IfcVoxNet: 3D segmentation and classification of voxelized IFC building models with Deep Learning

Johan Luttun, johan@aecgeeks.com
AECgeeks

Thomas Krijnen, thomas@aecgeeks.com
AECgeeks

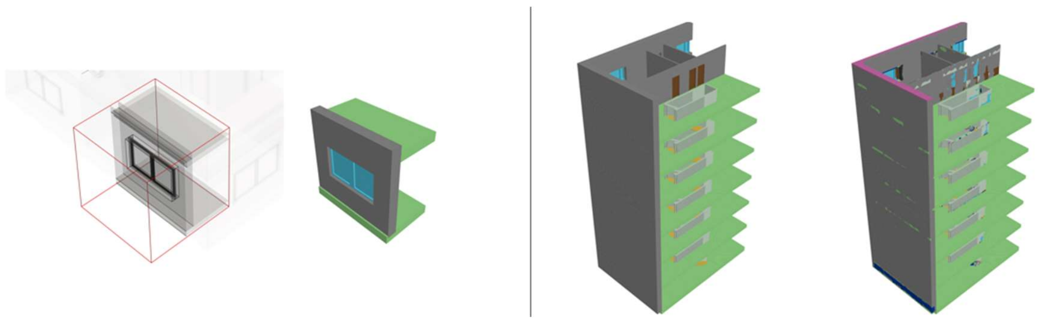

Figure 1. Left: 3D $64 \times 64 \times 64$ patch used for training. Right: ground-truth vs. model prediction on a bigger (never seen) 3D patch

Abstract
We propose a deep learning framework to segment and classify a voxel-based representation of IFC models.

While BIM-based building element classification has set promising avenues of research, most do not take into account the context of elements and they only learn on normalized shaped examples. We believe that element representations should be learned by providing neighboring information to the network. Some classes such as walls, columns and slabs could be very similar in a normalized input as they all can be boxes, but they are different classes that can be discriminated against from their surroundings, as a wall bounds spaces while a column only carries loads.

However, there is no idiomatic, standardized, nor universal way to decompose the physical building into a set of elements. It depends on preferences of the modeler, functionality of the tool, construction method and phase of the development. The later the phase, the more detailed and the more akin to how the building will be constructed as opposed to how it conceptually functions. Therefore, we present an approach to classify building elements based on a classification of individual voxels (cubic elements of mass on a regular grid obtained from the building element geometries) within the full building context. That means that the method is independent of individual modeler choices on ways of subdivision and can learn even features that are not an individual element, such as e.g typically a doorknob.

In this study we put a particular emphasis on the usability of the method in inference for use in applications. We argue that the voxelization step completely eliminates modeler choices as it is able to utilize the full building context and that a patch-based training approach benefits from advances in regularized 2D grid convolutional approaches.

Keywords: 3D segmentation, 3D deep learning, BIM, IFC, voxels

1 Introduction

1.1.1 Motivation

Building Information Modelling (BIM) is a relatively new paradigm in construction where information is exchanged in information models as opposed to traditional drawings. In such a model, the building is decomposed into a set of typed elements with geometric representations and various forms of data. The Industry Foundation Classes (IFC) are the predominant open standard to exchange such models.

The IFC schema does not impose a strict coupling between the type of a building element and its geometric form or nature or use in the building. Also the textual semantic definitions of building elements in the specification are relatively vague. As shown in Figure 2, a wall could be considered a column if its footprint is relatively short, or a railing if it does not run up to the ceiling and intends to prevent injury from falling, or a beam if it's the part above an opening.

As such, IFC models often contain inconsistently classified instances. While this problem can be rather subtle and still allows visualization and coordination use cases, it prevents automated flows to run smoothly by requiring manual corrections and compromises interoperability. To name a few examples, building permit evaluation requires consistent and reliable building element class labels (Noardo et al 2022) and automated computed aided manufacturing on discipline models.

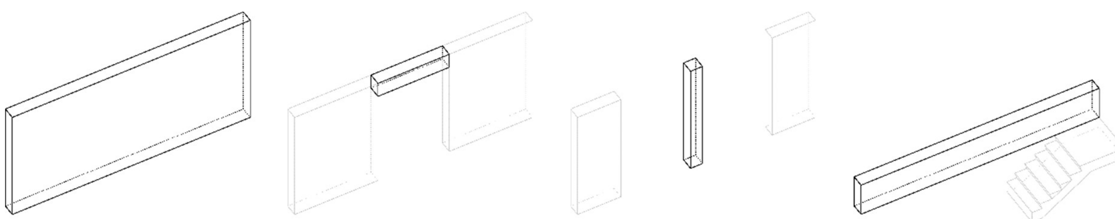

Figure 2. A wall and various elements typically drawn as walls that could also be characterized differently, as Beam, Column and Railing

To the best of our knowledge there exists no easy nor mainstream automated solution to solve this problem as an automated step in a validation process. In addition, we believe that given the difficulty of the task and the availability of data, it is a good candidate for a learning-based approach. Therefore, we want to take advantage of deep learning methods to learn how to use the information embedded in IFC models to participate in the solution to the problem, towards more reliable IFC models.

IFC models represent rich datasets, often under-exploited, where information about the building functional parts, its elements, and the relationships between each other can be modeled or derived. As a data schema, IFC allows to model building elements using the IfcBuildingElement abstract class which derives into instantiable classes such as IfcWall, IfcBeam, or IfcWindow. An IFC class models attributes and relationships as well as a physical representation of the element containing the description of its geometry. Although the IFC schema enables to incorporate several classification methods into an IFC instance, the IFC class already provides the classification of a particular instance.
In the context of Machine Learning and deep learning, this represents a dataset of labeled data from which supervised methods can be developed.

Machine Learning methods and in particular deep learning ones have proven remarkably effective for a broad range of tasks (Goodfellow et al 2016). Learning from 3D representations is no exception with works such as Choy et al (2016), Qi et al (2017), Zhou & Tuzel (2017) and Honocka et al (2019). However, 3D elements can have multiple data representations, which are the entry point of deep neural networks. When designing such a network, one needs to choose the appropriate representation given its problem. The volumetric representation is particularly interesting in the context of a building segmentation task, as it provides information about connectivity and fulfillment of a discretized space.

There is a high variability in 'idioms' of IFC usage based on – among others – the following factors: authoring tool and version; modeler choices; level of development, phase and intended use of the model; type of construction project (steel or concrete, residential or commercial, etc.), as shown in Figure 3. This has massive implications on the model as exchanged as a graph and mesh (decomposition versus singular representations, tremendous detail in geometric features that add detail such as fillets and chamfers) but are all in the end just ways of representing the same physical structure. The application of voxelization in this case is so interesting because it is agnostic to the amount of decomposition, modeler choices on where to break up elements as well as geometric detail.

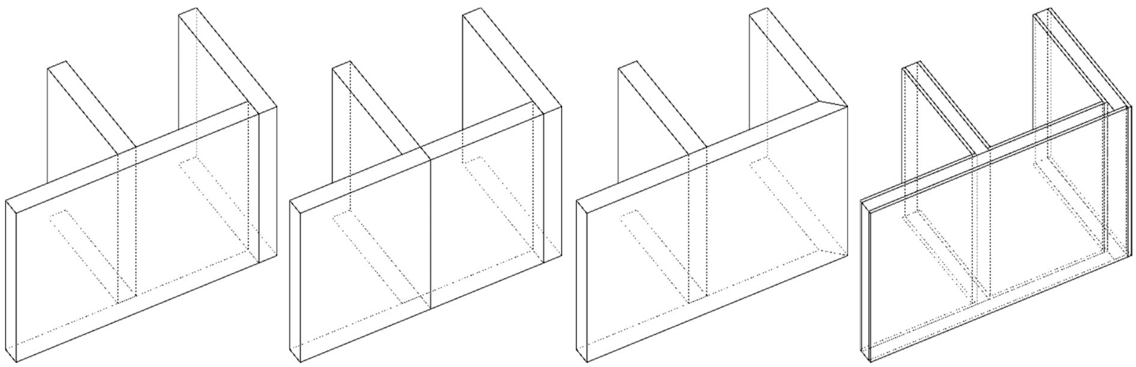

Figure 3. various realizations of the same wall configuration, resulting in different geometric forms and/or different topology

Voxel grids are one of the most rudimentary geometric forms to encode information. It can be derived from Boundary Representation (BRep), triangle meshes as well as point clouds. This means our approach can be used on projects with heterogeneous data or acquisition methods such as scan to BIM.

1.1.2 Related work

Krijnen & Tamke (2015) introduced element classification and anomaly detection by means of derived geometric quantities as an input signal using a simplistic neural network. This was later enhanced in Koo et al (2017) using Support Vector Machines, but still suffers from the same pitfalls that the geometric features the model operate on (surface area, volume, gyradius, etc.) are rather simplistic, not very descriptive of the element and susceptible to the different outcomes under different 'idioms' or level of development.

In Collins et al (2021), a graph representation of individual building elements is used for a classification task. It uses a graph neural network and obtains promising scores on classification of single elements.

Another example of deep learning applied to BIM models is Lomio et al (2018) where the method consists in classifying a building from screenshots of different angles using a pretrained ResNet (He et al 2015).

IFCNet (Emunds et al 2021) provides a valuable contribution by presenting a dataset of IFC elements extracted from IFC models and reviewed by domain experts. Providing such a benchmark constitutes an important step in the classification of BIM elements. The authors acknowledge the limitations of their contribution, especially the loss of information occurring when normalizing the elements.

In Emunds et al (2022), the authors use a point cloud representation of elements obtained from sampling points from a mesh representation. They present a tentative framework for the

integration of a neural network in a tool, and work from the learned representation for elements classification and retrieval tasks.
Luo et al (2023), follows-up on IFCNet, with an augmented IFCNet dataset called IFCNet++, which also includes IFC relationships as a feature vector concatenated to the embeddings of an element obtained from its multi-view representation.
Although the cited contributions offer promising opportunities for solving the misclassification problem, using different input signal representations, they only learn based on isolated elements. While they achieved excellent results on classification tasks, we believe such methods would lack robustness when presented with ambiguous cases. Moreover, the networks presented would only learn based on the intrinsic geometric properties of a class and its visual appearance, whereas a building element is also defined and distinguishable by its function within the building. This kind of learning makes the system very dependent on the dataset, as it will learn only from the shapes present in it.
Luo et al (2023) started to incorporate relationships information in their input data, which gives information about the number of connections an element has but does not provide information about how several instances can be related. In addition, the future interest stated in several of those works is to integrate the semantics of the IFC into the learning process, which limits the method to CAD generated models.

1.1.3 Contribution

In this paper, we wish to present a method for learning an efficient representation of building elements in their context for a segmentation task. The goal is to extract a volumetric representation of a part of a building model (patch) and feed it to a neural network whose output will be a 3D grid where each voxel would be assigned an IFC class among a set of predefined classes. On a practical level, this method gives the possibility to semantically check or enrich IFC models. From a research perspective point of view, we prove that it is possible to learn from occupancy grids, meaning that a volumetric representation offers a powerful starting representation for building elements. Our contributions are the following:

- Provide a method to voxelize IFC models and produce a segmented volumetric representation of building elements
- Train a neural network to segment a voxel grid derived from IFC elements
- Use the trained neural network in inference on unseen data

The starting representation for learning (3D patch) as well as an example of our trained model prediction on a large patch with its associated ground-truth are presented in Figure 1.

2 Method

2.1 Voxelization

2.1.1 Voxelization algorithm

A voxel grid is a 3D volumetric grid where regular discretized elements of mass as a field of values. In our research we start from the IFC geometry to first build a binary grid of zeros and ones for the respective building element classes under investigation. These are composed into a labelled occupancy grid where every voxel value corresponds to a single building element class. There is a global priority - roughly in line with the prevalence and expected physical dimensions of elements - that governs the final assignment of a cell in case of multiple classes occupying the same cell, which is fairly common due to the fact that a box-overlap algorithm is used for voxelizing the geometry.

We voxelize entire IFC models and retain specific classes. This provides us occupancy grids where each voxel has a class assigned. We obtain this discrete representation by using a voxelizer algorithm.

2.2 Representation choice for learning

2.2.1 Volumetric representation

The representation has a huge impact on the model performance and on solving the targeted task. We believe that a volumetric representation in the form of occupancy grids provides the benefits of connectivity as well as a clear visualization of classes. Even though it is computationally intensive (Zhou & Tuzel 2017), it keeps the essence of how building elements look like relative to each other: how they connect, how they are placed in the model, and how much relative space they take. In addition, it enables to extract the patches at specific locations giving flexibility on the examples that can be shown to the network.

2.2.2 Patch extraction and data augmentation

To train the model, we provide $64 \times 64 \times 64$ 3D patches extracted from the voxelized IFC files. The patching strategy enables to reduce the computational cost of training but also serves as a data augmentation step, because we can provide those patches “on-fly” during training. We keep for each voxelized building element its bounding box center and extract patches from this coordinate, at the desired patch size. We empirically found that $64 \times 64 \times 64$ was a convenient size to balance between computational cost and providing enough contextualization, that is enough different classes instances belonging to the patch. We also randomly apply transformations on the extracted patches such as flips or rotations.

2.2.3 Network architecture

We use a fully convolutional 3D UNet (Çiçek et al 2016)[1]. This network architecture can ingest patches of different sizes (typically multiple of 16), which provides flexibility for both the training and inference stages. Indeed, we can train on $64 \times 64 \times 64$ patches but infer on bigger or smaller patches depending on the use case. The 3D UNet is an extension of the well-known UNet (Ronneberger et al 2015) which has encountered many successes for image segmentation and more recently in generative tasks with diffusion models. The UNet is made of an encoder and decoder part, to which skip connections are added between high level features and lower-level ones to preserve details.

2.2.4 Training details

We trained the model for 47 epochs, with patches extracted from the 8 voxelized IFC files with batch size of 8 and the Adam Optimizer. We use the Cross-entropy loss as the cost function.

2.2.5 Dataset

The model has been trained on 8 publicly available models (Figure 4). Evaluation happens on 2 models not seen during training.

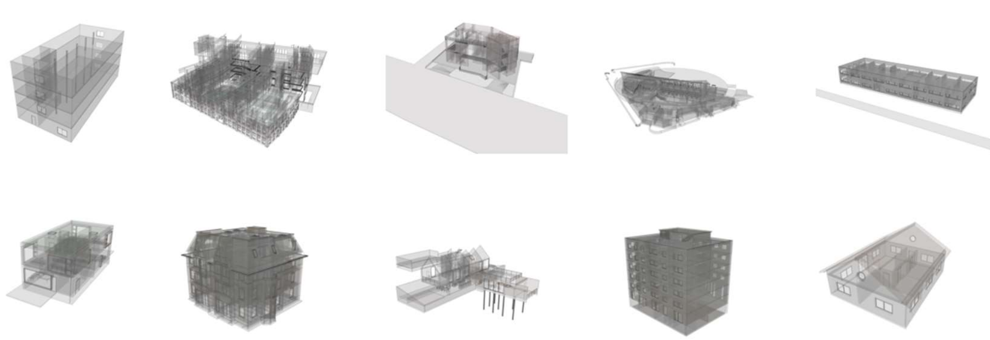

Figure 4. Wireframe view of the open-source IFC models used for training and testing

[1] Implementation taken from: *3D-UNet-pytorch*. [Online]. Available from: https://github.com/hanskrupakar/3D-UNet-pytorch.

Table 1. Open-source IFC models used for training and testing with no. of occurrences per IFC class

filename	IfcSlab	IfcWall	IfcBeam	IfcWindow	IfcDoor	IfcColumn	IfcRailing	IfcStairFlight	IfcFurnishingElement
train									
office_model_CV2_fordesign.ifc	8	33	0	8	9	10	0	0	0
https://openifcmodel.cs.auckland.ac.nz/api/download/20160414office_model_CV2_fordesign.ifc									
Architecture.ifc	24	1578	0	131	435	349	89	0	201
https://openifcmodel.cs.auckland.ac.nz/api/download/20210219Architecture.ifc									
Wellness center Sama.ifc	13	69	0	2	54	3	11	9	48
https://openifcmodel.cs.auckland.ac.nz/api/download/2022020320211122Wellness%20center%20Sama.ifc									
KT-ZCB (combined).ifc	279	549	552	233	97	146	55	25	1
https://openifcmodel.cs.auckland.ac.nz/api/download/20220221KT-ZCB%20(combined).ifc									
DigitalHub_FM-ARC_v2.ifc	23	178	14	47	64	62	13	10	0
https://github.com/RWTH-E3D/DigitalHub/blob/36565d529b4dadeca625de2b793d7e16700171e9/Version_2/DigitalHub_FM-ARC_v2.ifc									
Duplex_A_20110907_optimized.ifc	21	57	8	24	14	0	4	2	61
https://www.wbdg.org/bim/cobie/common-bim-files									
IFC Schependomlaan.ifc	279	930	174	259	205	23	90	0	0
https://github.com/buildingSMART/Sample-Test-Files/blob/3c73e7a664cc47e4129affb079bd4c656d29a98e/IFC%202x3/Schependomlaan/Design%20model%20IFC/IFC%20Schependomlaan.ifc									
rac_basic_sample_project.ifc	35	47	0	17	16	3	10	3	35
https://help.autodesk.com/view/RVT/2025/ENU/?guid=GUID-61EF2F22-3A1F-4317-B925-1E85F138BE88									
total	**682**	**3441**	**748**	**721**	**894**	**596**	**272**	**49**	**346**
test									
Molio_with_URIs.ifc	37	98	7	119	119	14	2066	564	38
https://github.com/buildingSMART/Sample-Test-Files/blob/d3ae2f11ed48a2e87b5dbcc6db14dbf197249153/IFC%202x3/Molio/Molio_with_URIs.ifc									
AC20-FZK-Haus.ifc	4	13	4	11	5	0	2	0	0
https://www.ifcwiki.org/index.php?title=File:AC20-FZK-Haus.ifc									
total	**41**	**111**	**11**	**130**	**124**	**14**	**2068**	**564**	**38**

3 Evaluation

To evaluate the performance of our method, we first proceed with a manual inspection of evaluating the trained network on patches from unseen IFC models. This highlights the kind of knowledge encapsulated in the network. Note that the examples highlighted in Figures 5, 6 and 7 were noteworthy with discrepancies between predicted and the original labels. These are not representative for the overall performance of the network.

Moreover, we apply the model on specific elements, by means of taking the patch at element center and taking the predominant predication for the component of voxels. This results in an element-wise confusion matrix useful for comparing our outcomes to the state of the art.

Finally, we use a sliding window over the full domain of unseen IFC models to gather overall statistics on performance and confusion. In the view of the authors this is least useful, because it blindly aggregates over useful observations of the model as well as predication errors. We should also note that we do not have a curated ground truth as IFC models are so diverse. Also, on an individual voxel basis, a window frame is thin, but the surrounding wall is much thicker, so a wider decision boundary in favor of the window results in quite low IoU scores, particularly because a small radius of wall inclusion results in a large number of false window positives, regardless of the discussion whether the model is correct and that there maybe is a missing windowsill.

3.1 Results

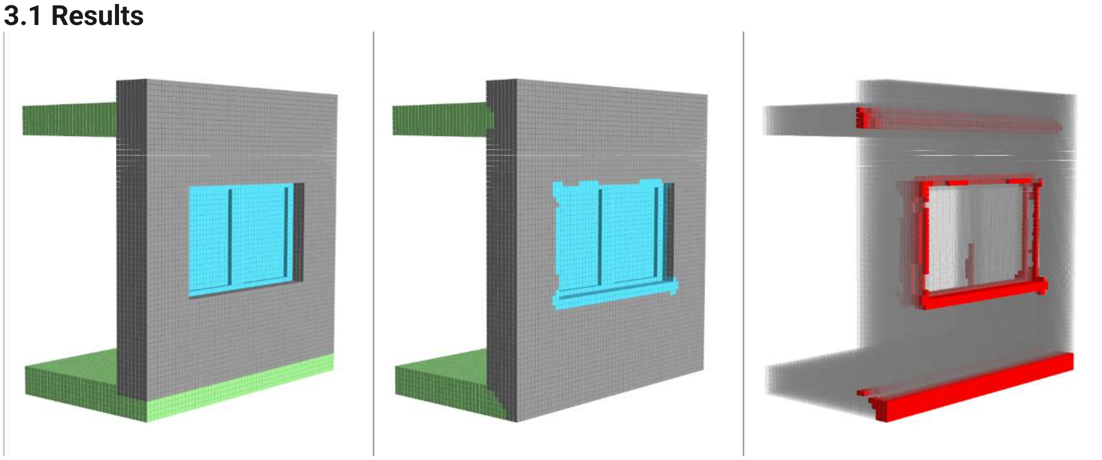

Figure 5. Window frame patch. Left: ground-truth. Right: model prediction. Left: error plot.

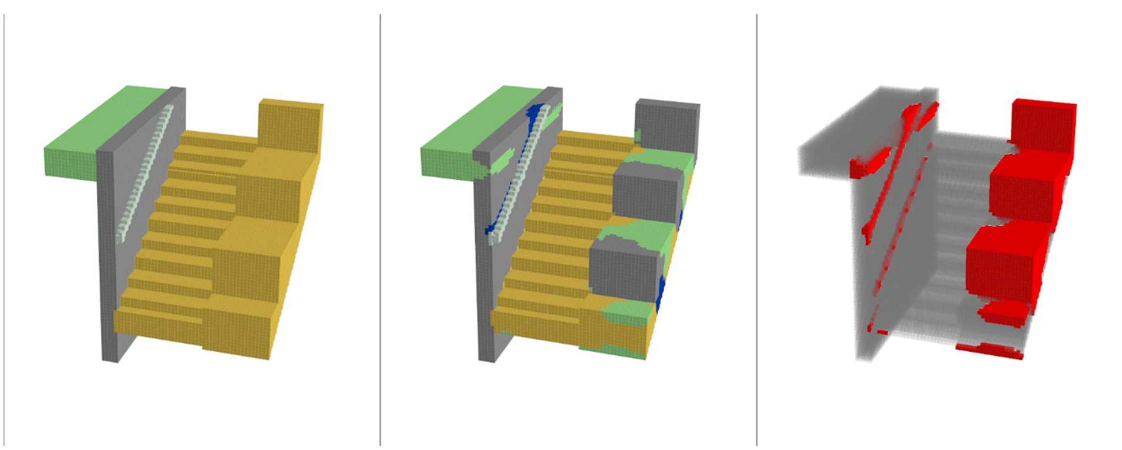

Figure 6. 'Podium staircase' patch. Left: ground-truth. Right: model prediction. Left: error plot.

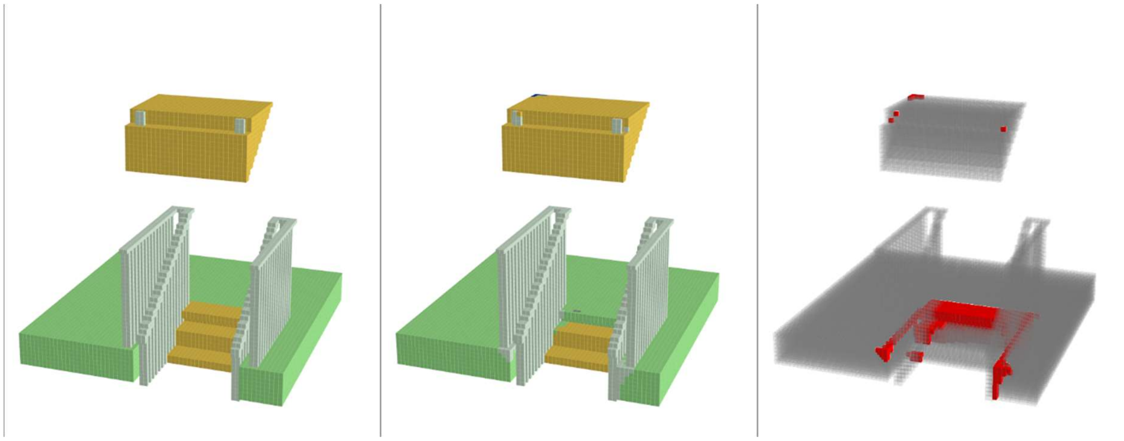

Figure 7. Stair/Slab connection patch. Left: ground-truth. Right: model prediction. Left: error plot.

Table 2. Element-based confusion matrix Molio

	Slab	Wall	Beam	Window	Door	Column	Railing	Stair	Furn
IfcSlab	37	0	0	0	0	0	0	0	0
IfcWall	0	98	0	0	0	0	0	0	0
IfcBeam	0	0	7	0	0	0	0	0	0
IfcWindow	0	2	0	106	11	0	0	0	0
IfcDoor	0	0	0	0	119	0	0	0	0
IfcColumn	0	11	0	0	0	3	0	0	0
IfcRailing	0	0	0	0	0	0	2066	0	0
IfcStairFlight	5	6	0	0	0	0	0	553	0
IfcFurnishingElement	3	25	1	0	0	0	0	0	9

Table 3. Element-based confusion matrix FZK Haus

	Slab	Wall	Beam	Window	Door	Column	Railing	Stair	Furn
IfcSlab	2	0	0	1	0		0	1	
IfcWall	0	13	0	0	0		0	0	
IfcBeam	0	3	0	1	0		0	0	
IfcWindow	0	0	0	11	0		0	0	
IfcDoor	0	0	0	0	5		0	0	
IfcColumn									
IfcRailing	0	0	0	0	0		2	0	
IfcStairFlight	0	0	0	0	0		0	0	
IfcFurnishingElement									

Table 4. Per-voxel accuracy confusion matrix Molio

	Slab	Wall	Beam	Window	Door	Column	Railing	Stair	Furn
Slab	0,6092	0,0156	0,0020	0,1598	0,0000	0,0000	0,0000	0,2134	0,0000
Wall	0,0109	0,9566	0,0050	0,0122	0,0024	0,0001	0,0001	0,0128	0,0000
Beam	0,0532	0,5652	0,0051	0,0102	0,0000	0,0000	0,0000	0,3664	0,0000
Window	0,0000	0,0458	0,0000	0,9541	0,0001	0,0000	0,0000	0,0000	0,0000
Door	0,0073	0,0710	0,0033	0,0624	0,8560	0,0000	0,0000	0,0000	0,0000
Column	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Railing	0,0127	0,0005	0,0000	0,0000	0,0000	0,0000	0,9868	0,0000	0,0000
Stair	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Furn	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

3.2 Discussion

From the confusion matrices (Tables 2, 3, 4), we observe that the network has learned a useful representation of building elements from different IFC models, except for some underrepresented or high intra-class variance classes. We do not report on Intersection over Union scores as they could be fairly low for those classes, to some extent inherent to inconsistent modeling. We observe for example for the windows that the model can predict frames to be larger, and also with the windowsill as part of the wall (Figure 5). In Figure 6, the podium staircaise is wrongly segmented as a wall. In Figure 7, we see that the model can confuse stairs landing as part of the slab. This is exemplified in element counts on Molio model. The high number of railings and stair flights comes from the fact that every bar and riser is modelled individually.
This demonstrates the ability of the approach to consider the context as it is - not surprisingly - very successful to assign the dominant class in these cases.

4 Conclusion and future work

We presented a method to learn from a voxelized representation of building elements belonging to open-source IFC files. We managed to obtain interesting insights about how the network learns and predicts classes when presented a voxel grid of zeros and ones.
We believe that learning a more granular decomposition of building elements (door knobs/drangers) with instance segmentation could be of interest to go further into the semantic/geometric enrichment of IFC models. This is related to the next step we would like to study, that is assessing the ideal voxel size and patch size.
Also we would find it relevant to consider something like material characteristics so that elements can be delineated by non-homogeneous materials. Currently it's impossible to learn something about coverings or an air gap. Finally, different tasks (e.g anonomaly detection on geometry) could be explored based on our trained model.

References

Choy, C.B., Xu, D., Gwak, J., Chen, K., et al. (2016) 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. [Online]. Available from: doi:10.48550/arXiv.1604.00449.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., et al. (2016) *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*. [Online]. Available from: doi:10.48550/arXiv.1606.06650.

Collins, F.C., Braun, A., Ringsquandl, M., Hall, D.M., et al. (2021) Assessing IFC classes with means of geometric deep learning on different graph encodings. In: Computing in Construction. [Online]. 2021 European Council on Computing in Construction. pp. 332–341. Available from: doi:10.35490/EC3.2021.168.

Emunds, C., Pauen, N., Richter, V., Frisch, J., et al. (2021) IFCNet: A Benchmark Dataset for IFC Entity Classification. [Online]. Available from: doi:10.48550/arXiv.2106.09712.

Emunds, C., Pauen, N., Richter, V., Frisch, J., et al. (2022) SpaRSE-BIM: Classification of IFC-based geometry via sparse convolutional neural networks. Advanced Engineering Informatics. [Online] 53101641. Available from: doi:10.1016/j.aei.2022.101641.

Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep Learning*. MIT Press.

Hanocka, R., Hertz, A., Fish, N., Giryes, R., et al. (2019) MeshCNN: a network with an edge. ACM Transactions on Graphics. [Online] 38 (4), 90:1-90:12. Available from: doi:10.1145/3306346.3322959.

He, K., Zhang, X., Ren, S. & Sun, J. (2015) *Deep Residual Learning for Image Recognition*. [Online]. Available from: doi:10.48550/arXiv.1512.03385.

Koo, B., Shin, B., & Krijnen, T. F. (2017). Employing outlier and novelty detection for checking the integrity of BIM to IFC entity associations. In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 34, p. 1). IAARC Publications.

Krijnen, T., El-Diraby, T., Konomi, T., & Attalla, A. (2021). Thermal analysis of IFC building models using voxelized geometries. In 38th International Conference of CIB W78 (pp. 437-446).

Krijnen, T., & Tamke, M. (2015). Assessing implicit knowledge in BIM models with machine learning. In Modelling Behaviour: Design Modelling Symposium 2015 (pp. 397-406). Springer International Publishing.

Lomio, F., Farinha, R., Laasonen, M. & Huttunen, H. (2018) Classification of Building Information Model (BIM) Structures with Deep Learning. In: 2018 7th European Workshop on Visual Information Processing (EUVIP). [Online]. November 2018 Tampere, IEEE. pp. 1–6. Available from: doi:10.1109/EUVIP.2018.8611701.

Luo, H., Gao, G., Huang, H., Ke, Z., et al. (2023) A Geometric-Relational Deep Learning Framework for BIM Object Classification. In: Leonid Karlinsky, Tomer Michaeli, & Ko Nishino (eds.). Computer Vision – ECCV 2022 Workshops. [Online]. 2023 Cham, Springer Nature Switzerland. pp. 349–365. Available from: doi:10.1007/978-3-031-25082-8_23.

Luttun, J., & Krijnen, T. (2021). An approach for data extraction, validation and correction using geometrical algorithms and model view definitions on building models. In Proceedings of the 18th International Conference on Computing in Civil and Building Engineering: ICCCBE 2020 (pp. 529-543). Springer International Publishing.

Noardo, F., Wu, T., Ohori, K. A., Krijnen, T., & Stoter, J. (2022). IFC models for semi-automating common planning checks for building permits. Automation in Construction, 134, 104097.

Qi, C.R., Su, H., Mo, K. & Guibas, L.J. (2017) PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. [Online]. Available from: doi:10.48550/arXiv.1612.00593.

Rogage, K. & Doukari, O. (2024) 3D object recognition using deep learning for automatically generating semantic BIM data. Automation in Construction. [Online] 162105366. Available from: doi:10.1016/j.autcon.2024.105366.

Ronneberger, O., Fischer, P. & Brox, T. (2015) *U-Net: Convolutional Networks for Biomedical Image Segmentation*. [Online]. Available from: doi:10.48550/arXiv.1505.04597.

Zhou, Y. & Tuzel, O. (2017) VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. [Online]. Available from: doi:10.48550/arXiv.1711.06396.