

ON THE SEMANTIC SEGMENTATION AND VALIDATION OF ELECTRICAL SUBSTATIONS

M. Bassier*

Dept. of Civil Engineering, TC Construction - Geomatics
KU Leuven - Faculty of Engineering Technology
Ghent, Belgium
maarten.bassier@kuleuven.be

Commission II / WG 4

KEY WORDS: Semantic segmentation, Scan-vs-CAD, Modeling, Deep learning, point clouds, Remote sensing

ABSTRACT:

Converting deep learning methods from benchmark testing to real applications is highly sought after both in academia and the industry. Key challenges that remain are the performance of the methods on new datasets, the preprocessing of the data and the integration of the results into application pipelines. Specifically for the implementation of semantic segmentation procedures, each of these challenges are still very much the subject of research. In this paper, we present a testcase to digitally twin and validate an electrical substation. Concretely, we discuss the data processing, training and the follow up integration of the results in the validation pipeline. In the experiments, we show that 86% initial F1-score can be achieved using the proper transfer learning on 14 classes and that this results in a 97% recall on the validation and 80% recall on the digitization of the substation. Overall, we show that the segmentation significantly contributes to these processes and that they are absolutely necessary for the automation of the digital twinning.

1. INTRODUCTION

Semantic segmentation procedures are becoming increasingly potent to process large-scale point cloud data. Specifically in the Architectural, Engineering and Construction industry, these semantically segmented point clouds are necessary to better create, analyze and validate construction digital twins. Current processes still require labor-intensive human interpretation of the raw point cloud data. Moreover, nuanced spatial analyses are currently unachievable because the point clouds are not segmented. A prominent task is the validation of digital twins with respect to the as-built conditions (Patraucean et al., 2015). As a metric for the accuracy of the model, the Euclidean distance is observed between both datasets, and the mean or standard deviation is reported for the distances up to a cut-off distance (Jadidi et al., 2015). This analysis is negatively impacted by stray points that do not belong to the object and thus produce misleading result. Analogue, automated modeling of geometric objects such as beams, pipes, walls, etc. require a clear delineation of the object boundaries to not produce false positives (Bassier, Vergauwen, 2020).

The semantic or instance segmentation is currently achieved through deep learning algorithms. Given sufficient training data, these models can predict class labels or segment object instance in new point clouds. These class labels are then used to separate the point cloud into its respective components that can be used for validation, modeling or analysis. Several deep learning architectures show very promising results for the generalization of various benchmark point cloud data. However, key obstacles still remain to integrate these networks into industry pipelines. A first aspect is the training data preparation, which includes the production of relevant and sufficient known observations

for each class. While some class imbalances have been tackled within the networks themselves, these methods have their limits and thus a careful data preparation is required. A second aspect is the data structuration of the input point clouds including the density, region of influence and potential features that all heavily impact the training and detection performance. Even a slight deviation from the benchmark data can lead to a reduction of 10-20% mIoU (De Geyter et al., 2022). A third aspect is the post-processing of the results and the integration of the segmented data into existing pipelines. There are different strategies that can be used to deal with false positives or outliers in the detection results, which can have a significant impact on the final application.

This research discusses each of the above aspects in the case study of an electrical substation. In summary, the main contributions are:

1. A literature study on the obstacles and solutions for deep learning adoption
2. An empirical study of the adaption of state-of-the-art deep learning for custom datasets
3. A practical case study to embed semantically segmented point clouds into a validation pipeline

The remainder of this work is structured as follows. The background and related work is presented in Section 2. In Section 3, the methodology is presented. The experiments results are discussed in Section 4. Finally, the conclusions are presented in Section 5.

*Corresponding author

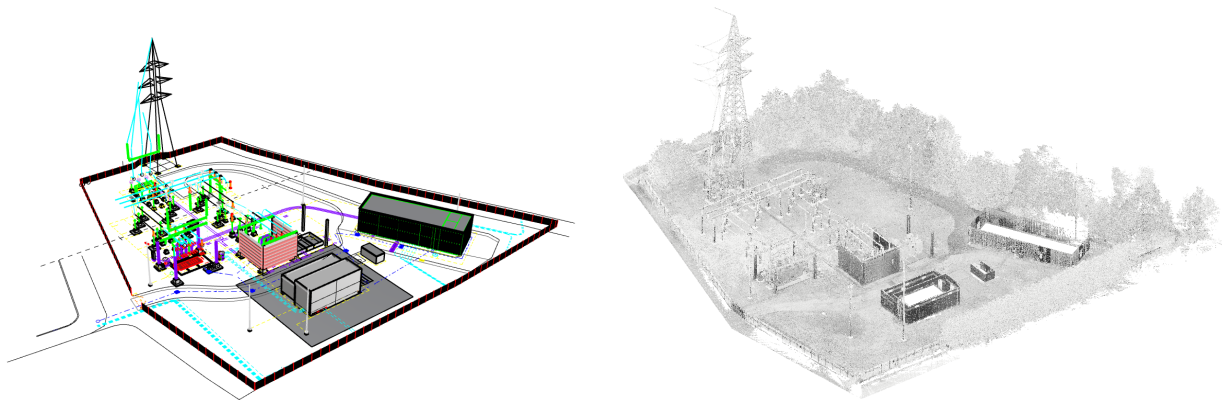


Figure 1: Overview of the 3D data of an electrical substation: (left) CAD model with several thousand elements used for project planning and maintenance and (right) terrestrial laser scanning point cloud with circa 90 million points.

2. BACKGROUND AND RELATED WORK

2.1 Deep learning methods

There currently are three popular methods for the semantic segmentation of point cloud data i.e. projection-based methods, volumetric methods and point-wise segmentation methods (Guo et al., 2019). Projection-based methods transform the 3D data into a series of 2D raster similar to images. These methods leverage the recent advancements in image processing and achieve very good segmentation results on objects that are captured entirely within the field-of-view i.e. small-scale objects such as building or street furniture. Additionally, the computational complexity of these methods $O(n^2)$ is significantly lower than their 3D counterparts. As a result, image processing networks can be larger which aid the interpretation of the information. However, projecting point clouds comes at a steep cost (Wu et al., 2023). Significantly more 2D rasters are needed of the same scene than point clouds. This is especially true for point clouds that are produced by structure-from-motion pipelines which have up to 90% overlapping imagery. More importantly, projection-based methods often fail to correctly segment objects that span across multiple images e.g. both sides of a wall. Overall, these methods are best suited to segment objects that fit within a single raster and that have strong texture signatures. SOA methods such as UNetFormer (Su et al., 2015) and SVQNet (Liu et al., 2016b) achieve over 80% mIoU on these types of data but are typically not proposed for unstructured point cloud data.

Volumetric methods offer a good alternative as they retain the fixed rasterization, but operate more directly on the 3D data. Methods such as OctNet (Riegler et al., 2017) and VoteNet (Ding et al., 2020) partition the 3D space into a series of regular voxels. The 3D rasterization yields an $O(n^3)$ complexity over the input space and thus these networks tend to be smaller. Additionally, larger cuboids are needed to encompass larger spaces and thus volumetric methods typically struggle with high density point clouds. Irregular octrees and data tiling are proposed to remedy this shortcoming i.e. OctNet requires less memory and runtime for high-resolution point clouds. Hybrid approaches are also proposed such as PointGrid (Le et al., 2018) that combine both point and grid representations for efficient point cloud processing. Overall, these methods work best when there are irregular point densities, and low scene detailing that allow for significant downsampling.

Point-wise segmentation methods are currently the preferred technique as these operate directly on the point cloud. The point cloud is batched into a fixed number of points, that is then fed to the network. Similar to the volumetric methods, these methods have an $O(n^3)$ complexity but retain the detailing of the point cloud. This is a promising technique, as long each batch contains sufficient information about the scene, which can be problematic in high-density point clouds. Methods such as PointNet (Qi et al., 2016) and KPConv (Thomas et al., 2019) batch the input point clouds into small blocks of $1 \times 1 \text{m}$, making it challenging to interpret larger objects. Recent networks use nearest neighbors i.e. RandLA-Net (Hu et al., 2020) and Point Transformer (Zhao et al., 2021) can process over 40k points per batch, which scales better with varying point density. Overall, these methods work best when there is sufficient detailing in each batch of the point cloud such as in complex indoor environments.

2.2 Training datasets

Industry projects do well to align themselves with online point cloud benchmarks as novel methods are designed and tested on these. Point cloud benchmarks can be divided into roughly three topics. Aerial datasets such as Vaihingen and Potsdam are characterized by their low but consistent density e.g. 25 points/m² and detailing given the constant flight altitude. They are usually structured in 2D regions or depth maps. Typical classes include vegetation and various man-made structures such as buildings, utility lines, roads and sometimes smaller objects such as cars or fences. As further downsampling is strongly discouraged and voxel-based methods do not typically perform well on these datasets. Projection-based methods work very well, since the combination with high density imagery uncovers a lot of missing details and all objects are within the field-of-view of the image. Point-based methods also work well due to the relatively consistent detailing in the point cloud due to its very low resolution (Thomas et al., 2019). As a result, the point batches contain sufficient context and generalize well.

The second type of datasets are urban mobile mapping or navigation datasets such as SemanticKITTI, NuScenes and Paris-lille-3D. These datasets vary in density depending on the proximity of the surroundings to the sensor. These datasets are either structured by per frame panoptic point clouds or in regions and typically include one or more streets. Typical classes include vehicles, pedestrians, the street and some man-made structures. Projection-based methods again work very well in this case as

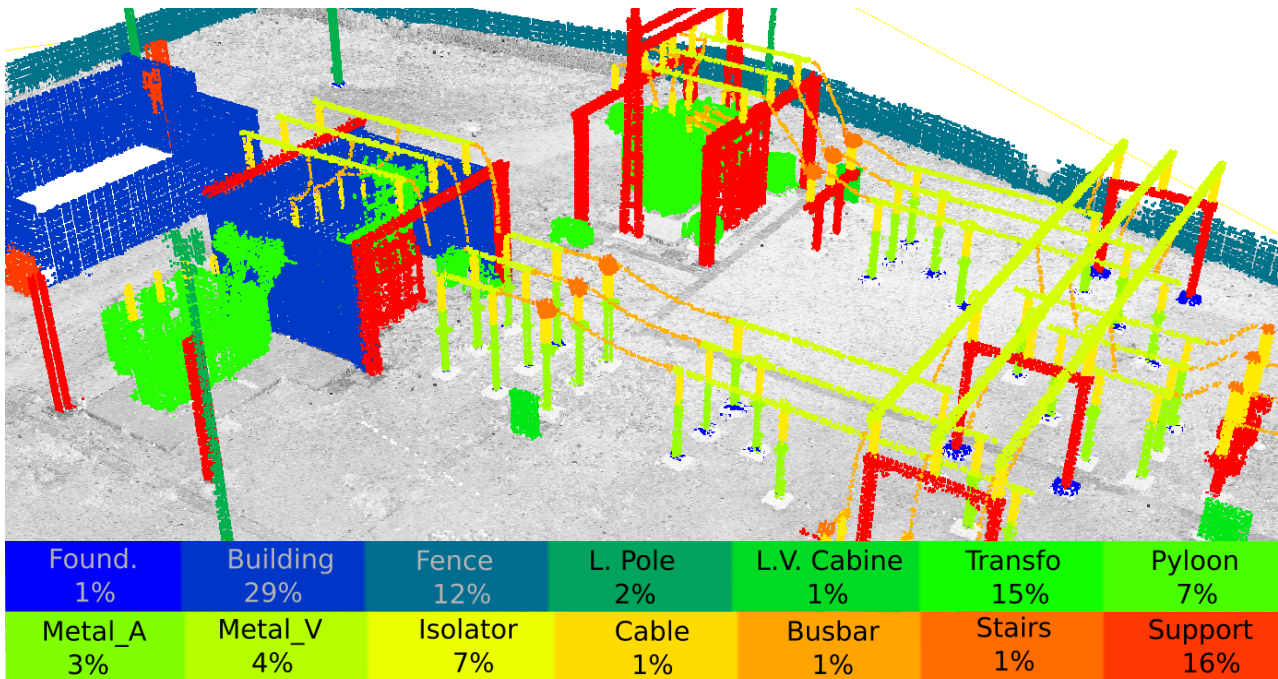


Figure 2: Overview of the ground truth labels of the 14 classes with the % of points per class in the electrical equipment.

illustrated by SSD (Liu et al., 2016a) and SphereFormer (Lai et al., 2023), especially in combination with image data. Analogue to the aerial datasets, most objects are observed within the field-of-view except for the street and buildings. However, the concatenation of these observations does not yield much additional information since only the facade and street deck are observed anyway. Voxel-based methods perform well due to the repetitiveness in the scenes but downscaling can pose problems from small-scale objects such as pedestrians (Tang et al., 2020). If the point clouds are unstructured, point-wise classification methods have the best performance (Liu et al., 2021).

The third type are indoor or terrestrial datasets such as S3DIS, Semantic3D and ScanNet. These datasets are high-density and typically organized per frame (RGBD) or per room. This last one is a challenge as industry projects do not have this separation. Typical classes include various structure, architecture and furniture elements. All three methods perform quite well on rooms i.e. on S3DIS, DeepViewAgg (Robert et al., 2022) (74.7% projection), Swin3D-L (Yang et al., 2023) (79.8% volumetric), StratifiedTransformer (Lai et al., 2022) (77.6% point-wise) all have top performances. If the datasets were unstructured, it is expected that the latter two will yield better performance.

2.3 Post-processing

Two typical procedures in digital twinning are validation and digitization/ modeling (Bonduel et al., 2017). In the former, the remote sensing inputs are evaluated to assess whether all objects are placed on site in the correct location. Typical validation techniques include Euclidean distance evaluation, collision detection and object detection and segmentation techniques (Son et al., 2015). The former two are unintelligent and work well for any object that is captured with sufficient point detailing and has a high geometric resemblance to its digital counterpart. The latter attempts to detect individual objects such as through instance segmentation. This works well for well-trained object

classes but is very difficult in new projects. A key issue to overcome is the inherent differences between the digital twin and the point clouds due to abstractions, modeling differences, relative positioning, visibility, etc. A common technique is to perform a localised registration between both geometries to improve the distance correlations i.e. through iterative closest point (ICP) variants or global registration pipelines (Bassier et al., 2020).

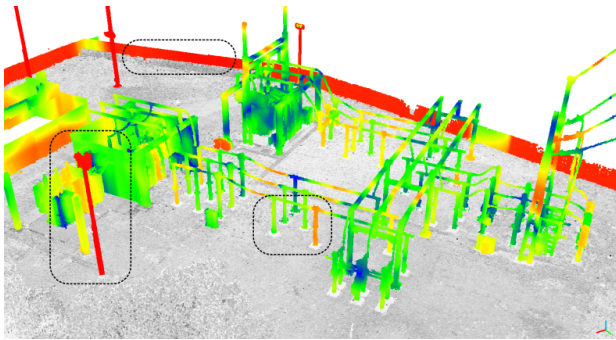
In digitization or modeling, the remote sensing inputs are evaluated whether all objects have a digital counterpart and model. This is typically performed by the above distance evaluations in combination with an instance or semantic segmentation to determine the type of object that is missing. Once the observations of an object are then isolated, it can be placed or modeled in-situ. These operations have the same problems as with the validation procedures. Typically, a two steps procedure is proposed where first a set of candidate partial geometries is proposed, after which the final geometries and their connections are modeled (Bassier, Vergauwen, 2020).

3. METHODOLOGY

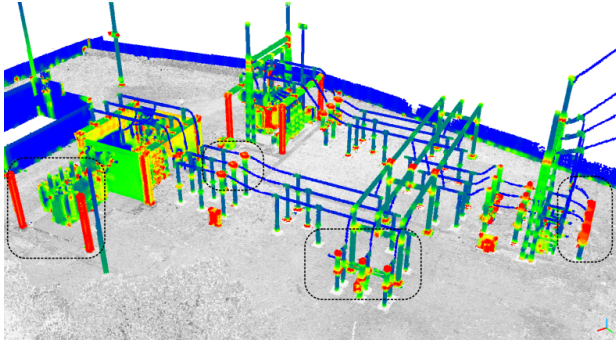
3.1 Dataset modalities

To devise a proper segmentation and validation workflow, we first identify the characteristics of the substation and the point cloud data. The substation used in this study is an Air Insulated Substation (AIS). There are over 300 of these stations in Flanders, Belgium and they are documented for project planning and maintenance. The substations consist of thousands of standardized elements including cables, transformers, voltage cabins, pylons, support structures and so on (Figure 1). The scope of this study is limited to the visible exterior elements so the buildings interior is not considered. Overall, each element can be well observed with remote sensing and there is few clutter on the site.

The substation is mapped using terrestrial laser scanners and thus has a varying density. In this study, the point clouds are



(a) Linearity (6m) is distinctive for lanterns and fences in contrast to other vertical structures.



(b) Sphericity (0.2m,0.3m,0.6m) is distinct for various cables, busbars and isolators in contrast to other elements.

Figure 3: Overview of the covariance features used in PointTransformer.

considered unstructured. Note that no color was captured as most objects are an indistinguishable gray or black. The overall density of the 90M point cloud is circa 1point/cm which is needed to validate the numerous elements on site. However, the data distribution for the different classes is dramatic. 51% is vegetation, 35% is terrain and only 14% actually belongs to the target equipment. 14 equipment classes are identified but these lack class balancing as shown in Figure 2, with less than 1% of the data belonging to the foundations, low voltage cabins, cables, busbars and stairs.

The CAD of the substation is a combination of vector and block geometries. Over 32000 geometries in 76 layers are present including mostly polylines, meshes, solids, points, hatches and text (Table ??). Model differences include the presence of abstract geometries i.e. schematic powerlines, invisible geometries under ground, wire-frame blocks, simplified geometries and so on (Figure 4). Moreover, some blocks definitions, layers and equipment classes do not align perfectly i.e. a block can have nested objects from multiple classes and so do certain objects in the same layer. Overall, the CAD was designed for project planning and operations, and in its native form, it is not directly usable for validation.

3.2 Semantic segmentation

Ideally, an instance segmentation is used for the interpretation of the point cloud data that suits the needs of the validation and digitization tasks. However, this is completely infeasible in new projects with hundreds of unique elements with varying sizes and geometric/texture signatures without excessive training data. Instead, a semantic segmentation is proposed of 14 equipment classes that helps identify object types on site.

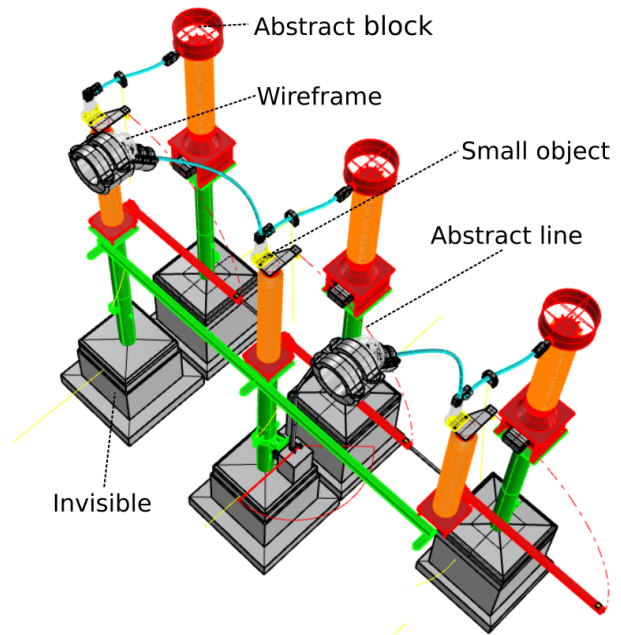


Figure 4: Overview of CAD vs PCD deficiencies.

The segmentation is performed in two stages. First, the vegetation and ground points are detected using commercial software. Second, a point-based semantic segmentation network is trained on the manually created samples of the unstructured point cloud. The reasons for picking a point-based network are that the unstructured data with significant detailing best fits with these types of methods. Additionally, the density of the point clouds varies greatly depending on the sensor setup. Finally, both large and small objects are present, which are subideal to process with voxel of projection-based approaches.

We consider Point Transformer (Zhao et al., 2021) for the semantic segmentation as it is a recent network that generalizes well over similar unstructured benchmark datasets including S3DIS and Semantic3D. Additionally, we have altered the Point Transformer implementation so it can be enriched with additional features i.e covariance features such as linearity, which allows the encoding of properties larger than the data tiling of a network. For instance, the lantern poles are challenging to discern from cylindrical steel columns but they are significantly taller. A linearity property within a search space of 6m will highlight the lantern poles while both classes will have the same properties in the network that most likely does not considered points within 6m as only 40k points are selected per batch. Point Transformer both with and without features is trained for 300 epochs in a cross-validation setting on the complete dataset with the appropriate class-balancing countermeasures i.e. data augmentation, perturbations, etc. Table ?? and Figure 3 show the features used and their signatures.

3.3 Validation & Digitization

CAD preparation The CAD geometries are converted to Open3D geometries for the evaluation. Two approaches are considered: (1) the CAD is parsed directly in python using the ezdxf DXF library. However, nested block definitions proved problematic especially in combinations with abstract geometry types. BODY, 3DSOLID, SURFACE, REGION objects also require the proprietary ACIS SDK from Spatial Inc.¹. (2) A Rhino Grasshop-

¹<https://www.spatial.com/products/3d-acis-modeling>

CAD		Open3D	
Blocks	960	0_Foundation	28
Points	139	1_Building	76
Curves	27000	2_Fence	12
Surfaces	195	3_L_Pole	4
PolySurfaces	2017	4_LC_Cabine	4
Meshes	1567	5_Transfo	73
Text	36	6_Pyloon	65
Hatches	58	7_Metal_A	421
		8_Metal_V	186
		9_Isolator	82
		10_Cable	549
		11_Busbar	15
		12_Stairs	39
		13_Support	798
Total	32686	Total	2352

Table 1: Overview of CAD objects and converted Open3D geometries.

Features	
Linearity (0.5m,6m)	$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planarity (0.5m)	$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1}$
Sphericity (0.2m,0.3m,0.6m)	$S_\lambda = \frac{\lambda_3}{\lambda_1}$
Omnivariance (0.5m)	$O_\lambda = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$
Anisotropy (0.1m)	$A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1}$
Sum of Eigenvalues (0.6m)	$\Sigma_\lambda = \lambda_1 + \lambda_2 + \lambda_3$
Change of curvature	$C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$
Height	Z
Intensity	I

Table 2: Overview of CAD objects and converted Open3D geometries.

per² script is developed where a set of simple routines can be used to automatically export only relevant geometries as polygonal meshes. The second approached is heavily favored as it provides better control over the evaluated geometries.

Validation To minimize the errors caused by relative and schematic placement of the CAD objects, we first propose a local registration between both geometries. Note that the registration of individual objects would be prone to misassociations with nearby objects. Analogue, a linear transformation for the entire site is prone to the same placement errors. Instead, we perform a k-nearest neighbor selection on the CAD objects so only a local region is transformed (Figure 5). Both a global FPFH and local ICP transformation were investigated but the ICP proved significantly better in most cases. This is due to the FPFH features between the CAD and PCD being suboptimal due to model abstractions. Instead, the larger but more simplistic parts are typically the same in both geometries and are favored by ICP variants.

The validation itself is formulated as follows. An object is considered present if a significant portion of its geometry lies close

²<https://grasshopper.app/>

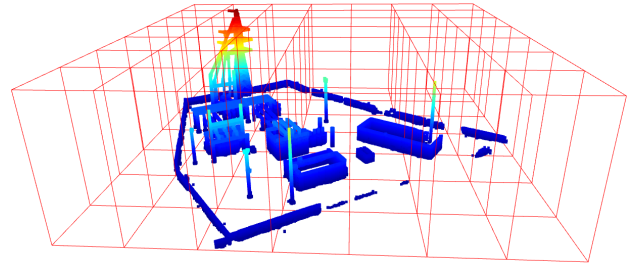


Figure 5: Overview point cloud subdivision for the localised registration.

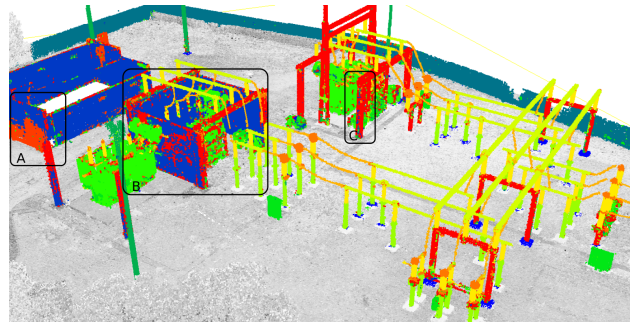


Figure 6: Overview of the semantic segmentation results and problems: Confusion between (A) stairs and building, (B&C) Metal-v and Building.

to a portion of the point cloud that has a matching class. To this end, the CAD meshes are sampled with the same density as the input point cloud P . The resulting reference point cloud Q is used to count the inliers in the point cloud (Eq. 1,2).

$$Q' = \left\{ q_j \mid q_j \in Q, p_i \in P_i : \underset{q_j}{\operatorname{argmin}} \|p_i - q_j\| \leq t_d \right\} \quad (1)$$

$$\text{presence} = \begin{cases} 1, & \frac{|Q'|}{|Q|} \geq t_c \\ 0, & \text{else} \end{cases} \quad (2)$$

where $y_{p_i}, y_{q_j} \in \zeta = \{\text{transformator}, \text{ground}, \text{etc.}\}$. This proximity can be further expanded with additional conditioning on the normals of the points but yielded no better results. The result is a binary classification of the CAD objects whether they are present or not.

Modeling The modeling is considered a two-step procedure. First the point cloud is filtered to isolate the observations of missing objects. Second, their corresponding CAD representations are computed and added to the model. For the first step, we reverse the distance evaluation in Eq.1. The condition $y_{p_i} = y_{q_j}$ is not kept under the assumption that two objects cannot coexist in the same space even if they have different classes (Figure 7a). As such, P' solely contains new geometries (Eq. 3).

$$P' = \left\{ p_i \mid p_i \in P_i, q_j \in Q : \underset{p_i}{\operatorname{argmin}} \|p_i - q_j\| \geq t_d \right\} \quad (3)$$

The members of P' are then clustered using Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The result is a set of point clusters P' per class that surpass a minimum threshold of points t_p and with a spanning distance d_{min} (Figure 7b).

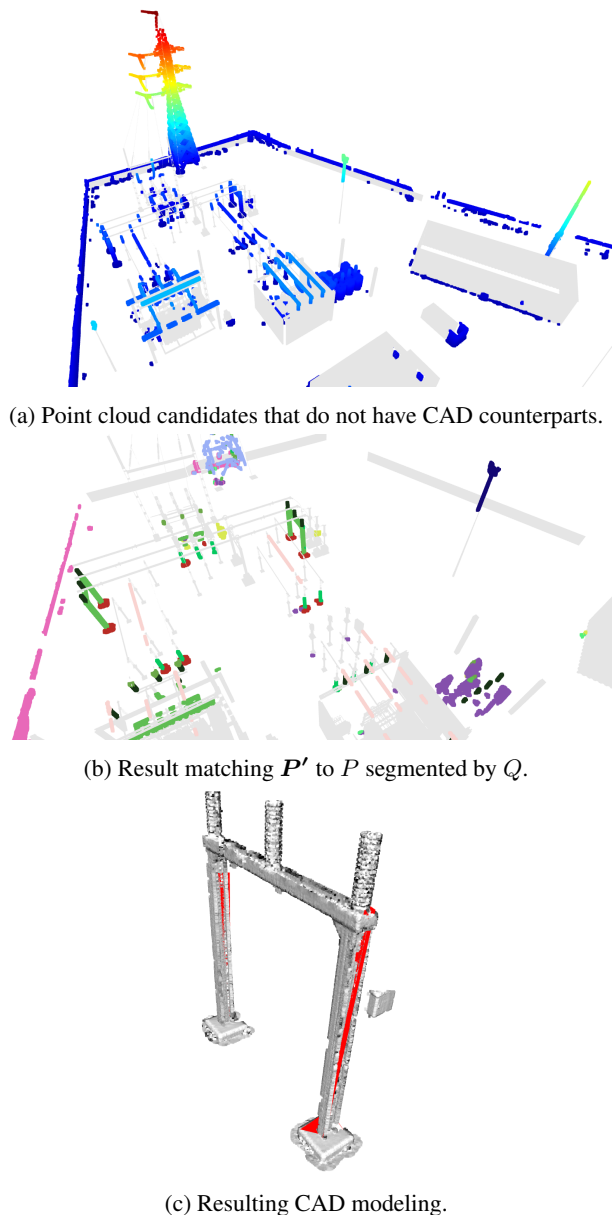


Figure 7: Overview of the modeling pipeline.

For the second step, we associate the newly formed clusters in P' with segmented clusters from P that do have a match in Q to determine whether there is similar element that is scanned on the site. We specifically use P as the source for the association instead of Q to mitigate the numerous model abstraction errors that also prevent the global registration pipeline to succeed. Instead, we propose the use of FPFH features in a global pipeline to assess the similarity between members in P' and the segmented P . However, for all but the best matches, we leave it up to the user to decide which object should be modeled at a certain location. For the clusters that do find a match, we use the transformation parameters from the global registration as the new insertion point of the detected CAD object (Figure 7c).

4. RESULTS AND DISCUSSION

For the inspection, the CAD was restructured to a total of 2352 objects in 14 layers conform the detection classes. 88% (2074) objects have a valid Open3D TriangleMesh geometry and 70% of those geometries are considered present on site in their designed shape (loosely within half a meter tolerance) through manual inspection. This is low given that the site is fully operational but this is largely due to above described shortcomings of the TLS data and the CAD model. Note that the distribution of missing/improper objects also greatly varies depending on the size i.e. the 50% largest geometries have an average built status of 93%, while the lower half only have an average 67% built status. This is even more drastic for the 10% smallest objects that only have a confirmed presence of 25%.

For the development of the segmentation model, several configurations were testing with tiling and subsampling of the data. The results of the cross validation are reported in Table ?? . The best achieved result is 86% F1-score and 67.7% mIoU with Point Transformer with additional features which is fairly close to what is performed by this network on benchmark data (i.e. 73.5% mIoU on S3DIS by Point Transformer) (Table ??). This is impressive, even for a cross-validation, on a new dataset with 14 classes with 6 classes suffering from significant class imbalance. This is reflected in Figure 6, where well represented classes clearly outperform the underrepresented classes. However, the segmentation lacks consistency (error B and C) with stray point found inside of other classes. This potentially could be solved by running the results through a Conditional Random Field to enforce consistency. Additionally, some classes have poor class definitions such as the stairs, which can be improved by adding more data and splitting this class (error A).

For the validation, several evaluation criteria were tested including objects are considered present (A) if at least 1 point in the point cloud lies within t_d of their surface, (B) if no portion of the objects is further than t_d from the point cloud, (C) if a percentage of the object is within t_d of the point cloud and (D) if a percentage of the object is near its semantic counterpart in the point cloud (Table ?? and Figure 8). The distance threshold $t_d = 0.2m$ and the % threshold to 70%. The unintelligent methods A en C scored well while B underperformed. There is an 7% increase in detection by using the semantically segmented point clouds. However, misclassifications also lead to a slightly lower precision rates for method D. Note that most errors are also objects that are very hard to assess by a manual operator due to the limitations of the point cloud.

For the digitization, the DBSCAN minimum number of points was set to 500 and the spanning distances to 0.2m. 326 additional clusters were found of varying sizes (although most are relatively small), with the most clusters belonging to the Pylon (105) and fence (51) classes. There are 300 potential CAD objects to choose from (that have more than 400 points). For the majority of clusters, there are only 1-2 CAD objects with a relatively similar size ratio. For the global feature registration, 30 nearest neighbors are considered for the FPFH features. In total, 32 (10%) new objects could be confirmed and modeled on site, indicating a strict selection procedure. This is to be expected due to severe abstractions of the objects and the occlusions in the point cloud. A strict selection is also desirable as a modeler can easily fill in the remainder of the classified objects.

The outliers of both the validation and digitization clearly show a relationship between the size (and thus density) of an observed

	Ground	Building	Fence	Pole	LV. Cabin	Transf.	Pylon	Metal-a	Metal-h	Isolator	Cable	Busbar	Stairs	Support
Point Transformer	65.6	79.5	91.3	71.8	30.6	66.4	71.9	59.8	88.6	81.7	61.5	28.6	31.1	58.2
Point Transformer (f)	68.9	80.5	88.4	78.6	32.1	73.3	83.4	67.3	91.2	89.6	66.2	32.0	31.5	64.6

Table 3: IuO scores of the different methods on the substation dataset.

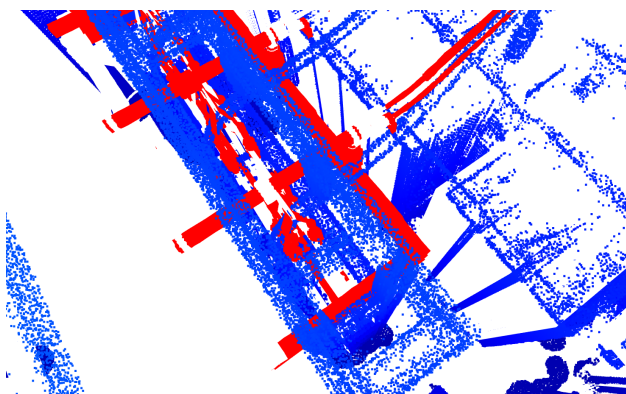


Figure 8: Overview of the validation through segmentation: (red) objects close to the point cloud (blue) that are not built and can only be detected through semantic segmentation.

Validation		Digitization	
Total elements	2074	Total elements	406
Built elements	1454	Modeled elements	33
Method A		Matching	
Recall	91%	Recall	80%
Precision	80%	Precision	92%
Method B		Modeling	
Recall	74%	Recall	15%
Precision	79%	Precision	66%
Method C			
Recall	90%		
Precision	79%		
Method D			
Recall	97%		
Precision	74%		

Table 4: Overview of validated and digitized CAD objects.

entity and whether it is detached from other elements. Missing small objects nearby other objects of the same class are extremely difficult to detect, even by a manual operator. Most outliers are due to positioning and model abstractions. This can be lowered by changing the thresholds but the recall vs precision ratio is strictly application dependent. Overall, the CAD would significantly benefit from aggregating multiple small objects in larger blocks, especially if their presence is correlated.

5. CONCLUSION

This paper presents a framework to adapt deep learning semantic segmentation in industry pipelines such as validation and digitization procedures. The presented methods discuss the training of such networks and their impact on the validation and digitization of an electrical substation. The goal of this research is to investigate to which extent deep learning methods can be

employed in current workflows and what their automation potential is. The main contribution is the adoption of a network using a single new dataset that does not resemble any of the benchmark datasets, and leverage it in an automated industry procedure.

The experiments show that current deep learning networks can indeed be transferred to industry projects and that they can significantly contribute to the automation of such tasks. Specifically for the validation and digitization of an electrical substation, we state that the semantic segmentation improves the procedure and makes it more nuanced. For the digitization, it is more important since it is a key step towards the automated modeling of the objects. However, some errors still remain. Especially for smaller objects with weak geometric signatures or objects that have significant clutter, the distance evaluation is flawed. An instance segmentation, perhaps with addition of images to support the detection, could improve the detection. However, such a method would require extensive amounts of training data which is costly and may be challenging to generalize as well.

In future work, we will explore to which extent images can contribute to validation and digitization frameworks and how synthetic training data can be generated from existing digital twins to lower training costs for instance segmentation methods.

ACKNOWLEDGEMENTS

This project has received funding from the FWO Postdoc grant (grant agreement: 1251522N) and the Geomatics research group of the Department of Civil Engineering, TC Construction at the KU Leuven in Belgium.

REFERENCES

- Bassier, M., Vergauwen, M., 2020. Unsupervised reconstruction of Building Information Modeling wall objects from point cloud data. *Automation in Construction*, 120(November 2019), 1–20. <https://doi.org/10.1016/j.autcon.2020.103338>.
- Bassier, M., Vincke, S., de Winter, H., Vergauwen, M., 2020. Drift invariant metric quality control of construction sites using BIM and point cloud data. *ISPRS International Journal of Geo-Information*, 9(9).
- Bonduel, M., Bassier, M., Vergauwen, M., Pauwels, P., Klein, R., 2017. Scan-To-Bim Output Validation: Towards a Standardized Geometric Quality Assessment of Building Information Models Based on Point Clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W8(November), 45–52. <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W8/45/2017/>.
- De Geyter, S., Vermandere, J., De Winter, H., Bassier, M., Vergauwen, M., 2022. Point Cloud Validation: On the Impact of Laser Scanning Technologies on the Semantic Segmentation for BIM Modeling and Evaluation. *Remote Sensing*, 14(3), 1–20.

- Ding, Z., Han, X., Niethammer, M., 2020. Votenet +: An Improved Deep Learning Label Fusion Method for Multi-Atlas Segmentation. *Proceedings - International Symposium on Biomedical Imaging*, 2020-April, 363–367.
- Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2019. Deep learning for 3D point clouds: A survey.
- Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. Randla-Net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11105–11114.
- Jadidi, H., Ravanshadnia, M., Hosseinalipour, M., Rahmani, F., 2015. A step-by-step construction site photography procedure to enhance the efficiency of as-built data visualization: a case study. *Visualization in Engineering*, 3(1), 1–12.
- Lai, X., Chen, Y., Lu, F., Liu, J., Jia, J., 2023. Spherical Transformer for LiDAR-based 3D Recognition. *arXiv*. <http://arxiv.org/abs/2303.12766>.
- Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified Transformer for 3D Point Cloud Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022-June, 8490–8499.
- Le, V.-H., Vu, H., Nguyen, T. T., Le, T.-L., Tran, T.-H., 2018. Acquiring qualified samples for RANSAC using geometrical constraints. *Pattern Recognition Letters*, 102, 58–66. <https://doi.org/10.1016/j.patrec.2017.12.012>.
- Liu, K., Gao, Z., Lin, F., Chen, B. M., 2021. FG-Conv: Large-Scale LiDAR Point Clouds Understanding Leveraging Feature Correlation Mining and Geometric-Aware Modeling. *Proceedings - IEEE International Conference on Robotics and Automation*, 2021-May, 12896–12902.
- Liu, T., Huang, X., Ma, J., 2016a. Conditional Random Fields for Image Labeling.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C., 2016b. SSD: Single shot multi-box detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS, 21–37.
- Patraucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., Haas, C., 2015. State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29, pp.162–171.
- Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv preprint arXiv:1612.00593*. <http://arxiv.org/abs/1612.00593>.
- Riegler, G., Ulusoy, A. O., Geiger, A., 2017. OctNet: Learning deep 3D representations at high resolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, 6620–6629.
- Robert, D., Vallet, B., Landrieu, L., 2022. Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022-June, 5565–5574.
- Son, H., Bosché, F., Kim, C., 2015. As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey. *Advanced Engineering Informatics*, 29(2), 172–183. <http://www.sciencedirect.com/science/article/pii/S147403461500021X>.
- Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3D shape recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter, 945–953.
- Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S., 2020. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12373 LNCS, 685–702.
- Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., Guibas, L., 2019. KPConv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October, 6410–6419.
- Wu, X., Wen, X., Liu, X., Zhao, H., 2023. Masked Scene Contrast: A Scalable Framework for Unsupervised 3D Representation Learning. *arXiv*. <http://arxiv.org/abs/2303.14191>.
- Yang, Y.-Q., Guo, Y.-X., Xiong, J.-Y., Liu, Y., Pan, H., Wang, P.-S., Tong, X., Guo, B., 2023. Swin3D: A Pre-trained Transformer Backbone for 3D Indoor Scene Understanding. *arXiv*. <http://arxiv.org/abs/2304.06906>.
- Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2021. Point Transformer. *Proceedings of the IEEE International Conference on Computer Vision*, 16239–16248.