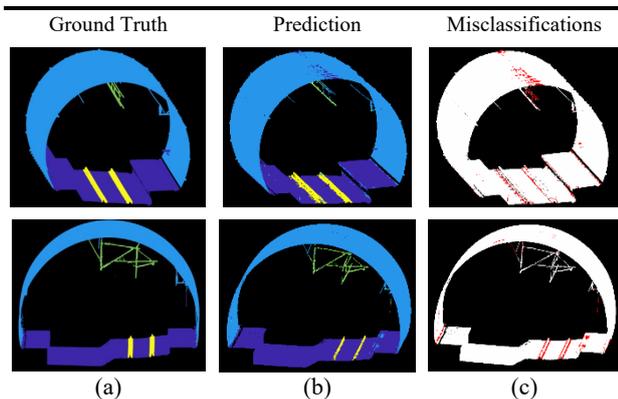


Figure 3. Semantic segmentation results of KPConv with transfer learning for tunnel A (top row) and B (bottom row). (a) Ground truth. (b) Prediction. (c) Misclassified points (in red).

The most notable discrepancies in Figure 3 are located around the rail sections, which agrees with the confusion matrix results. Although this suggests that the rails segmentation is not accurate, this may be a misleading result. The labelling process described in Section 3.1 may include some errors specially in the boundaries, as illustrated in Figure 4. This figure shows that in the original ground truth there are some ground regions to the left and right of the rails that are, arguably, wrongly categorised as rails.

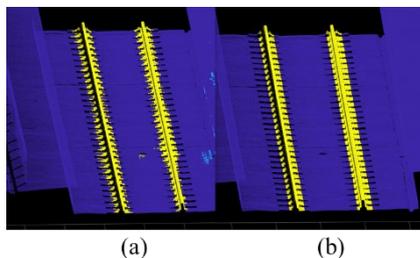


Figure 4. Prediction (a) and ground truth (b) of the rails' region for the KPConv segmentation with transfer learning.

Indeed, when this area is inspected, it is appreciated that the true rail points are properly segmented. This is illustrated in Figure 5, where most of the misclassified labels actually belong to the ground.



Figure 5. Section of KPConv semantic segmentation with transfer learning showing the misclassified points (in red) in the rails area (XZ view).

The second test performed trains the network from scratch, as explained in Section 3.2.1. This gives a deeper insight in the ability of the KPConv network to perform scene segmentation in our problem. Figure 6 shows that when retraining the model with the data as described in Section 3.1, the accuracy and loss converge significantly faster and exhibits lower fluctuations. Moreover, the accuracy reaches values over 99% with this method.

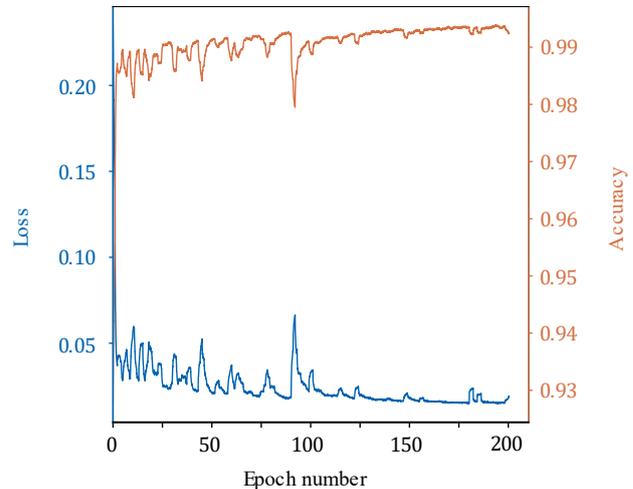


Figure 6. Evolution of the point-wise cross-entropy loss and the overall training accuracy with the epoch number for KPConv retraining.

The confusion matrix for this case is shown in Table 4 and the classification performance metrics are summarised in Table 5. These tables imply that the performance of the model is significantly improved when compared to the transfer learning case. This improvement is particularly significant in the classification of wiring and rails, achieving a F1-score increase of 8.7% and 12.1%, respectively. However, as discussed previously, the classification of the data used as the “ground truth” is not perfect, especially in the wiring and rails segmentation. Consequently, the obtained results are not perfect either, even though the classification metrics and confusion matrix show excellent results. This issue is explored and further discussed in Section 4.3.

GT \ Prediction	Ground	Lining	Wiring	Rails
Ground	27 221 034	108 345	112	399 523
Lining	57 076	74 251 386	22 554	0
Wiring	0	13 590	750 332	0
Rails	71 564	0	0	2 402 546

Table 4. Confusion matrix for the retrained KPConv model results

Metric \ Class	Ground	Lining	Wiring	Rails
Precision	99.53%	99.84%	97.07%	85.74%
Recall	98.17%	99.89%	98.22%	97.11%
F1-score	98.84%	99.86%	97.64%	91.07%

Table 5. Classification metrics for the retrained KPConv model results including precision, recall and F1-score.

Figure 7 shows, analogously to Figure 3, the semantic segmentation results of the KPConv retrained model. It is appreciated that the misclassifications around the rails have decreased substantially. Nevertheless, the model segments ground regions into the rails category because of the errors in the heuristically-labelled data, as seen in Figure 4. This means that the model is overfitting the training dataset.

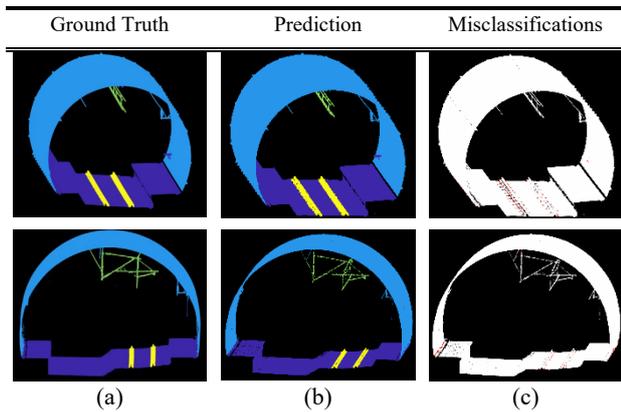


Figure 7. Semantic segmentation results of the retrained KPConv for tunnel A (top row) and B (bottom row). (a) Ground truth. (b) Prediction. (c) Misclassified points (in red).

4.2 PointNet architecture

The PointNet model described in Section 3.2.2 is trained with the same training data than for KPConv, as described in Section 2. As it can be seen in Figure 8, the model trains rapidly using a relatively small number of epochs, achieving more than 98% of training accuracy. Comparing this result with that of Figures 2 and 6, the number of training epochs in this network is significantly lower. The model is not trained further because there were no noticeable improvements after 30 epochs.

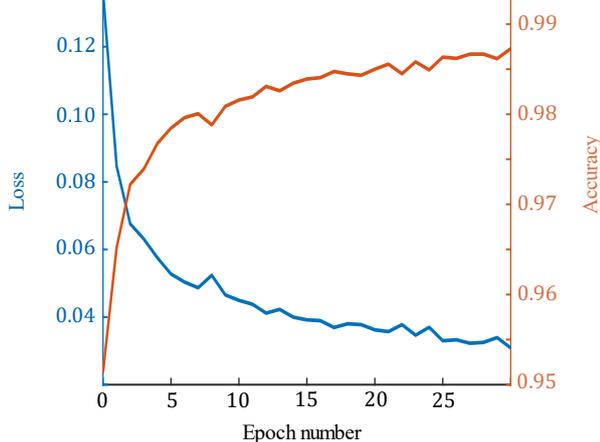


Figure 8. Evolution of the point-wise cross-entropy loss and the overall training accuracy with the epoch number for the training of PointNet.

The model is tested using the 19 railway tunnel sections defined in Section 2, allowing direct comparison with the results obtained with KPConv. Table 6 shows the results of the semantic segmentation for all the test data in the form of a confusion matrix.

GT \ Prediction	Ground	Lining	Wiring	Rails
Ground	26 676 609	674 182	1	378 222
Lining	400 183	73 878 122	82 711	0
Wiring	0	99 147	664 775	0
Rails	407 702	9062	0	2 057 346

Table 6. Confusion matrix for the PointNet model results.

It can be seen that, even with a large class unbalance where most of the points belong ground and lining classes, the performance

metrics (Table 7) are positive and the system reaches F1-scores of 83.81% and 89.75% for the minority classes of rails and wiring, respectively.

Metric \ Class	Ground	Lining	Wiring	Rails
Precision	97.06%	98.95%	92.65%	84.47%
Recall	96.20%	99.39%	87.02%	83.15%
F1-score	96.63%	99.17%	89.75%	83.81%

Table 7. Classification metrics for the PointNet model results including precision, recall and F1-score.

Figure 9 shows the predictions for one of the test sections in both tunnels of the case study. Overall, all the defined classes are properly segmented after the process. In Figure 9 the misclassified points are highlighted. As it can be seen, boundaries between elements largely contribute to the number of misclassifications, especially in the case of rails. Note that this issue was already mentioned in Section 3.1, regarding the labelling process.

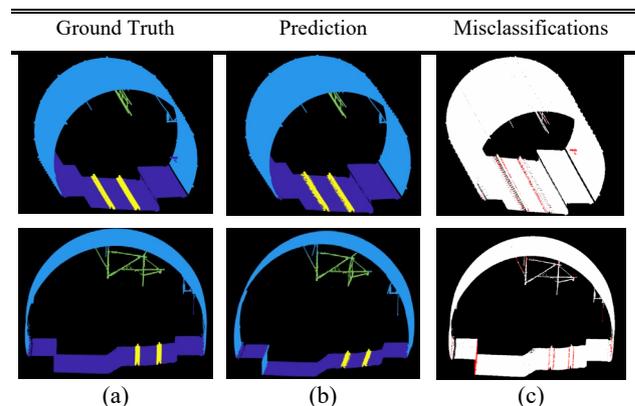


Figure 9. Semantic segmentation results of the retrained PointNet for tunnel A (top row) and B (bottom row). (a) Ground truth. (b) Prediction. (c) Misclassified points (in red).

4.3 Comparison and discussion

In previous sections, it was seen that misclassifications in object boundaries were one of the main issues affecting the performance of the models. Data labelling has an influence on this issue, as some points, especially on rail boundaries, are arguably labelled as rail when they may belong to the ground. In order to offer a better insight, the ground truth of a single section was refined to perform a comparison between the three proposed models, providing more precise labels of the rails and their boundaries, as shown in Figure 10. This manual refinement improves the original data used for training (see Figure 2b) and sets a better ground for comparison.

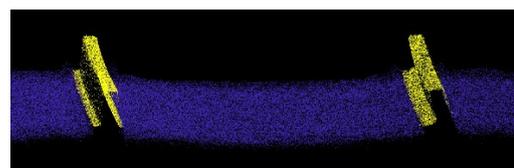


Figure 10. Manual segmentation of the rails.

The performance metrics obtained from the three proposed models, KPConv with transfer learning, KPConv with retraining, and PointNet are shown in Table 8. Figure 11 includes the misclassified points for each model. From these results, as well

	Precision			Recall			F1-score		
	KPConv TL	KPConv R	PointNet	KPConv TL	KPConv R	PointNet	KPConv TL	KPConv R	PointNet
Ground	99.22%	98.84%	99.15%	96.69%	94.75%	91.22%	97.94%	96.75%	95.02%
Lining	98.55%	99.87%	98.02%	99.81%	99.51%	99.66%	99.18%	99.69%	98.83%
Wiring	98.83%	99.99%	100%	87.42%	92.15%	87.66%	92.77%	95.91%	93.42%
Rails	87.82%	39.38%	42.18%	89.35%	100.00%	99.90%	88.58%	56.51%	59.33%

Table 8. Performance metrics for the results from the application of KPConv transfer learning (TL), KPConv retrain (R), and PointNet to a refined point cloud with manual labelling of points in the boundaries of rails.

as from the results shown in Section 4.1 and Section 4.2, it can be seen that both models throw good results for the task of semantic classification of railway tunnels. As expected from the results in Thomas et al. (2019), KPConv outperforms PointNet in terms of F1-score. However, it is relevant to note that training time of PointNet was, per epoch, up to three times faster than for KPConv. The reason for this being the initial subsampling process that this network performs and its higher overall complexity.

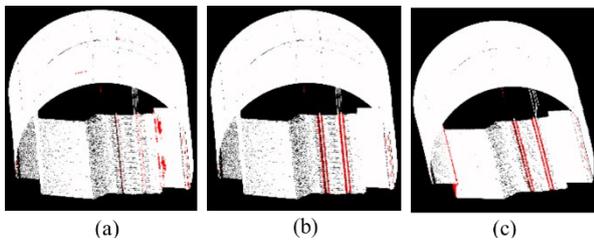


Figure 11. Misclassifications of the predictions in a manually labelled point cloud. (a) KPConv with transfer learning, (b) KPConv retrained, (c) PointNet.

It is also interesting to compare the performance of the two KPConv models presented in this work. Over the whole test set, the model trained from scratch outperforms the model pretrained with weights from the Semantic3D dataset. However, when the labels are refined in the rail area, the number of false positives increases considerably, what shows a clear symptom of overfitting: Considering the results in Table 8, the F1-score of the retrained model drops to 56.51% while for the model with transfer learning increases slightly to 88.58%. This means that the transfer learning model relies more on the geometry learned within its pretrained weights, thus avoiding overfitting. Therefore, we could assume that this model would perform better for semantic segmentation of new railway tunnels with different geometries.

To sum up, all three models have shown acceptable performances when comparing with automated, heuristic processes. However, a finer segmentation (as distinguishing among different types of wires) would likely require a heuristic post-processing.

5. CONCLUSIONS

This work presents the application of two deep learning models, PointNet and KPConv, for the semantic classification of 3D point clouds from railway tunnels. While PointNet was a pioneering work and it is well-known in the literature, KPConv is an architecture that has recently reached the state of the art. The models were trained with data from two different railway tunnels which were labelled combining an automated, heuristic approach in a first place, and a subsequent semi-automated refining process. Four classes were defined: Ground, lining, wiring and rails.

A KPConv and a PointNet model were trained for scratch using part of the case study data. In addition, a second KPConv model was considered following a transfer learning approach, which used pretrained weights from a model trained with the Semantic3D dataset.

Results showed that KPConv clearly outperforms PointNet. However, it is observed that, when trained from scratch, the KPConv model tends to present overfitting, which is not as present in the transfer learning model. Therefore, the KPConv model with transfer learning is expected to provide better results in different tunnel typologies and therefore provide the most valuable segmentations among the explored models.

As future work, different tools are to be developed in order to improve the digitalisation of the infrastructure and its assets. Taking into account that point clouds are already considered as a tool for infrastructure monitoring, it is expected to use the results of their processing to create and complete infrastructure information models (IIM). These models are born following the standards created for building modelling (BIM), and so, international Open Standards should be used when merging the information extracted from point clouds with the data models.

Thus, the next step should be the definition of a methodology that starting with point cloud data, creates a data model containing not only geometric and semantic data but also parametric information.

ACKNOWLEDGEMENTS

This work has used computational resources from the Supercomputing Centre of Galicia (CESGA).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769255. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

REFERENCES

- Arastounia, M., 2017. An Enhanced Algorithm for Concurrent Recognition of Rail Tracks and Power Cables from Terrestrial and Airborne LiDAR Point Clouds. *Infrastructures* 2, 8. doi.org/10.3390/infrastructures2020008
- Arastounia, M., 2015. Automated recognition of railroad infrastructure in rural areas from LIDAR data. *Remote Sensing*, 7, 14916–14938. doi.org/10.3390/rs71114916
- Arastounia, M., Oude Elberink, S., 2016. Application of template matching for improving classification of urban railroad point clouds. *Sensors*, 16. doi.org/10.3390/s16122112
- Balado, J., Martínez-Sánchez, J., Arias, P., Novo, A., 2019. Road environment semantic segmentation with deep learning from

- MLS point cloud data. *Sensors*, 19, 3466. doi.org/10.3390/s19163466
- Che, E., Jung, J., Olsen, M.J., 2019. Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors* 19, 810. doi.org/10.3390/s19040810
- Chen, L., Jung, J., & Sohn, G., 2019. Multi-Scale Hierarchical CRF for Railway Electrification Asset Classification From Mobile Laser Scanning Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8), 3131-3148. doi.org/10.1109/JSTARS.2019.2918272
- Erdelj, M., Król, M., Natalizio, E., 2017. Wireless Sensor Networks and Multi-UAV systems for natural disaster management. *Computer Networks*, 124, 72-86. doi.org/10.1016/j.comnet.2017.05.021
- European Commission, 2019. Transport in the European Union. Current Trends and Issues.
- Guan, H., Li, J., Cao, S., Yu, Y., 2016. Use of mobile LiDAR in road information inventory: A review. *International Journal of Image and Data Fusion*, 7(3), 219-242. doi.org/10.1080/19479832.2016.1188860
- Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017. Semantic3D.net: A new Large-scale Point Cloud Classification Benchmark. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-1/W1, 91-98, doi.org/10.5194/isprs-annals-IV-1-W1-91-2017
- Hackel, T., Stein, D., Mairdorfer, I., Lauer, M., Reiterer, A., 2015. Track detection in 3D laser scanning data of railway infrastructure. *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. IEEE, pp. 693-698. doi.org/10.1109/I2MTC.2015.7151352
- Jung, J.; Chen, L.; Sohn, G.; Luo, C.; Won, J.-U., 2016. Multi-Range Conditional Random Field for Classifying Railway Electrification System Objects Using Mobile Laser Scanning Data. *Remote Sensing*, 8, 1008. doi.org/10.3390/rs8121008
- Kingma, D.P., Ba, J.L., 2015. Adam: a Method for Stochastic Optimization. *Int. Conf. Learn. Representations* 1-15. arXiv:1412.6980
- Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution on X-transformed points. *Advances in neural information processing systems* p. 820-830.
- Liu, W., Sun, J., Li, W., Hu, T., Wang, P., 2019. Deep learning on point clouds and its application: A survey. *Sensors* 19(19), 4188. doi.org/10.3390/s19194188
- Lou, Y., Zhang, T., Tang, J., Song, W., Zhang, Y., Chen, L., 2018. A Fast Algorithm for Rail Extraction Using Mobile Laser Scanning Data. *Remote Sensing*, 10(12), 1998. doi.org/10.3390/rs10121998
- Luo, C., Jwa, Y., Sohn, G., 2014. Context based multiple railway object recognition from mobile laser scanning data. In *2014 IEEE Geoscience and Remote Sensing Symposium*, Quebec City, QC pp. 3602-3605. doi.org/10.1109/IGARSS.2014.6947262
- Ma, L., Li, Y., Li, J., Wang, C., Wang, R., Chapman, M.A., 2018. Mobile laser scanned point-clouds for road object detection and extraction: A review. *Remote Sensing*, 10(10), 1531. doi.org/10.3390/rs10101531
- Puente, I., González-Jorge, H., Riveiro, B., Arias, P., 2013. Accuracy verification of the Lynx Mobile Mapper system. *Optics & Laser Technology*, 45, 578-586.. doi.org/10.1016/j.optlastec.2012.05.029
- Qi, C.R., Su, H., Mo, K.K., Guibas, L.J., 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77-85. doi.org/10.1109/CVPR.2017.16
- Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA
- Sánchez-Rodríguez, A., Riveiro, B., Soilán, M., González-deSantos, L.M., 2018. Automated detection and decomposition of railway tunnels from Mobile Laser Scanning Datasets. *Automation in Construction*, 96, 171-179. doi.org/10.1016/j.autcon.2018.09.014
- Sánchez-Rodríguez, A., Soilán, M., Cabaleiro, M., Arias, P., 2019. Automated inspection of Railway Tunnels' power line using LiDAR point clouds. *Remote Sensing* 11(21), 2567. doi.org/10.3390/rs11212567
- Soilán, M., Lindenbergh, R., Riveiro, B., Sánchez-Rodríguez, A., 2019a. PointNet for the automatic classification of aerial point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-2/W5, 445-452. doi.org/10.5194/isprs-annals-IV-2-W5-445-2019
- Soilán, M., Riveiro, B., Balado, J., Arias, P., 2019b. Comparison of heuristic and deep learning-based methods for ground classification from aerial point clouds. *International Journal of Digital Earth*, 1-20. doi.org/10.1080/17538947.2019.1663948
- Soilán, M., Sánchez-Rodríguez, A., del Río-Barral, P., Perez-Collazo, C., Arias, P., Riveiro, B., 2019c. Review of Laser Scanning Technologies and Their Applications for Road and Railway Infrastructure Monitoring. *Infrastructures* 4(4), 58. doi.org/10.3390/infrastructures4040058
- Teledyne Optech, 2019. URL www.teledyneoptech.com/en/home/ (28 June 2019).
- Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 6410-6419. doi.org/10.1109/ICCV.2019.00651
- Ye, X., Li, J., Huang, H., Du, L., Zhang, X., 2018. 3D recurrent neural networks with context fusion for point cloud semantic segmentation, in: *Lecture Notes in Computer Science (LNCS)*, vol. 11211 (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). doi.org/10.1007/978-3-030-01234-2_25