Neural Network for IFC class recognition

Wojciech Tecław, (wojciech.teclaw@sintef.no)
SINTEF Community, Trondheim, Norway & Department of Civil and Environmental Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Artur Tomczak, (artur.b.tomczak@ntnu.no)
Department of Civil and Environmental Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Marcin Łuczkowski, (marcin.luczkowski@ntnu.no)
Department of Structural Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Keywords: 3D object recognition, BIM, Classification, IFC, Machine Learning

Abstract
The Industry Foundation Classes (IFC) represent the cornerstone of Building Information Modelling (BIM), serving as a universal and standardized schema for information exchange in the construction and building industry. The extensive range of IFC classes poses a risk of element misclassification. Wrongly classified elements undermine data reliability, analytical processes, estimations and quantity take-offs. The study addresses the issue by introducing convolutional neural networks (CNN) trained to recognize individual elements based on their geometry. Comparing machine-identified entities with those labelled by a human allows the detection of potential mistakes. The model efficiency is substantiated through rigorous testing on a large dataset of categorized IFC elements extracted from hundreds of BIM models. The accuracy, performance and limitations are analysed and discussed. The development of this CNN-based approach marks a stride towards more efficient BIM data validation and, ultimately, data-driven design and construction processes.

1. Introduction
The Industry Foundation Classes (IFC) standard plays a key role in Building Information Modeling (BIM). It facilitates collaboration, curating, and exchanging data describing the built environment.
Central to the IFC are classes, called *entities*, that standardise how elements should be defined. Usually, the authoring software assigns a class based on category mapping, or user assigns it manually. The vast number of classes, coupled with a diverse array of software solutions in the market, often results in the wrong classification of elements (Holzer, 2011). In some cases, models solely consist of generic (proxy) elements, lacking any particular class, making them unusable in practice.
Recognizing these challenges, our research aims to investigate the feasibility of automatic IFC class recognition purely based on geometrical features. This way, the software could spot elements with a high likelihood of being misclassified, notifying the

user. The intention is not to replace manual categorization, as it is often the expert decision whether a certain geometry should represent one class or another.
To achieve our goals, we trained an artificial intelligence (AI) model based on a collection of thousands of properly labelled IFC objects. This training set is designed to represent a broad spectrum of IFC categories, enabling the AI to effectively learn and predict the classification of new, unlabelled geometrical objects.
AI-assisted BIM checking might help find misclassified objects, increasing the overall BIM quality. This could lead to significant improvements in efficiency and accuracy in building design, reducing the overhead associated with manual data verification and increasing the reliability of the BIM models used across various stages of construction.

2. Literature Review

The IFC addresses interoperability challenges in the architecture, engineering, and construction (AEC) industry. It provides a standardized data schema supporting data exchange across diverse software platforms, enhancing collaborative efforts and streamlining project workflows. However, the complexity of IFC schema and the manual effort required in classifying BIM elements often leads to inconsistencies and errors, undermining the potential of BIM systems to achieve accurate data management and application (Sobhkhiz and El-Diraby, 2023).
Recent advancements in machine learning introduced novel approaches to automating the classification of IFC entities. The study from Krijnen and Tamke demonstrates the potential of AI for assessing implicit knowledge in BIM data (Krijnen and Tamke, 2015). It employs supervised and unsupervised machine learning methods to search for misclassified elements in IFC models. The research simplified the geometrical features into three aspects: surface area, volume, and the radius of gyration (gyradius). Another paper emphasizes the potential of deep learning in automatically generating semantic BIM data (Rogage and Doukari, 2024). The research highlights the application of convolutional deep belief networks in recognizing 3D objects, thereby automating the enrichment of semantic data, improving interoperability, and reducing manual data entry errors. Koo et al. delve into geometric deep learning models for , which classifying infrastructure BIM elements based on their geometric features, such as Multi-View Convolutional Neural Networks (MVCNN) (Koo et al., 2021) and PointNet (originating from Qi et al., 2016). These models have shown substantial promise, with both MVCNN and PointNet demonstrating high performance in capturing subtle geometric differences essential for accurate IFC mapping. This study underscores the importance of sophisticated model architectures in handling the complex geometries typical of BIM elements and ensuring their correct classification based on IFC standard.
Emunds et al. (Emunds et al., 2022) introduce SpaRSE-BIM, a model that employs sparse convolutional neural networks to classify IFC-based geometry while offering semantic enrichment effectively. This approach not only aids in maintaining data consistency but also optimizes processing speeds, making it viable for real-time applications in complex BIM environments.
Despite these technological advances, several challenges persist. The work of Sobhkhiz and El-Diraby on the dynamic integration of unstructured data with BIM through machine learning and concept networks highlights the ongoing need to handle unstructured data efficiently (Sobhkhiz and El-Diraby, 2023). Their no-model approach using graph theory and NLP to classify documents into IFC classes represents a pivotal shift towards more adaptable and robust BIM data management systems.
Moreover, Wu and Zhang's (Wu and Zhang, 2018) exploration of geometric theorems to automate BIM object classification and the subsequent use of support vector machines

by Koo et al. (Koo et al., 2019) for the same purpose exemplify the breadth of computational strategies being employed to refine the process of IFC categorization. These methodologies illustrate the diverse approaches researchers take to overcome the intrinsic challenges posed by construction projects' varied and complex nature.
The integration of AI and BIM, especially for IFC classification, is expected to evolve further, with continuous improvements in model accuracy, computational efficiency, and adaptability to new types of BIM data.

3. Methodology

3.1. Research approach

This proof-of-concept study focuses on developing an algorithm that recognizes building elements by analyzing their geometric features. The study began with a literature review and examination of the problem domain, essential for establishing the algorithm's requirements and constraints. This section details the data-gathering process, the model architecture and training of neural networks, and the integration of these components into a functional application.

3.2. Algorithm design

Based on the identified requirements and engineering practices we developed a conceptual framework for category validation. The workflow begins with extracting a single element from the IFC model, as illustrated in Figure 1. Then, the IFC element's geometry is converted to a point cloud with a fixed number of points.
Next, we pass this set of points to the neural network which predicts the probability of the element's IFC category based on its geometry. If the category with the highest probability matches the element's assigned category, the algorithm proceeds to the next element. A review request is sent to the user if there is a discrepancy between the predicted and assigned categories.

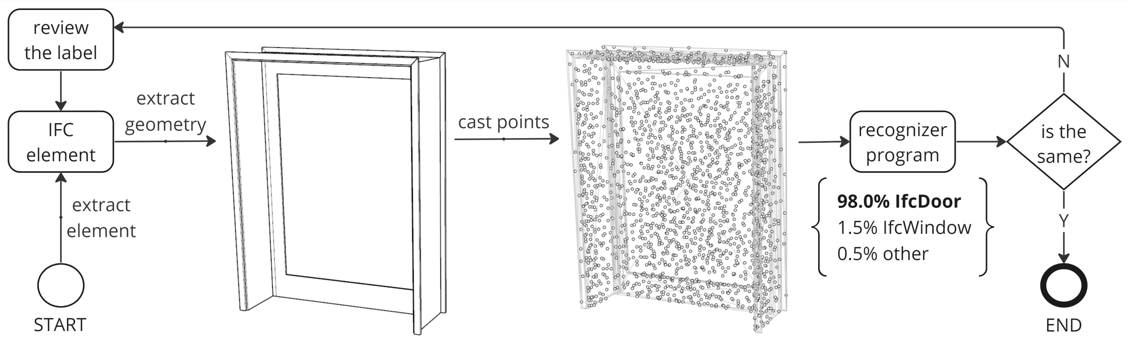

Figure 1: The workflow

The proposed framework relies significantly on the AI model, with its accuracy and reliability hinging on the diversity of datasets used during training. However, it is important to acknowledge that the model can never guarantee 100% accuracy in its predictions. Consequently, the user's role remains essential in the workflow, serving as the final decision-maker who approves or rejects the suggestions provided by the system.

3.3. Data Collection

To train the model, we need vast input data containing elements with geometry and adequate class label - IFC entity. We collected 245 IFC models from various public repositories, primarily buildingSMART International, 2021; OSArch Community, 2021; The University of Auckland, 2021. From those, using a bespoke script, we extracted 884'008 instances of elements that contained unique geometry. The set has representatives of 84 IFC classes, with the predominant being structural framing (IfcBeam, IfcColumn, and IfcMember), reinforcement (IfcReinforcingBar), pipes and ducts (IfcFlowSegment). Figure 2 presents the count of elements per each class. Each instance was then saved into an OBJ file for easier processing, together with metadata including the label, source file name, location in the model, identification, and matrix of rotation. The resultant dataset was published as supplementary material on Zenodo repository (Tomczak et al., 2024).

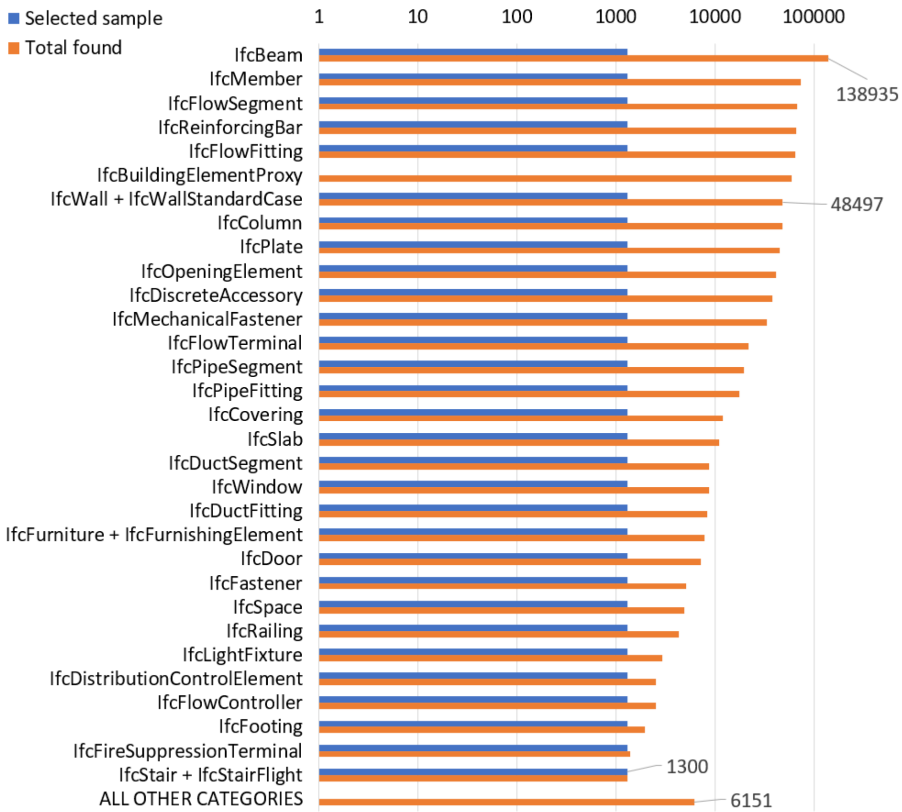

Figure 2: Representation of each IFC class in the dataset (logarithmic scale)

From the obtained dataset, we picked a subset of categories with a substantial representation of more than 1200 samples. Additionally, to ensure consistency and clarity of the analysis, we removed the undefined proxies (IfcBuildingElementProxy) and consolidated similar or hereditary categories (IfcStair and IfcStairFlight, IfcWall and IfcWallStandardCase, IfcFurniture, and IfcFurnishingElement). This process yielded a comprehensive and structured dataset of 23 suitable for the rigorous investigation of geometric patterns. For efficiency of the processing, we then extracted the metadata from OBJ to separate JSON files.

3.4. Data preprocessing

All the elements in the dataset were randomly distributed in their corresponding categories into three datasets: training, validation, and testing. 80% of the elements were used for training the neural network, 10% was dedicated to validating the progress of the neural network training, and the final 10% to test the post-training benchmark. In the next step, a mesh representation (OBJ) was converted to a point cloud (XYZ), by casting 2048 points into each element's surface (this number matches the PointNet model's training specifications) using the Trimesh Python library (Dawson-Haggerty, 2022). This way, all the objects, regardless of their shape complexity, are expressed in an equal number of points, suitable for AI applications. To reduce the influence of the scale factor, we normalised the dataset to a range from 0 to 1 unit while maintaining the proportion between dimensions. Such preparation is then used as input for the CNN model.

3.5. AI model architecture

Based on the literature review, the PointNet architecture of CNN was selected (Qi et al., 2016). It employs a series of transformation networks that help align the points into a canonical or standard orientation before feature extraction occurs, improving the model's robustness and accuracy. This is crucial for tasks like classification and segmentation, where the spatial arrangement of points can significantly impact performance. Furthermore, PointNet can handle varying sizes of point sets by independently using a shared MLP (multi-layer perceptron) on each point, enabling it to learn local and global structures effectively. To train the model, 512 epochs (steps) were conducted with a constant learning rate (lr) equal to 0.001.
The code was written in Python mainly using the PyTorch library (The Linux Foundation, 2024). The model was trained using Google Colab, working on the Ubuntu 22.04 system with 83.5 GB of RAM. Additionally, to accelerate the training the graphics processing unit (GPU) Nvidia A100 (40Gb) was used.

4. Results

The model took five and a half hours to train, but thanks to this, the actual class recognition on pretrained model is a matter of seconds.
The result of the AI model training is the model with the highest accuracy of 61.7%. The distribution of the correct and incorrect predictions based on the IFC class is shown in the figure 3.

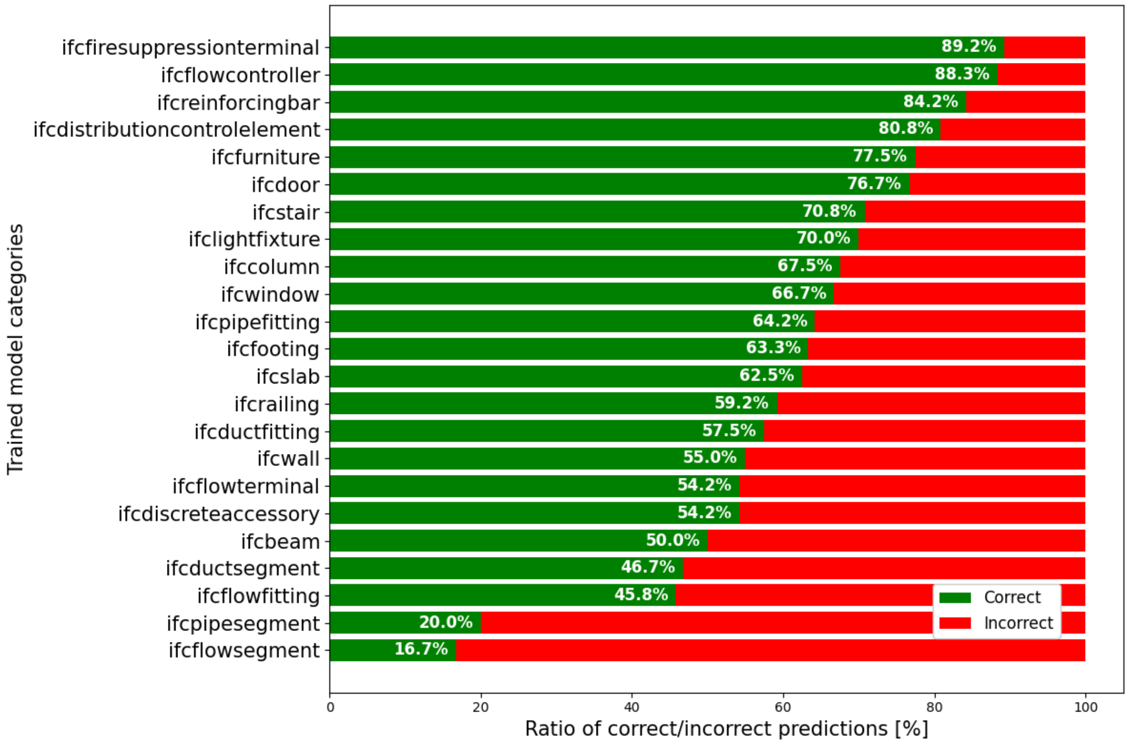

Figure 3: Training result - correct and incorrect predictions per IFC category

The classes characterized by the lowest accuracy rates include fluid transport: IfcFlowSegment, IfcPipeSegment, and IfcDuctSegment, which demonstrated accuracies of 16.7%,

20.0%, and 46.7%, respectively. A related second category features fittings, which exhibit a range of accuracies between 45.8% and 57.5%.
Conversely, the categories featuring higher accuracies are IfcReinforcementBar, IfcFireSuppressionTerminal, IfcFlowController, and IfcDistributionControlElement, with accuracies of 84.2%, 89.2%, 88.3%, and 80.8% respectively. These results underscore a significant variation in model performance across different classes, reflecting the potential for targeted improvements in class-specific training approaches.

5. Discussion

The training resulted in an average model accuracy of approximately 61,7%. This accuracy is significantly lower than that documented in the original PointNet concept (Qi et al., 2016), necessitating caution when interpreting the results. Still, it can warn in most cases of element misclassification, increasing overall efficiency. It is important to remember that in BIM, the geometry is only a simplified representation that conveys just a part of the information. Other important aspects, such as classification and properties, can determine the category.
Despite utilizing a balanced, and diverse dataset, several factors may have caused low accuracy observed in our AI model's performance. A primary concern is the lack of distinct differences between categories of geometry, particularly evident in the low accuracy rates for IfcFlowSegment and its derivative categories, IfcDuctSegment and IfcPipeSegment. The round shapes common to both ducts and pipes create significant confusion for the model as it struggles to differentiate between these similar geometries.
To address the required classification precision, it's evident from our observations that distinguishing between elements such as a pipe and a duct may not be as critical as identifying the overall category of a flow segment. A practical approach to enhancing model performance would be to consolidate the categories of IfcPipeSegment and IfcDuctSegment into a single, broader category labelled IfcFlowSegment. Additionally, considering the similar challenges with IfcFlowFitting, merging this category could simplify the classification task and potentially elevate the model's accuracy.
Moreover, architectural models' complexity often introduces additional challenges. Variations in geometric shapes, sizes, and contextual positioning can further complicate the model's ability to classify elements accurately. Enhancing the training dataset with more specific features or employing advanced augmentation techniques could potentially address some of these challenges and lead to better model performance. Another alternative is the usage of MVCNN, which might provide better accuracy by leveraging multiple views of an object to capture a more comprehensive range of geometric details, potentially improving the classification of complex architectural elements. Nevertheless, current approaches generally overlook the potential benefit of integrating the spatial context of elements within the construction model into the classification process. While geometry plays a crucial role, the placement of an element in relation to others can be equally important for accurate classification. Addressing this aspect could lead to further improvements in model performance, making it more robust in handling the intricacies of AEC designs.
Alternatively, this problem might be addressed by splitting the classes with a more domain-focused approach, limiting the number of classes. However, such an approach would require additional input from users to classify model characteristics into domains such as Architecture, Structure, Ventilation, or Infrastructure. This user-driven classification could help the model by providing contextual information that it might otherwise struggle to infer from geometric data alone.
Additionally, manually curating the dataset to exclude unrecognizable elements could

further enhance the model's accuracy. With further refinement and user collaboration, the model can become a more reliable tool in reducing manual work and enhancing the efficiency of architectural and structural design processes.

6. Conclusion and Future Work

This study successfully applied a machine learning model for automatic IFC class recognition and proposed the framework for automated validation of IFC categories in a BIM file. The model's application can significantly reduce manual classification efforts in BIM processes, enhancing data reliability and efficiency.

This research has identified a number of promising avenues for future development. To expand the scope of the study, it would be beneficial to enhance the dataset with real-life IFC models from disciplines that have not yet been explored, such as roads or rails. Improving the model could include refining the accuracy of class prediction and incorporating additional factors such as normalization rates and actual rotation. Furthermore, adopting a multi-agent approach could improve performance by leveraging various methodologies.

Integrating contextual information about the surrounding elements could lead to more precise classifications. Investigating alternative representations, such as voxel or raster images, and extending the research to include more IFC classes or other classification systems like CCI or Uniclass could significantly extend the tool's applicability. Finally, developing a user-friendly interface would likely encourage its adoption among practitioners, thereby increasing its practical utility.

Acknowledgements

This study was partially funded by the HumanTech project (https://humantech-horizon.eu/), funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 101058236.

References

buildingSMART International. (2021). Sample test files. https://github.com/buildingSMART/Sample-Test-Files

Dawson-Haggerty, M. (2022). Trimesh 4.4.0. https://trimesh.org/

Emunds, C., Pauen, N., Richter, V., Frisch, J., & van Treeck, C. (2022). Sparse-bim: Classification of ifc-based geometry via sparse convolutional neural networks. *Advanced Engineering Informatics*, *53*, 101641. https://doi.org/https://doi.org/10.1016/j.aei.2022.101641

Holzer, D. (2011). Bim's seven deadly sins. *International Journal of Architectural Computing*, *9*(4), 463–480. https://doi.org/10.1260/1478-0771.9.4.463

Koo, B., Jung, R., Yu, Y., & Kim, I. (2021). A geometric deep learning approach for checking element-to-entity mappings in infrastructure building information models [Cited by: 25; All Open Access, Gold Open Access]. *Journal of Computational Design and Engineering*, *8*(1), 239–250. https://doi.org/10.1093/jcde/qwaa075

Koo, B., La, S., Cho, N.-W., & Yu, Y. (2019). Using support vector machines to classify building elements for checking the semantic integrity of building information models. *Automation in Construction*, *98*, 183–194. https://doi.org/https://doi.org/10.1016/j.autcon.2018.11.015

Krijnen, T., & Tamke, M. (2015). Assessing implicit knowledge in bim models with machine learning. *Modelling Behaviour*, 397–406. https://doi.org/10.1007/978-3-319-24208-8_33

OSArch Community. (2021). Aeco workflow examples. https://wiki.osarch.org/index.php?title=AECO_Workflow_Examples

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2016). Pointnet: Deep learning on point sets for 3d classification and segmentation. http://arxiv.org/pdf/1612.00593v2

Rogage, K., & Doukari, O. (2024). 3d object recognition using deep learning for automatically generating semantic bim data. *Automation in Construction*, *162*, 105366. https://doi.org/https://doi.org/10.1016/j.autcon.2024.105366

Sobhkhiz, S., & El-Diraby, T. (2023). Dynamic integration of unstructured data with bim using a no-model approach based on machine learning and concept networks. *Automation in Construction*, *150*, 104859. https://doi.org/https://doi.org/10.1016/j.autcon.2023.104859

The Linux Foundation. (2024). Pytorch 2.2.1. https://pytorch.org/

The University of Auckland. (2021). Openifc model repository. https://openifcmodel.cs.auckland.ac.nz/

Tomczak, A., Tecław, W., & Łuczkowski, M. (2024). The geometry of 884k construction products extracted from ifc/bim models with ifc labels. https://doi.org/https://doi.org/10.5281/zenodo.10730758

Wu, J., & Zhang, J. Automated bim object classification to support bim interoperability [Cited by: 13]. In: *2018-April*. Cited by: 13. 2018, 706–715. https://doi.org/10.1061/9780784481301.070