https://doi.org/10.2352/EI.2023.35.17.3DIA-100

Few-Shot Learning on Point Clouds for Railroad Segmentation

Abdur Razzaq Fayjie, Patrick Vandewalle; KU Leuven; Belgium

Abstract

Semantic segmentation on 3D point clouds can guide infrastructure maintenance of complex environments, such as railroads. The existing fully-supervised segmentation models rely on a large collection of annotated data and often fail to generalize on dynamic *railroad environments when the infrastructure changes. To attenuate these limitations, we propose a multi-prototype few-shot point cloud segmentation method for railroad environments. Our model leverages the geometric features of point clouds to learn to represent each object class in multiple prototypes. We evaluate our model on real-world data collected over railway tracks in Belgium. The experimental results show that our model achieves promising results on complex data distributions of railroad infrastructures.*

Introduction

The railway is one of the major modes of transportation globally. In 2020, the EU was seen 147 billion passenger kilometers of passenger transport, 338 billion tonne-kilometers of freight transport, and 19 million twenty-foot equivalent units of container transport. Belgium, one of the member countries in the EU with considerable railway infrastructure, invested 320 million euros in 2019 for infrastructure maintenance[1]. These statistics exhibit the importance of automated railway maintenance. Semantic segmentation plays a vital role in scene understanding and monitoring and can be applied to automated infrastructure monitoring for railways. Several Building Information Modeling (BIM) based models have been proposed to digitize the railway infrastructure for segmentation. These models are highly characterized by industry standards and interoperability. However, these models lack a straightforward way to incorporate geometric information. Conversely, mobile laser scanners can record raw 3D point cloud data to incorporate geometric information of the railway infrastructure with high-speed trains. Based on the point cloud data, heuristic-based [3, 15, 17], PCA-based [1], and SVM-based [6] methods for railway segmentation have been developed.

In recent years, the advancement of deep learning has motivated several fully-supervised semantic segmentation models for railway tracks [9, 14]. However, these models heavily rely on a large collection of annotated data for training. Moreover, the train and test set data are drawn from the same space for these models. Conversely, complex real-world environments such as railroads are dynamic, and a new class of infrastructure may appear after training. Collecting and annotating a large-scale dataset for railroads is expensive, error-prone, and laborious. Fully-supervised models often fail to generalize to an unseen class of infrastructure with limited annotations. To alleviate these issues, Few-shot learning employs a few labeled examples to learn a model, which is oblivious to unseen classes at test time [12, 8, 4, 10]. In this direction for 3D point clouds, several works [2, 7, 16] have been proposed. These works employ carefully designed publicly available datasets to evaluate the models, which usually fail to reflect the model's ability to deal with complex real-world complex environments.

In this work, we develop a novel few-shot segmentation method for railroad infrastructures that aims to segment an unlabeled point cloud, called a query sample for a novel class, given a few annotated samples, called support samples from the same class. Our work is based on prototypical networks [8], which represents each class with a prototype by averaging the embeddings of support samples. Following the work by Zhao N. *et al*. [16], we define multiple prototypes for a single class to better capture the complexity of the data distribution, raised due to the geometric structures of correlated points. To capture the geometric structure of correlated points, we employ DGCNN [13] as our backbone network. In summary, our contributions are as follows,

1. We propose a few-shot segmentation model on 3D point clouds for railway infrastructure. Our model leverages dynamic graph networks to capture the geometric information from the 3D point clouds for complex structures.
2. We evaluate our model on the real-world dataset for railroad segmentation, collected over various railway tracks in Belgium.

Related Work

Fully-supervised models for railroad segmentation. Soilan *et al*. employ two popular point cloud networks, PointNet [5], and KP-Conv Net [11] for railroad segmentation to segment ground, lining, wiring, and rails in two different railway tunnels. FarNet [14] aggregates spatial attention to feature information using an attention module to learn from the spherical projection of point clouds.

Few-shot learning on point clouds. Chen X. *et al*. [2] proposes a multi-view few-shot segmentation method that leverages multi-view representations of point clouds to build compositional prototypes for a single object class. Zhao N. *et al*. [16] proposes an attention-aware mechanism for multi-level feature representations. Unlike single prototypes for each class in prototypical networks [8], this work represents each class with multiple prototypes to capture large variations of geometric structure within the same class. In our work, we adapt this formulation in designing prototypes for our object classes. Sharma and Kaul [7] propose to utilize unlabelled point cloud data in a self-supervised way that employs a cover tree for hierarchical partitioning of point

Methodology

Problem Formulation

We follow the standard notations to define our few-shot segmentation problem over three sets, a training set $\mathcal{D}_{tr} = \{p_i^t, y_i^t\}_{i=1}^{N_{tr}}$,

[1]source: https://data.oecd.org/transport/passenger-transport.htm.

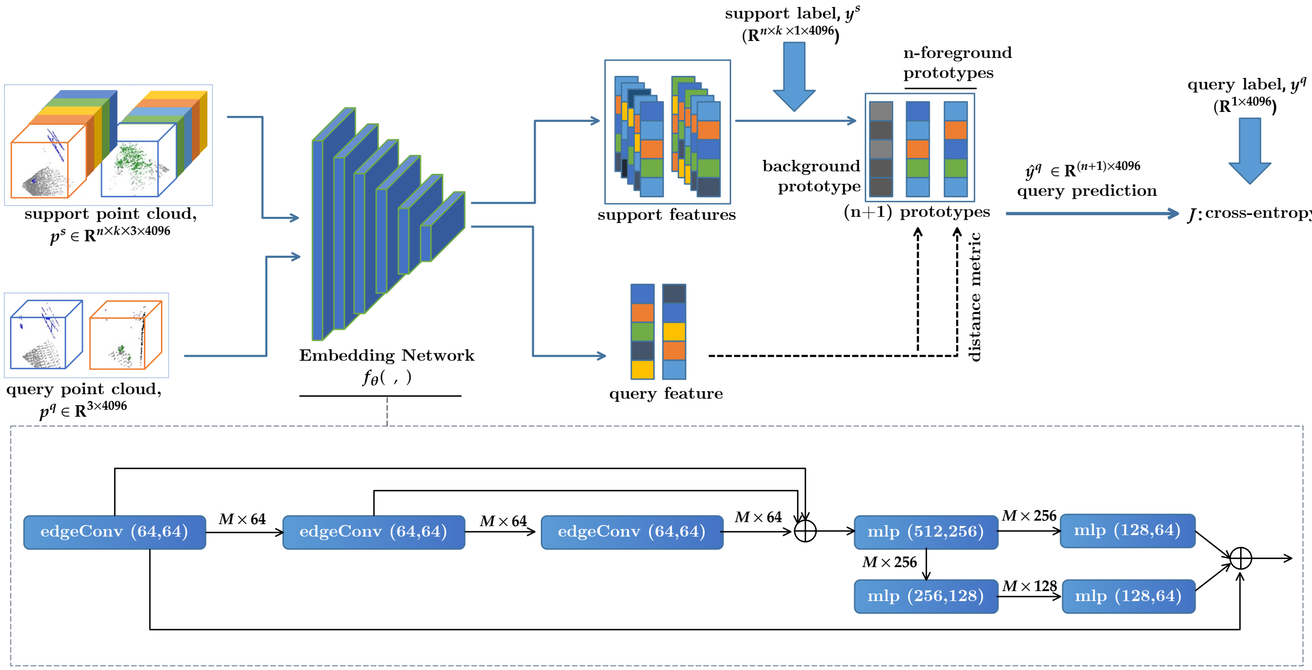

Figure 1. Our architecture for few-shot point cloud segmentation on railroad environments. In this figure, we illustrate a 2 - way 5 - shot setting, i.e. n = 2, k = 5.

a support set $\mathcal{D}_{sup} = \{p_i^s, y_i^s\}_{i=1}^{N_{sup}}$, and a test set $\mathcal{D}_{te} = \{p_i^q\}_{i=1}^{N_{te}}$. In this setup, $p_i \in \mathbb{R}^{3\times 4096}$ denotes an input point cloud of 2x2 meters cropped from a larger area, and $y_i \in \{0,1\}^{1\times 4096}$ denotes the corresponding mask for the input point cloud. N represents the number of point cloud samples in each set, characterized by subscript tr, sup, and te for the train, support, and test set, respectively. The test classes, denoted by $C_{te} \subset C$ are shared with the support classes C_{sup} but disjoint with train classes, C_{tr}, *i.e.* $C_{tr} \cap C_{te} = \emptyset$.

During training, a neural network $f_\theta(.)$ parameterized by θ is learned on $\mathcal{D}_{tr}$ for segmentation task. Once trained, the network is inferred to segment a previously unseen class, $c \subset C_{te}$ from $\mathcal{D}_{te}$, given a K annotated point cloud samples from $\mathcal{D}_{sup}$. To replicate this inference mechanism during training, we apply the episodic training paradigm [12]. Assuming a $C-way$, $K-shot$ few-shot segmentation task, we construct each episode by sampling, a) a support training set, $\mathcal{D}_{tr}^{\mathcal{S}} = \{p_t^s, y_t^s(c_i)\}_{i=1}^{k} \subset \mathcal{D}_{tr}$ for each class $c_i \in C_{tr}$, and (b) a query training set, $\mathcal{D}_{tr}^{\Omega} = \{p_t^q, y_t^q(c_i)\} \subset \mathcal{D}_{tr}$. Here, p_s^t, and p_t^q denote the support and query point clouds with y_t^s and y_q^t being the corresponding ground truths, respectively, for a sampled class $c_i \subset C_{tr}$.

Embedding Network for Feature Extraction

Figure 1 presents our architecture for few-shot point cloud segmentation on railroad environments. We employ dynamic graph networks, specifically, DGCNN [13] to capture the contextual geometric information of correlated points. The *edgeconv* layers of DGCNN exploits the point-wise correlations for local geometric features. The MLP layer of the network produces global output features that represent the semantic information. We apply a concatenation operation to fuse the local and global point cloud features to construct the features for the support and query samples. These features are represented in a manifold space, learned by a distance function over the support prototypes and the query.

Class Prototypes and Distance Function

For a $C-way$, $K-shot$ segmentation task, we average the support features to obtain C foreground prototypes and one background prototype. We exploit Euclidean distance and cosine distance as the learnable distance function to measure the similarities of support prototypes and the query.

Training Objective

Given a support set, $\mathcal{D}_{tr}^{\mathcal{S}}$ and a query $p_t^q \in \mathcal{D}_{tr}^{\Omega}$, the network $f_\theta(\mathcal{D}_{tr}^{\mathcal{S}}, p_t^q)$ is trained to predict the label distribution, $\hat{y}_t^q \in \mathbb{R}^{M\times(c+1)}$ for p_t^q, that aims to find the optimal parameters θ^* as for the model as follows,

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(\mathcal{D}_{tr}^{\mathcal{S}}, \mathcal{D}_{tr}^{\Omega})} \left[\sum_{i=0}^{n_{it}} \mathcal{J}(y_t^q, f_\theta(\mathcal{D}_{tr}^{\mathcal{S}}, p_t^q) \right], \tag{1}$$

where $\mathcal{D}_{tr}$ is the training set and $\mathcal{J}$ is the loss function, iterated over the n_{it} iteration.

Experiments

Dataset and setup.

The dataset used in our work is collected by Infrabel[2]. We refer to this dataset as the *Infrabel-5 Railroad Segmentation dataset*. The data is collected over railway tracks all over Belgium. A Z+F 9012 lidar sensor is mounted on the front part of the train to record the point cloud data. The data is manually annotated using the Cloud Compare[3] annotation tool. To our best knowledge, there is no publicly available railroad point cloud dataset for segmentation. The Infrabel-5 Railroad Segmentation dataset contains eight

[2]Infrabel is a Belgian government-owned public limited company that builds, owns, maintains, and upgrades the Belgian railway network, makes its capacity available to railway operator companies, and handles train traffic control.

[3]CloudCompare is a 3D point cloud processing software.

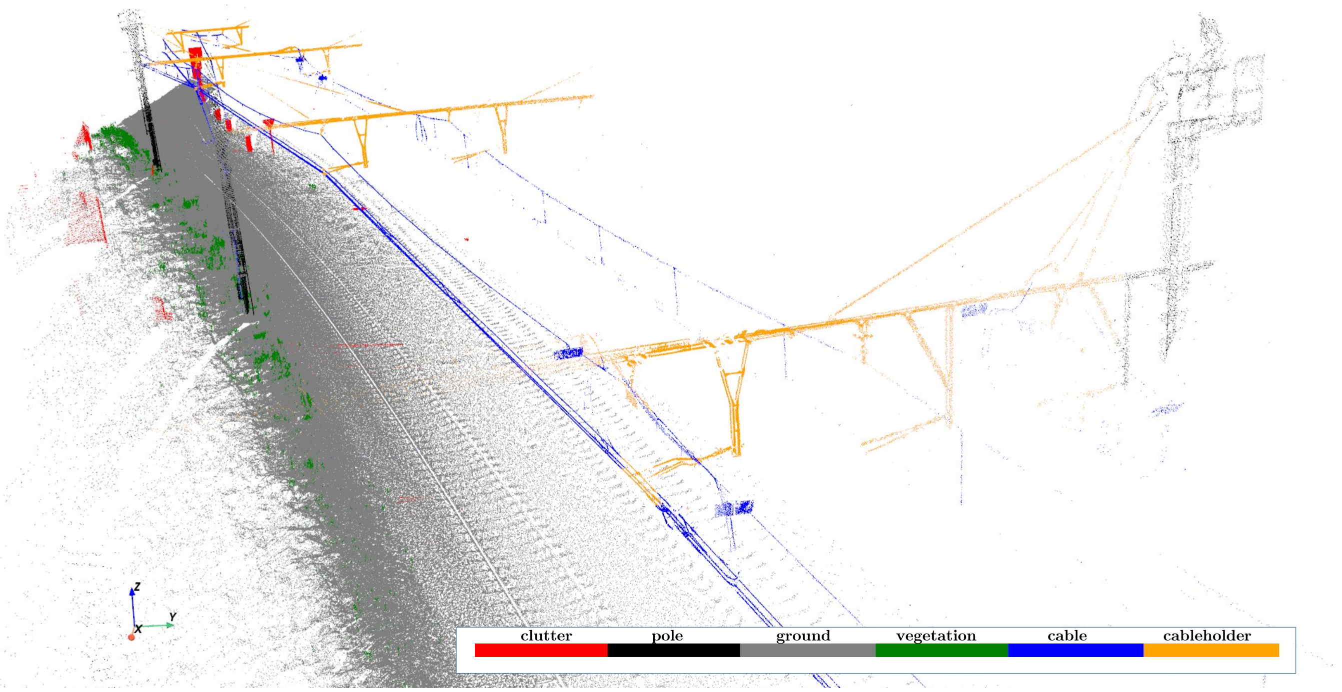

Figure 2. *Infrabel-5 Segmentation Dataset.*

large-scale point clouds, each of approximately eight million raw points in 3D space collected from eight railway tracks of different cities. First, we preprocess the data by filtering out the outliers. It results in comparatively smaller point clouds, approximately four million points per railway track. We present our data in figure 2.

The Infrabel-5 Railroad Segmentation dataset contains primarily five categories; cable, cable holder, ground, vegetation, and pole. The additional clutter category contains all the points not classified as one of the defined classes. The dataset represents highly imbalanced data, presenting the real-world complex data scenario. To adapt this dataset into our few-shot setting, we first split the dataset into two sets; a training set and a test set, where the training set does not contain the classes from the test set. We define three class-wise settings for experimental evaluations of our model in Table 1. In each setting, we consider different combinations of train-test (C_{tr}-C_{te}) classes.

Class-wise settings for experimental evaluations of our model.

Setting	C_{tr}	C_{te}
1	cable holder, cable, pole	ground, vegetation
2	ground, vegetation, pole	cable holder, cable
3	cable holder, pole, vegetation	ground, cable

To fit the data into our limited GPU memory, we apply a sliding voxel. The voxel is of size 2×2 meters in the XY plan. The Z dimension remains constant over a large area of railway tracks and is considered to be the maximum height of a point in the large area represented in 3D. We slide this voxel to extract the points from a large area of point clouds into a 2×2 meters of the ground plan. Then, we randomly sample 4096 points from each of the 2×2 voxels.

Implementation Details. Figure 1 presents the segmentation network with the support and query point cloud samples. Based on DGCNN, each edge-convolution layer of our network is followed by batch normalization and a leaky ReLU. We leverage the publicly available point cloud segmentation dataset, s3DIS to train our network following an episodic training paradigm. In this phase, we employ Adam optimizer with a learning rate of 1e-3. We apply L2 regularization, with a value 1e-4.

The trained network is then considered for the downstream few-shot segmentation task on the Infrabel-5 Railroad Segmentation dataset. In training, we design each episode by sampling K annotated point clouds of 2x2 meters from $\mathcal{D}_{tr}^{\mathcal{S}}$ for the network to predict the label distribution of a query sampled from $\mathcal{D}_{tr}^{\mathcal{Q}}$. We apply Adam optimizer with a learning rate of 1e-3. However, we reduce the learning rate by 0.5 in each 5K iterations. Moreover, we employ an L2 regularizer with a value of 1e-4. The trained network is then evaluated on a query point cloud $p_i^q \in D_{te}$, given K support examples from $\mathcal{D}_{sup}^{\mathcal{D}}$. In both training phases, we choose cross-entropy loss as the objective function, $\mathcal{J}$ in eq. 1. The training is carried out in an Nvidia Tesla V-100 GPU with 32 GB graphics memory. We train the network for 30K iterations.

Evaluation Metric. We employ mean Intersection over Union (mIoU) to evaluate our model performance. We average mIoU over all the test classes to report the final performance of the model.

Results

Quantitative Result. Table 2 and 3 present the experimental results of our model on 1-way 5-shot and 1-way 1-shot settings,

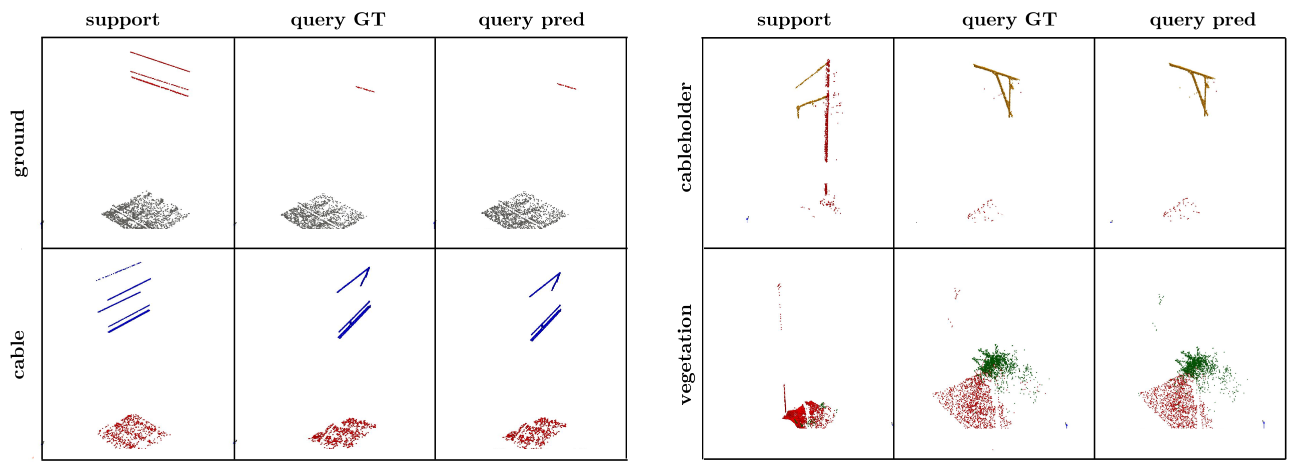

Figure 3. A qualitative analysis of our segmentation in $1-way\ 5-shot$ setting. We depict four different classes, ground, cable, cableholder, and vegetation with grey, blue, orange, and green color, respectively. The red depicts the background class in each case.

respectively. We employ two distance functions, euclidean distance, and cosine distance to measure the similarity between the support prototypes and the query. From the results, we see that cosine distance produces a better segmentation result, 1-3% for the last two class-wise settings. Furthermore, the 1-way 5-shot setting achieves 7.5% better mIoU in the last two class-wise settings while 9.5% better mIoU in the first class-wise settings. The results manifest that the segmenting vegetation from the ground results in comparatively a lower mIoU than the segmenting cables and grounds. It is due to the complex but connected geometric structure of ground and vegetation, where finding an optimal segmentation plan from limited annotated data is a complex problem. This problem may also arise due to the considerable amount of ground data points compared to vegetation in our imbalanced dataset.

Segmentation results on 1-way 5-shot setting in with three different class-wise settings. The evaluation is reported in mIoU.

Setting	cosine distance	euclidean distance
1	67.66	66.71
2	79.69	78.49
3	92.86	90.54

Segmentation results on 1-way 1-shot setting in with three different class-wise settings. The evaluation is reported in mIoU.

Setting	cosine distance	euclidean distance
1	56.67	56.82
2	72.19	69.25
3	85.50	82.20

Qualitative Result. We depict a visual segmentation for $1-way$ $5-shot$ for railroad segmentation in figure 3. We present a support point cloud (columns 1,4), the query point cloud (columns 2,5), and the prediction over the query point cloud (columns 3,6) in each row for four classes, namely, ground (first row, left), cable (second row, left), cableholder (first row, right), and vegetation (second row, right), respectively. These classes are represented in grey, blue, orange, and green respectively, whereas the background class is depicted in red. From the figure, we can observe that the segmentation is accurate for the disconnected background classes. For example, ground and cables with each other as the background class (first and second row, left) give an accurate segmentation. On the other hand, connected classes such as the vegetation with the ground as the background class generates some false positives. From this visualization, it is clear that the segmentation result is highly dependent on the setting of the classes.

Conclusion

In this work, we have proposed a few-shot segmentation model for complex railroad environments. Our model leverages the dynamic graph networks, specifically DGCNN, to extract global and local geometric features. We further derive multiple prototypes from the support features for each object class to accurately model the complex data distribution. In future work, we plan to investigate various existing embedding networks to compare the performance of our model in railroad segmentation. We are also interested in exploring multi-view prototypes.

Acknowledgement

We would like to thank Infrabel for supporting us in data collection and preparing the Infrabel-5 Railroad Segmentation dataset.

References

[1] Mostafa Arastounia. “Automated recognition of railroad infrastructure in rural areas from LiDAR data”. In: *Remote Sensing* 7.11 (2015), pp. 14916–14938.

[2] Xiaoyu Chen et al. “Compositional prototype network with multi-view comparision for few-shot point cloud semantic segmentation”. In: *arXiv preprint arXiv:2012.14255* (2020).

[3] Sander Oude Elberink and Kourosh Khoshelham. “Automatic extraction of railroad centerlines from mobile laser scanning data”. In: *Remote sensing* 7.5 (2015), pp. 5565–5583.

[4] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. Lille. 2015, p. 00.

[5] Charles R Qi et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[6] A Sánchez-Rodrıéguez et al. "Automated detection and decomposition of railway tunnels from Mobile Laser Scanning Datasets". In: *Automation in Construction* 96 (2018), pp. 171–179.

[7] Charu Sharma and Manohar Kaul. "Self-supervised few-shot learning on point clouds". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7212–7221.

[8] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[9] M Soilán et al. "Semantic segmentation of point clouds with pointnet and kpconv architectures applied to railway tunnels". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (2020), pp. 281–288.

[10] Flood Sung et al. "Learning to compare: Relation network for few-shot learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208.

[11] Hugues Thomas et al. "Kpconv: Flexible and deformable convolution for point clouds". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6411–6420.

[12] Oriol Vinyals et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems* 29 (2016).

[13] Yue Wang et al. "Dynamic graph cnn for learning on point clouds". In: *Acm Transactions On Graphics (tog)* 38.5 (2019), pp. 1–12.

[14] Zhangyu Wang et al. "FarNet: An Attention-Aggregation Network for Long-Range Rail Track Point Cloud Segmentation". In: *IEEE Transactions on Intelligent Transportation Systems* (2021).

[15] Bisheng Yang and Lina Fang. "Automated extraction of 3-D railway tracks from mobile laser scanning point clouds". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.12 (2014), pp. 4750–4761.

[16] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. "Few-shot 3d point cloud semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8873–8882.

[17] Lingli Zhu and Juha Hyyppa. "The use of airborne and mobile laser scanning for modeling railway environments in 3D". In: *Remote Sensing* 6.4 (2014), pp. 3075–3100.

Author Biography

Abdur R. Fayjie received his Master of Engineering in Mechanical Engineering from Kunsan National University (2018). Currently he is a researcher at KU Leuven on 3D computer vision.