Integrated Framework for IFC Model Generation and Structural Layout from Architectural Sketch Plans

Shima Afshar
M.Sc. in Architecture, Department of Architecture and Urban Planning, Shahid Beheshti University, Tehran, Iran(shi.afshar@mail.sbu.ac.ir)

Mohammad Tahsildoust*
Associate Professor, Department of Architecture and Urban Planning, Shahid Beheshti University, Tehran, Iran (m_tahsildoust@sbu.ac.ir)

ORIGINAL RESEARCH ARTICLE

Afshar, Shima, and Mohammad Tahsildoust. "Integrated Framework for IFC Model Generation and Structural Layout from Architectural Sketch Plans". *Soffeh* 36, no. 2 (2026): 83–108. DOI: http://doi.org/10.48308/soffeh.2026.242295.1468

Received: October 30, 2025
Revised: December 02, 2025
Accepted: December 14, 2025
(Pages: 83–108)

Abstract

Background and Objectives: Recent advances in Artificial Intelligence (AI) and Building Information Modelling (BIM) have enabled closer integration between data-driven automation and architectural design processes. Nevertheless, current AI-based BIM modelling approaches—particularly in the plan recognition stage—remain heavily reliant on large labelled datasets, making data preparation time-consuming and labour-intensive. This dependency limits rapid iteration in early-stage architectural design, where speed and responsiveness are essential. To address these challenges, this study proposes an integrated automation framework that leverages hybrid computational methods to accelerate the translation of sketch-based floor plans into structured BIM models.

Materials and Methods: The proposed workflow comprises three main components. First, 2D raster floor plans are converted into vector data using a hybrid approach that combines deep learning-based opening detection (YOLOv11) with rule-based wall extraction. Second, the framework automatically recommends initial column layouts optimised according to spatial and functional design criteria. Third, it programmatically generates Industry Foundation Classes (IFC)-

Keywords:
Building Information Modelling (BIM),
Deep Learning,
Industry Foundation Classes (IFC),
Architectural Plan Interpretation,
Structural Layout Optimisation.

*. Corresponding Author Email Address: m_tahsildoust@sbu.ac.ir

http://doi.org/10.48308/soffeh.2026.242295.1468

Research Questions:

1. .How can hybrid computational methods integrate deep learning and rule-based techniques to efficiently convert sketch-based floor plans into structured BIM models?

2.In what ways can reliance on large labelled datasets be minimised during the plan recognition stage of AI-based BIM modelling?

3.How effectively can the proposed framework automatically recommend and optimise initial structural column layouts according to spatial and functional design criteria?

4.What is the accuracy and efficiency of the integrated framework in object detection, wall recognition, and IFC-compatible 3D model generation?

5.How can this integrated automation framework reduce manual effort and accelerate the early-stage architectural design process by bridging manual plan interpretation and BIM workflows?

compliant 3D models. Geometric information extracted from sketch plans produces an IFC file encompassing walls, floors, doors, windows, and preliminary structural column layouts, closely reflecting the original design intent.

Results and Conclusion: Experimental evaluation demonstrates robust performance across all stages. The object detection model achieved a mAP50:95 of 81%, with an inference time of 0.0041 seconds per image. Wall recognition reached 100% accuracy, while the column layout algorithm attained 95.83% alignment with executed plans. IFC generation was completed in under one minute, ensuring efficiency and practicality. These results underscore the framework's capacity to substantially reduce manual effort, minimise reliance on large labelled datasets, and expedite the transition from conceptual sketches to BIM. By integrating hybrid computational techniques, the proposed approach offers a reliable, automated, and scalable solution for early-stage architectural design, bridging the gap between manual plan interpretation and BIM workflows, and providing a foundation for future research in intelligent architectural modelling.

1. Introduction

In recent decades, the architecture, engineering, and construction (AEC) industry has increasingly adopted digital technologies for generating three-dimensional (3D) models, ranging from NURBS-based modelling to computer-aided design (CAD) systems, and, more recently, Building Information Modelling (BIM). Among these, object-oriented BIM platforms have become particularly influential, offering the ability to exchange detailed information among stakeholders, preserve data consistency across the project lifecycle, and minimise errors caused by rework or interdisciplinary clashes. This shift is reflected in the global market, where BIM was valued at approximately USD 8.1 billion in 2022 and is projected to reach USD 14.6 billion by 2028, representing a compound annual growth rate (CAGR) of 11.3%.[1]

Architectural design is an iterative, multi-phase process

1. Prasad Padwal, 'Building Information Modeling (BIM) Market Size & Outlook,' *Leadership and Management in Engineering*, (2024), https://www.researchgate.net/publication/377969878_Building_Information_Modeling_BIM_Market_Size_Outlook.

2. Farzad Jalaei et al., 'Integrating Decision Support System (DSS) and Building Information Modelling (BIM) to Optimize the Selection of Sustainable Building Components,' *Journal of Information Technology in Construction* 20 (2015): 399–420, https://www.itcon.org/paper/2015/25.

3. Farzad Jalaei and Ahmad Jrade, 'Integrating Building Information Modelling (BIM) and Energy Analysis Tools with Green Building Certification System to Conceptually Design Sustainable Buildings', *Journal of Information Technology in Construction* 19 (2014): 500, https://doi. org/10.1061/9780784413517.015.

in which early-stage decisions critically affect downstream performance, cost, and constructability outcomes[2]. Inadequate consideration during these stages often leads to design inefficiencies and increased project costs. BIM provides an integrated framework to support informed decision-making by enabling multidisciplinary collaboration and interoperable data exchange, thus minimising redundant modelling efforts. Integrating real-time structural layout recommendation systems within BIM environments can further enhance early design workflows by allowing architects to evaluate alternative configurations based on structural performance and constructability criteria, fostering more holistic and data-driven design outcomes.[3]

Despite these advantages, BIM adoption in the early design stages remains limited,primarily due to the high costs and time required to create detailed models, the need for specialised expertise, and the adjustments required in conventional workflows. Recent research has attempted to address these barriers through automation, particularly by leveraging artificial intelligence (AI) and intelligent application programming interfaces (APIs) to generate initial building components with minimal manual intervention. A central focus of these efforts is the automated creation of Industry Foundation Classes (IFC) models – anopen, platform-independent standard that ensures bothgeometric and semantic data exchange across BIM platforms. Automation not only reduces human error and accelerates updates but also ensures that models remain consistent and synchronised across stakeholders. Prior studieshighlight that IFC automation improves data-to-model conversion, increasesproductivity, and enhances model quality. In theearly design stages, timely 3D models enable spatial analysis, clash detection, and performance simulations. However, manual conversion from two-dimensional (2D) drawings toBIM remains aslow and error-prone process, underscoring the need for automation to support rapid iterations, greater accuracy, and efficiency in both new construction and renovation projects.Previous studies have explored this challenge from multiple perspectives. Some have focused on automating the transformation of CAD files into 3D BIM models[4], using rule-based geometric algorithms that apply predefined geometric rules. Others have concentrated on raster- to- BIM pipeline, which combine deep learning, rule-based, or hybrid methods.[5] Liang et al.presented a fully automated BIM dataset generation framework that integrates a modified ResNet-34 and U-Net to produce paired synthetic 2D floor plans and 3D BIMs, introducing the novel ResBIM dataset for standardised 2D-to-BIM evaluation.[6] Additionally,some recent advances increasingly leverage deep learning to enhance point cloud segmentation and automate BIM creation. For example, Wang et al.[7] utilised a hybrid approach that combines RANSAC-based multi-plane segmentation with a lightweight convolutional encoder–decoder network to automate BIM reconstruction from drone-acquired point clouds. In parallel, Erişen et al.[8] introduced a Vision Transformer–based multi-task deep learning pipeline that jointly performs single-image depth estimation and semantic segmentation to reconstruct BIM-compatible, semantically enriched 3D indoor models directly from RGB imagery, while also enabling domain adapta-

4. Joie Lim et al., 'Automated Generation of BIM Models from 2D CAD Drawings,' paper presented at Computer Aided Architectural Design Research in Asia, (2018), https://doi.org/10.52842/conf.caadria.2018.2.061.

5. Aleixo Cambeiro Barreiro et al., 'Automatic Reconstruction of Semantic 3D Models from 2D Floor Plans, ' *18th International Conference on Machine Vision and Applications (MVA)*, (2023), https://doi.org/10.23919/MVA57639.2023.10215746; Sungsoo Park and Hyeoncheol Kim, '3DPlanNet: Generating 3D Models from 2D Floor Plan Images Using Ensemble Methods,' *Electronics* 10, no. 22 (2021), https://doi.org/10.3390/electronics10222729%3Furlappend=%3Futm_source=researchgate; Yunfan Zhao et al., 'Reconstructing BIM from 2D Structural Drawings for Existing Buildings,' *Automation in Construction* 128, no. 103750 (2021), https://doi.org/10.1016/j.autcon.2021.103750.

tion to recognise previously unrepresented object classes in 3D point-cloud space.

Another key challenge during early-stages design is thegeneration of structural layout, particularly when architectural forms are still evolving, and complete structural information is not yet available. Early decisions regarding column placement and structural grid configuration significantly influence constructability, spatial performance, and structural integrity. Traditionally, these decisions rely heavily on engineers' expertise and iterative adjustments, which can be time-consuming and inconsistent. Automated structural layout generation has the potential to accelerate decision-making, reducehuman error, and enable rapid exploration of alternatives.

Prior research on structural layout automation has applied optimisation algorithms[9] and generative deep learning models,[10] to recommend structural layouts, focusing primarily on parameters such as cost, structural member sizing,[11] code compliance,[12] environmental constraints[13] and column coordinates.[14] However, in the early design stages, functional alignment between the column grid and the architectural layout is critical for ensuring usable and efficient spaces. Accordingly, this study prioritises functional criteria in generating initial column layouts,thereby providing a design solution that supports both architectural intent and downstream structural analysis.

By integrating architectural modelling automation with structural layout generation,this research establishes a unified and intelligent workflow for conceptual building design. The resulting IFC models are compatible with widely used platforms, including Autodesk Revit, Graphisoft ArchiCAD, Navisworks, and BlenderBIM for modelling and coordination; Autodesk Robot Structural Analysis, Tekla Structures, and ETABS for structural analysis; and Simergy, IES-VE, and RIUSKA for energy simulations.[15] Automating IFC model creation at the early design stages not only accelerates multidisciplinary evaluations but also strengthens the foundation for subsequent architectural, structural, and performance-based analyses.

In response to these challenges and opportunities, this studyproposes an integrated framework, as illustrated in Figure 1,which:

- Combines a hybrid deep learning, YOLO (You Only Look Once), and rule-based approach to interpret raster floor plan images and convert them into vectorised data.
- Employs a brute-force search algorithm to recommend an optimised initial column arrangement according to predefined performance criteria, derived from the vectorised plan.
- Automatically generates a 3D IFC model from structured JSON data generated in previous stages, ensuring compatibility with BIM platforms and analysis tools.

The deep learning and IFC automation modules were implemented in Google Colab, while

6. Xing Liang et al., 'Fully Automated Synthetic BIM Dataset Generation Using a Deep Learning-Based Framework,' *Automation in Construction* 181, no. 106584 (2025), https://doi.org/10.1016/j.autcon.2025.106584.

7. Dejiang Wang et al., 'Existing Buildings Recognition and BIM Generation Based on Multi-Plane Segmentation and Deep Learning,' *Buildings* 15, no. 691 (2025), https://doi.org/10.3390/buildings15050691.

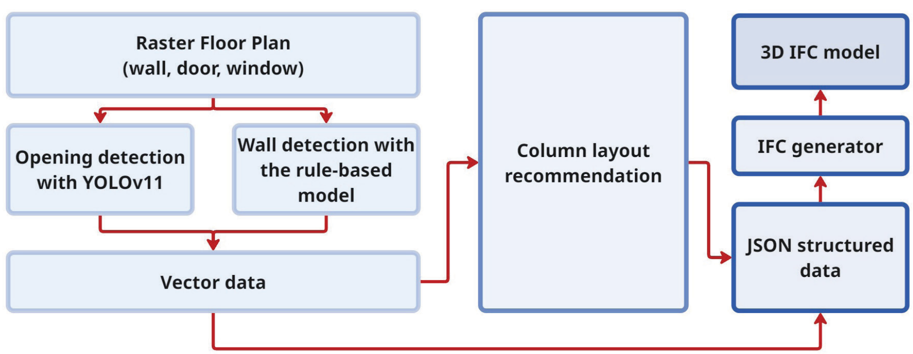


Figure 1. Overall workflow of the proposed framework for converting a raster floor plan into a 3D IFC model with a recommended column layout.

the column layout algorithm was developed in Grasshopper for Rhinoceros. Evaluations confirmedboth accuracy and practicality:the object detection module achieved an mAP50 of 81.85% at a 0.0041 second per image (SPI) inference time, the structural layout recommendation reached 95.83% alignment with executed projects, and the IFC generator produced complete models significantly faster than expert manual modelling. The resulting IFC modelsincluded both geometric and semantic data for walls, doors, windows, columns, and slabs, ensuring usability for further development in BIM platforms such as Autodesk Revit.

2. Literature Review

This section reviews prior research relevant to automated BIM workflows, focusing on three core components of the present study: (1) automated extraction of architectural elements from 2D raster floor plans, (2) structural layout recommendation in the early design stages, and (3) automated generation of IFC-compliant models. The review synthesises journal articles, conference proceedings, and review studies, selected through targeted search strategies designed to match the capabilities of different academic databases.

2. 1. Automated Data Extraction from 2D Raster Floor Plans

As illustrated in Figure 2, raster images of 2D floor plans are the most frequently used input for automated 3D model reconstruction, representing 49% of studies by 2024 due to their wide availability, ease of access, and strong compatibility with computer vision techniques[16]. In line with this trend, the present research also adopts raster floor plans as its primary input for automated BIM conversion.

Two principal approaches dominate the extraction of architectural information from raster images: rule-based methods and data-driven approaches. Rule-based strategies rely on predefined heuristics, geometric recognition, and vectorisation techniques to identify and classify architectural elements (e.g., walls, doors, and windows). Data-driven approaches, by contrast, employ deep learning models trained on labelled datasets to detect and classify objects. Hybrid pipelines often combine these approaches, using rule-based post-processing to refine the outputs of object detection networks.

Early developments in this field were dominated by rule-based techniques. For example,Gimenez et al.[17] identified walls as pairs of parallel lines, assigned openings as voids within those walls, and defined spaces as closed loops. Similarly,Q. Lu et al.[18] extracted structural information from CAD drawings by detecting columns via the Circle Hough Transform and using Optical Character Recognition (OCR) to classify textual annotations.

Recent advances, however, are increasingly data-driven. Faster R-CNN has been widely adopted, as demonstrated by Zhao et al.[19] for beam and column detection, and by Barreiro et al.[20] for door and window detection combined with semantic wall segmentation via Feature Pyramid Networks (FPN) with ResNet.Urbieta et al.[21] applied Mask R-CNN with OCR to detect and extract semantic attributes from both architectural and structural plans.

Some studies integrate multiple deep learning architectures within a single pipeline to le-

8. Serdar Erişen et al., 'Single Image to Semantic BIM: Domain-Adapted 3D Reconstruction and Annotations via Multi-Task Deep Learning,' *Remote Sensing* 17, no. 2910 (2025), https://doi.org/10.3390/rs17162910.

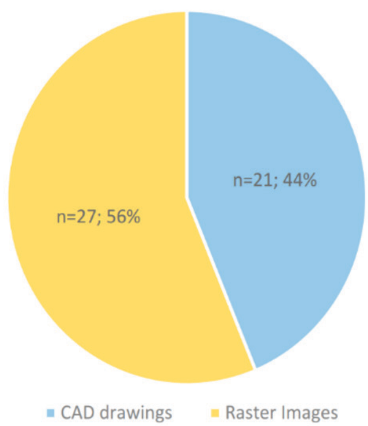
9. Alper Kanyilmaz et al., 'A Genetic Algorithm Tool for Conceptual Structural Design with Cost and Embodied Carbon Optimization,' *Engineering Applications of Artificial Intelligence* 112, no. 104711 (2022), https://doi.org/10.1016/j.engappai.2022.104711; Xianchuan Meng et al., 'Optimizing Support Locations in the Roof–Column Structural System,' *Applied Sciences* 11, no. 6 (2021), https://doi.org/10.3390/app11062775; Wenchen Shan et al., 'Integrated Method for Intelligent Structural Design of Steel Frames Based on Optimization and Machine Learning Algorithm,' *Engineering Structures* 284, no. 115980 (2023), https://doi.org/10.1016/j.engstruct.2023.115980; Mohamed Sherif et al., 'Automated BIM-Based

verage the strengths of different models. For instance,Kippers et al.[22] combines U-Net and Fast-SCNN for wall segmentation with Faster R-CNN, CentreNet, SSD MobileNet, and RetinaNet for door and window detection, and incorporates LSTM-based OCR for reading space labels. Likewise,Jang et al.[23] employs CNN models such as DarkNet53, Deeplabv3, and PSPNet for wall and door segmentation, applies ensemble modelling to improve Intersection over Union (IoU) scores, and uses centreline and corner detection to generate topological representations. Inthe study by Liu et al.,[24] raster floor plans were converted into CAD-compatible vector models by combining a ResNet-152 convolutional network with integer programming to enforce geometric and semantic constraints. In Z. Lu et al.,[25] a shared neural network detected both graphical elements and textual labels in rural residential floor plans, followed by Mixed-Integer Quadratic Programming (MIQP)-based segmentation of rooms and extraction of topological connectivity graphs.Finally, Zhao et al.[26] uses a YOLO-based framework to detect structural components from 2D plans using a model pre-trained on ImageNet and fine-tuned with 360 annotated images, demonstrating robust detection of structural components with post-processing via Non-Maximum Suppression for refinement.

Hybrid approaches have also been proposed to combine the interpretability of rule-based logic with the adaptability of data-driven detection. For instance,Leon-Garza et al.[27] applied fuzzy logic to classify pixels as wall or background using contextual information, refined by experts refining the rules.Doukari and Greenwood[28] employed a knowledge-base-driven approach, applying image processing techniques to aerial building imagery to infer semantic attributes such as building type and material usage. The 3DPlanNetEnsemble framework by

Structural Design and Cost Optimization Model for Reinforced Concrete Buildings,'*Scientific Reports* 12, no. 1 (2022), https://doi.org/10.1038/s41598-022-26146-6.

10. Bochao Fu et al., 'Dual Generative Adversarial Networks for Automated Component Layout,'*Automation in Construction* 146, no. 104661 (2023), https://doi.org/10.1016/j.autcon.2022.104661; Chong Zhang et al., 'End-to-End Generation of Structural Topology for Complex Architectural Layouts with Graph Neural Networks,'*Computer-Aided Civil and Infrastructure Engineering* 39 (2022): 756–75, https://doi.org/10.1111/mice.13098%3Furlappend=%3Futm_source=researchgate.



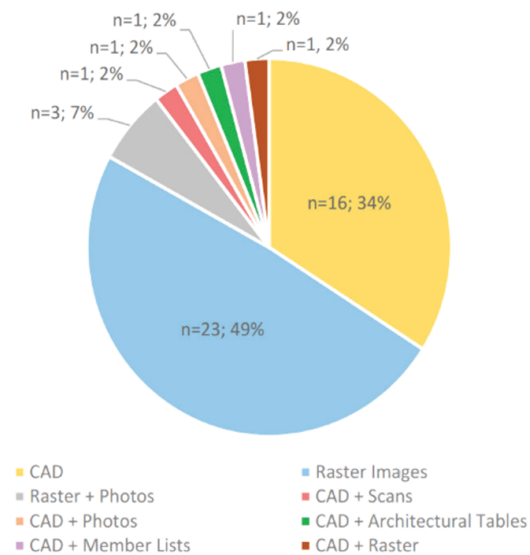


Figure 2. Proportion of different input data sources employed in previous studies on automated 3D model generation (Feist et al., 2024).

Park and Kim[29] merged logistic regression for wall detection and TensorFlow-based object detection for openings and spaces with geometric rules, including alignment checks, duplicate line removal, and merging of disconnected areas.

2. 2. Structural Layout Recommendation

In early design, structural layout generation is increasingly addressed through computational strategies that balance constructability, cost, and performance. Recent studies addressing structural layout recommendation in the early design stages adopt a spectrum of computational strategies, each leveraging distinct input types and optimisation objectives. These methods can be categorised as evolutionary algorithms, generative deep learning modules, hybrid combinations of machine learning and optimisation, and deterministic methods, with some frameworks bridging more than one category.

Evolutionary algorithms remain a well-established choice for multi-objective structural optimisation, particularly when balancing economic and performance considerations.Sherif et al.[30] integrate Revit, Dynamo, and Python with a genetic algorithm to minimise combined concrete and steel costs in reinforced concrete buildings, using architectural plans, spatial constraints, material properties, loading conditions, and dimensional limits as inputs. Similarly, Kanyilmaz et al.[31] employ the Non-dominated Sorting Genetic Algorithm(NSGA-II) to simultaneously optimise construction cost, embodied carbon, and usable free space, driven by basic project parameters such as site dimensions, number of floors, soil conditions, material characteristics, and gravity loads.

Generative deep learning methods extend the capability of structural design automation by learning spatial and topological patterns directly from data.The ApproxiFramer model by Ampanavos et al.[32]'s ApproxiFramer applies a CNN–ResNet–LSTM architecture to iteratively generate column positions from low-resolution, sketch-like plans, trained on an augmented synthetic dataset and optimised for both positional and categorical accuracy. InZhang et al.,[33] StrucTopo-GAN combines a pix2pixHD Generative Adversarial Network(GAN) for node detection with a Graph Neural Network (GNN)-based generator–discriminator pair to predict beam connectivity, reconstructing realistic topology graphs from paired architectural and structural plan datasets. Building on the same generative paradigm, FrameGAN by Fu et al.,[34] uses a two-stage pix2pixHD GAN workflow to sequentially produce column layouts and bracing configurations for steel frame–braced structures, validating results through expert evaluation, material cost comparison, and structural performance analysis.

Hybrid strategies bridge the adaptability of machine learning with the efficiency of optimisation.Shan et al.[35] demonstrate this integration by combining metaheuristic searches with LightGBM surrogate models trained on finite element analysis results. This reduces computational cost while guiding the search toward minimum-weight steel frame designs under constraints on drift, stress, stability, and geometry.

Deterministic optimisation approaches, while less data-driven, offer mathematically rigorous control over layout refinement.Meng et al.[36] adopt an Optimality-Criterion-based method

11. Shan et al., 'Integrated Method for Intelligent Structural Design of Steel Frames Based on Optimization and Machine Learning Algorithm'; Sherif et al., 'Automated BIM-Based Structural Design and Cost Optimization Model for Reinforced Concrete Buildings.'

12. Shan et al., 'Integrated Method for Intelligent Structural Design of Steel Frames Based on Optimization and Machine Learning Algorithm.'

13. Kanyilmaz et al., 'A Genetic Algorithm Tool for Conceptual Structural Design with Cost and Embodied Carbon Optimization.'

for column placement in slab–column systems, starting from a candidate grid and iteratively removing inefficient supports using finite element–derived stress data, while maintaining stiffness balance and ensuring smooth stress distribution.

Taken together, these strategies illustrate a progression from conventional optimisation toward increasingly data-driven and hybridised methods, reflecting a broader shift in structural design automation toward approaches that integrate domain-specific engineering knowledge with the adaptive learning capabilities of modern AI. However, a notable limitation across existing studies is the lack of consideration for functional criteria—such as spatial usability, circulation efficiency, or architectural intent—when optimising structural layouts in the early design stages. In practice, the early phases of building design demand a harmonious balance between architectural and structural systems, ensuring that structural efficiency is achieved without compromising the functional and aesthetic qualities of the space.

2. 3. Automated Generation of IFC Models

In the reviewed literature, automated IFC model generation has been implemented using a variety of toolkits and platforms.Doukari and Greenwood[37] employed the XBIM toolkit to programmatically construct IFC entities from enriched semantic data. The approaches by Urbieta et al.[38] andLeon-Garza et al.[39] utilised the open-source IfcOpenShell library to translate extracted geometric and semantic information into IFC-compliant models. Furthermore, Zhao et al.,[40] a custom IFC creation platform developed in C# was used to generate BIM objects from detected structural components and associated attributes. These tools collectively illustrate the range of available solutions, from specialised open-source libraries to bespoke software environments, for converting processed building data into standardised IFC models.

2. 4. Summary of Gaps and Research Direction

The literature highlights promising advances in raster-to-BIM automation, structural layout generation, and IFC model creation, as summarised in,yet, several gaps remain:

- Purely data-driven approaches often rely on extensive, manually labelled datasets that are costly and time-consuming to produce. The proposed hybrid method aims to achieve comparable performance using a more compact dataset, thereby alleviating the burden of large-scale data preparation.
- Few structural layout algorithms explicitly integrate **functional and spatial criteria**, which are essential in early-stages design.
- Existing IFC workflows vary in interoperability and robustness, with limited optimisation for early-phase architectural data.

To address these gaps, the present research adopts a hybrid detection pipeline (YOLO + rule-based refinement) for efficient floor plan interpretation, introduces a functionality-oriented structural layout algorithm, and employs IfcOpenShell for streamlined IFC generation. Additionally, the influence of image resolution on detection accuracy and computational speed was examined by repeating the training process on multiple input sizes.

14. Spyridon Ampanavos et al., 'Structural Design Recommendations in the Early Design Phase Using Machine Learning,' preprint, 2021, https://doi.org/10.48550/arXiv.2107.08567; Fu et al., 'Dual Generative Adversarial Networks for Automated Component Layout'; Kanyilmaz et al., 'A Genetic Algorithm Tool for Conceptual Structural Design with Cost and Embodied Carbon Optimization'; Zhang et al., 'End-to-End Generation of Structural Topology for Complex Architectural Layouts with Graph Neural Networks.'

15. Mohamed Elagiry et al., 'IFC to Building Energy Performance Simulation: A Systematic Review of the Main Adopted Tools and Approaches,' *8th Conference of IBPSA*, 2020.

16. Sofia Feist et al., 'Automatic Reconstruction of 3D Models from 2D Drawings: A State-of-the-Art Review,' *Eng* 5, no. 2 (2024), https://doi.org/10.3390/eng5020042%3Furlappend=%3Futm_source=researchgate.

3. Methodology

This research develops a hybrid algorithm combining deep learning and rule-based logic to automatically convert simplified 2D floor plan images into initial 3D IFC models, alongside a structural layout recommendation system for early-stage design. The workflow comprises

Table 1. Summary of previous studies on IFC model generation from 2D drawings.

	Study	Objectives	Methodology	Input Data	Final Results
1	(Zhao et al., 2021)	BIM reconstruction from existing 2D structural drawings	Faster R-CNN detection and IFC generation in C#	500 structural 2D drawings	Precision 94%, recall 91%, mAP 90.41%, weighted mAP 91.28%
2	(Barreiro et al., 2023)	Semantic 3D reconstruction from 2D floor plans	Faster R-CNN for door/window detection, FPN for walls	5,000 labeled floor plans from CubiCasa5K	IoU 80%
3	(Urbieta et al., 2023)	BIM model generation from architectural plans	Mask R-CNN for component detection, IFC creation via IfcOpenShell	50,000 training images over 2,500 epochs	mAP 96.7%
4	(Kippers et al., 2021)	3D building modelling using CityJSON and deep learning	U-Net, Fast-SCNN, CentreNet, RetinaNet for component detection	5,000 CubiCasa floor plans	F1-scores: walls 0.90, doors 0.93, windows 0.85
5	(Jang et al., 2020)	Indoor space reconstruction from floor plans, conversion to IndoorGML and CityGML	CNNs (DarkNet53, Deeplabv3, PSPNet), vector graph extraction, corner and centreline detection	319 floor plan images	IoU: doors 0.7805, walls 0.688 (ensemble: 0.7283)
6	(Gimenez et al., 2016)	IFC reconstruction from scanned architectural drawings using rule-based methods	C++ implementation, wall through geometric rules	90 scanned floor plans	Detection accuracy: walls 86%, openings 62%; Jaccard index: walls 0.76, openings 0.53
7	(Leon-Garza et al., 2022)	IFC generation by combining Type-2 fuzzy logic with CNN classification	Fuzzy inference system and wall/background classification	Architectural floor plan images	Accuracy: fuzzy logic 97.5%, CNN 99.3%; fuzzy logic improved decision transparency
8	(Doukari & Greenwood, 2020)	BIM model enrichment using knowledge base and aerial imagery	building density analysis, IFC completion via XBIM API	Aerial images of building sites	Extraction of unit types and materials, insertion into IFC via API
9	(Q. Lu et al., 2020)	Geometric digital twinning using rule-based techniques	Column detection with Circle Hough Transform, morphological operations	Scanned 2D CAD architectural drawings	Four output text files for IFC generation, evaluated by extraction and matching accuracy
10	(Park & Kim, 2021)	3D modelling through hybrid data-driven and rule-based approach	Logistic regression, TensorFlow object detection API, geometric rules	30 training and 30 testing plans from a dataset of 110,000 raster plans	Wall detection accuracy >95%, dimensional accuracy >97%
11	(Liu et al., 2017)	Accurate raster-to-CAD vector conversion with geometric/semantic constraints	ResNet152, Integer Programming for constraint enforcement	870 manually labeled architectural plans	Opening detection accuracy 67%, recall 91.4%
12	(Z. Lu et al., 2021)	Interpretation of rural architectural plans using joint neural network and MIQP	Component detection, room segmentation, topological graph extraction	800 real-world floor plans	Overall accuracy 87–92%, IoU 74–82%
13	(Zhao et al., 2020)	Structural component detection from 2D drawings	YOLO, pre-trained on ImageNet	450 structural drawings	Precision >80%, recall >90%

three main sections:
3.1. Floor Plan Interpretation: detection of openings (doors and windows) using YOLOv11, and wall extraction using OpenCV-based geometric rules, combined through a lightweight Graphical User Interface (GUI) that allows interactive sketching and editing.
3.2. Structural Layout Recommendation: generation of initial column placements in Grasshopper, guided by parking requirements, opening positions, and spatial constraints.
3.3. IFC Model Generation:transformation of processed vector data into IFC files via IfcOpenShell.
The methodology and its main phases are illustrated in Figure 3.

3. 1. Floor Plan Interpretation

The first stage of the proposed framework focuses on extracting building components from simplified 2D floor plan raster images through a dual approach combining deep learning-based object detection and rule-based geometric processing, a hybrid strategy that reduces the need for extensive data labelling while maintaining accuracy in raster-to-BIM tasks.[41] To fa-

Table 2. Summary of Previous Studies on Structural Layout Recommendation.

	Study	Methodology	Inputs	Outputs	Limitations
1	(Sherif et al., 2022)	Integration of BIM, structural design, and a genetic algorithm within the Revit–Dynamo environment using Python scripting	Architectural plan, performance limits, material properties, structural loads, dimensional constraints	Optimised structural layout, section dimensions for columns, reinforcement and concrete quantities, cost estimation	Limited to regular rectangular plans; suitable for late design stages
2	(Kanyilmaz et al., 2022)	NSGA-II multi-objective optimisation implemented in Python	Plot dimensions, material type, loads, number of floors, soil conditions, building function	Column and beam layout, slab type, material usage, total cost, embodied carbon	Limited to rectangular plans with uniform spans
3	(Ampanavos et al., 2021)	CNN model with LSTM for stepwise prediction of columns from plan images	Floor plan image, previously generated columns, pixel coordinates	Column coordinates	Predicted columns may not align with existing walls
4	(Zhang et al., 2022)	GAN combined with GNN for generating structural topology from floor plan images	Architectural floor plan image	Column positions and beam connections	Column placement deviation from intended axes
5	(Fu et al., 2023)	Two-stage GAN with pix2pixHD architecture for column and bracing arrangement	Stage 1: architectural plan; Stage 2: column layout and paired structural images	Column and bracing layout	—
6	(Shan et al., 2023)	Combination of machine learning models and classical optimisation with finite element analysis	Geometric parameters of I-shaped members, gravity and seismic loads, code-based constraints	Optimised structural member design in terms of lightness, cost-effectiveness, and code compliance	Suitable for late design stages
7	(Meng et al., 2021)	Optimality-Criterion algorithm for column layout optimisation	Slab geometry, material properties, number of columns, initial candidate grid	Optimal column locations based on stiffness and preservation of architectural form	Architectural plan and functional considerations not integrated into column placement

cilitate user interaction and enable direct plan input with precise placement of architectural elements such as openings, a graphical user interface (GUI) was developed using Konva.js. Through this interface, users can draw their desired floor plan, which is then processed within the proposed framework. The framework automatically detects and classifies building components, generates a structurally aligned column layout based on the recognised geometry, and ultimately produces a 3D model in IFC format. The interface layout and its main features are illustrated in Figure 4.

3. 1. 1. YOLO-Based Object Detection for Openings

Following best practices in recent floor-plan detection literature, we generated a synthetic dataset from RPLAN-inspired layouts and applied geometric augmentations (mirroring, rotation, scaling, translation) to improve generalisation. Synthetic dataset strategies follow recent large-scale floor-plan pipelines that combine programmatic layout generation and geometric augmentation to improve model generalisation. The synthesised dataset comprises 9,600 annotated images and was partitioned into 70/30 train /test splits for all experiments (see Appendix A for dataset examples). The YOLOv11Lmodel, the latest version of YOLO for the detection task released by Ultralytics, was then trained on this dataset for 100 epochs with a learning rate of 0.0001, a batch size of 32, and an image size of 768 pixels, using the defaultpre-trained weights on the COCO dataset. The YOLO model was chosen as it generates a separate label file for each image, unlike models such as Fast R-CNN or Mask R-CNN that store all annotations in a single JSON. This structure simplifies automated dataset generation in Grasshopper and

17. Lucile Gimenez et al., 'Automatic Reconstruction of 3D Building Models from Scanned 2D Floor Plans,' *Automation in Construction* 63 (2016): 48–56, https://doi.org/10.1016/j.autcon.2015.12.008.

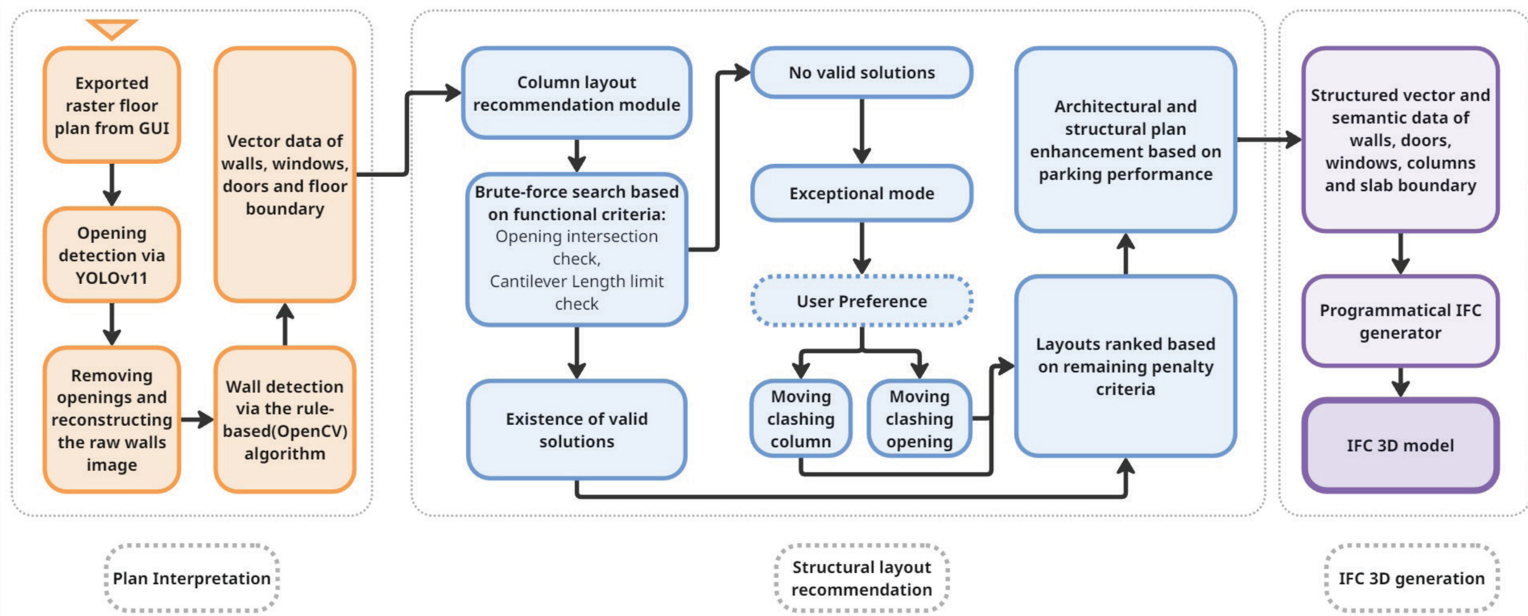


Figure 3. Methodology phases.

supports efficient large-scale data preparation. YOLOv11 was selected as it offers the highest detection accuracy among previous YOLO versions, according to Ultralytics documentation.

To evaluate the influence of input resolution on both detection performance and computational efficiency, the training process was repeated at two additional resolutions: 512 and 864 pixels. YOLO-family detectors have been shown to deliver strong trade-offs between speed and detection accuracy on architectural floor plans, particularly when combined with multiscale training and dataset augmentation strategies.[42]

3. 1. 2. Data Preparation and Labelling Process

The dataset generation process was fully automated using a custom algorithm implemented in Grasshopper,with 200 floor plans from the RPLAN dataset serving as the primary source for data generation. For placing doors and windows in the dataset, existing walls in each plan were first filtered based on length to identify those long enough to host an opening. Then, for each plan, a random selection of 1 to 4 walls longer than 3 metres was chosen to host windows, and 1 to 4 walls longer than 1 meter were selected to host doors. The bounding rectangles of the openings were automatically drawn in each plan, allowing the extraction of coordinate information for the YOLO dataset, which was saved automatically. Since a dual approach combining deep learning and rule-based processing was used for floor plan interpretation, there was no need to label the walls for segmentation separately. This eliminated the requirement for a dedicated wall dataset and significantly reduced the time-consuming labelling process.

To introduce variability and better represent realistic architectural scenarios, a different number of openings was assigned to each floor plan, ensuring diversity in placement and density. Furthermore, the generated plans were scaled at three factors (0.8, 0.9, and 1.0) and rotated in multiple directions. These variations enabled the YOLO model to learn to detect objects across different sizes and orientations, enhancing its robustness and accuracy in recognising openings in diverse floor plan layouts.

3 .1. 3. Rule-Based Wall Extraction

Following the detection of doors and windows using the trained YOLO model, the next step involved wall identification through a rule-based algorithm and vectorisation in Grasshopper. The detection process relied on distinguishing white pixels, representing walls, from the black background. To achieve this, the YOLO output underwent preprocessing, in which detected openings were removed and the corresponding wall segments reconstructed. The image was then converted into a binary format, ensuring that walls were represented in white and the background in black, thereby facilitating accurate image processing.

The binarised image was imported into Grasshopper, where a rule-based algorithm, implemented via OpenCV, extracted wall positions by tracing white pixel boundaries against the black background. These boundaries were then converted into sets of points and subsequently into polyline vector paths for further processing.

In parallel, the YOLO predictions for openings were processed to restore their geometric posi-

18. Qiuchen Lu et al., ‘Semi-Automatic Geometric Digital Twinning for Existing Buildings Based on Images and CAD Drawings,’ *Automation in Construction* 115, no. 4 (2020), https://doi.org/10.1016/j.autcon.2020.103183.

19. Zhao et al., ‘Reconstructing BIM from 2D Structural Drawings for Existing Buildings.’

20. Barreiro et al., ‘Automatic Reconstruction of Semantic 3D Models from 2D Floor Plans.’

21. Martin Urbieta et al., ‘Generating BIM Model from Structural and Architectural Plans Using Artificial Intelligence,’ *Journal of Building Engineering* 78, no. 107672 (2023).

22. R. G. Kippers et al., ‘Automatic 3d Building Model Generation Using Deep Learning Methods Based on CITYJSON and 2d Floor Plans,’ *The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences* XLVI-4-W4-2021 (2021): 49–54, https://doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-49-2021.

tions within the plan. The model's output files, containing object class and normalised bounding box coordinates, were read into Grasshopper and converted into numerical lists. By denormalising these values relative to the image dimensions and aligning them with a defined plan origin, each opening was accurately drawn as a rectangular representation in its correct spatial location.

3. 1. 4. Extracting Relevancies

To ensure that each opening is accurately matched with its corresponding host wall, a two-step procedure was applied. First, due to inevitable discrepancies between the predicted coordinates of openings (obtained from the deep learning model) and their actual host walls in the Grasshopper environment, a geometric correction was introduced. Each wall was enclosed in a rectangular boundary generated using the Centre Box command, with its length equal to the wall's length and a constant width of 1.5 metres, a limit that was experimentally tuned. Whenever the centroid of an opening fell within this boundary, that wall was designated as its host. Subsequently, the centroid of the opening was projected onto the wall's centreline to guarantee precise alignment. This process was performed independently for doors and windows. Figure 5 shows the overall process of data extraction from the floor plan raster image.

In the implementation stage, the name of the host wall for each opening was extracted by midpoint matching: the midpoint of every detected opening host wall was compared against the placement coordinates stored in the JSON file of walls. When a correspondence within the predefined tolerance was found, the script assigned the respective wall's name to the opening. This ensured that each door and window in the output JSON was explicitly linked to its correct host wall.

23. Hanme Jang et al., 'Indoor Reconstruction from Floorplan Images with a Deep Learning Approach,' *International Journal of Geo-Information* 9, no. 2 (2020), https://doi.org/10.3390/ijgi9020065%3Furlappend=%3Futm_source=researchgate.

24. Chen Liu et al., 'Raster-to-Vector: Revisiting Floorplan Transformation,' *2017 IEEE International Conference on Computer Vision (ICCV)* (Venic), 2017, https://doi.org/10.1109/ICCV.2017.241.

25. Zhengda Lu et al., 'Data-Driven Floor Plan Understanding in Rural Residential Buildings via Deep Recognition,' *Xiaopeng* 567 (2021): 58–74, https://doi.org/10.1016/j.ins.2021.03.032.

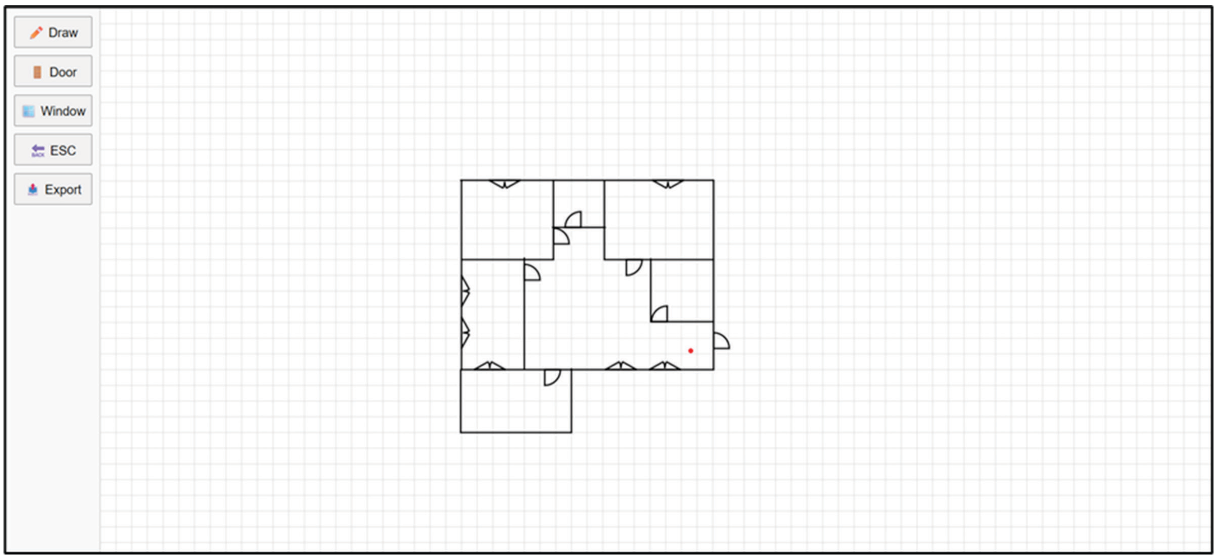


Figure 4. Developed GUI Environment for Plan Drawing.

3. 1. 5. OptimisationParameters and Penalty Functions

Although evolutionary multi-objective methods (e.g., NSGA-II) have been widely applied to structural optimisation,[43] discrete axis permutations in small-n combinatorial search may produce unstable convergence; hence, exhaustive screening with strict feasibility filtering was preferred. Candidate structural axes were extracted from wall positions and encoded as binary variables (0 = inactive, 1 = active). For n axes, 2^n permutations defined potential layouts (e.g., 512 configurations for nine axes). Column positions were derived from intersections of active axes within the floor boundary. To ensure that the optimised solution satisfies both structural and architectural requirements, six penalty functions are defined:

(1) Column–opening conflicts
(2) Cantilever lengths exceeding limits
(3) High variance in span lengths
(4) Spans outside permissible ranges
(5) Excessive column count
(6) Columns outside the plan boundaries

User-defined limits for span ranges and cantilever lengths provided adaptability across design contexts.

3. 1. 6. Optimisation Filtering and Exceptional Case Strategy

After structuring the permutations of potential structural axes and their associated penalty values, the next step involves identifying the optimal layout. Due to the discrete and non-gradual nature of variable changes, evolutionary algorithms such as the Genetic Algorithm failed to produce stable optimal solutions. Therefore, a brute-force search approach was adopted, evaluating all possible configurations while applying a strict screening system to eliminate infeasible options and prioritise solutions with lower penalty scores. The screening process consists of two main stages. In the first stage, any configuration violating hard constraints—such as exceeding the maximum allowable cantilever length or directly intersecting with existing openings—is removed. In the second stage, the remaining layouts are ranked based on a hierarchical evaluation of penalty criteria, with the most acceptable solutions placed at the top of the shortlist. Finally, the algorithm returns the five best-performing column arrangements as the final recommendation set.

Given the significant influence of parking

26. Yunfan Zhao et al., 'A YOLO-Based Method to Recognize Structural Components from 2D Drawings,' paper presented at Construction Research Congress: Computer Applications, Tempe, 2020, https://doi.org/10.1061/9780784482865.080.

27. Hugo Leon-Garza et al., 'A Type-2 Fuzzy System-Based Approach for Image Data Fusion to Create Building Information Models,' *Information Fusion* 88 (2022): 115–25, https://doi.org/10.1016/j.inffus.2022.07.007.

28. Omar Doukari and David Greenwood, 'Automatic Generation of Building Information Models from Digitized Plans,' *Automation in Construction* 113, no. 103129 (2020), https://doi.org/10.1016/j.autcon.2020.103129.

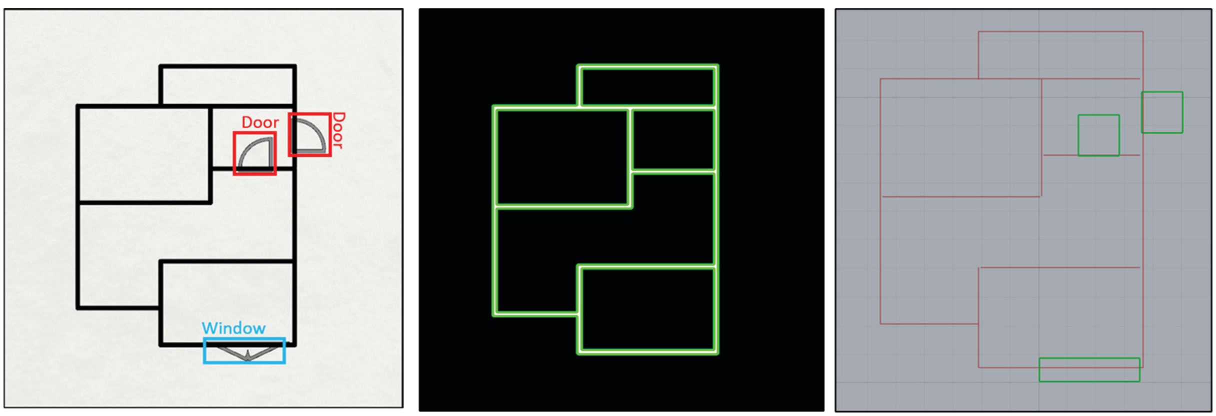


Figure 5. Data extraction process: (a) Opening detection using YOLO, (b) Wall extraction using OpenCV, (c) Vector data.

requirements on structural layouts, the optimisation process integrates parking design regulations. Based on Section 4 of the *National Building Regulations of Iran, Topic Four: General Building Requirements*,[44] the minimum centre-to-centre spacing between columns is set to 5 m for two parking bays and 7.5 m for three bays. Consequently, axis positions are adjusted to align with multiples of 2.5 m, ensuring compatibility with single,double- and triple-vehicle layouts. Internal walls and their associated openings are then aligned with the nearest parallel structural axes, while four boundary axes remain fixed to preserve the site limits.

In cases where no arrangement satisfied the strict constraints, an exceptional mode was activated, as shown in Figure 6. This mode allowed the introduction of off-grid columns and provided the user with the option to resolve conflicts either by relocating the opening or by shifting the column. In either case, the nearest potential structural axis perpendicular to the chosen relocation direction was identified, and the element was repositioned along this axis so that conflicts were removed while maintaining the regularity of the initial grid. This interactive adjustment ensured that even under highly constrained conditions, the final configuration preserved both architectural coherence and structural feasibility. Through this dual-stage screening process, combined with an exception-handling strategy, the proposed approach ensures that the selected structural layout achieves both geometric alignment with the architectural plan and compliance with performance-based constraints.

3. 2. IFC Model Generation

The final stage of the proposed framework focused on generating the IFC model from the processed floor plan data. All extracted and refined geometric information from Grasshopper was exported in a structured JSON format, automatically produced by custom scripts to ensure direct compatibility with the IFC generation workflow. This JSON file contained both geometric and semantic attributes for each architectural element, including walls, openings (doors and windows), columns, and slabs. For every element, properties such as name, type, dimensions (length, height, thickness), spatial coordinates (position and orientation),

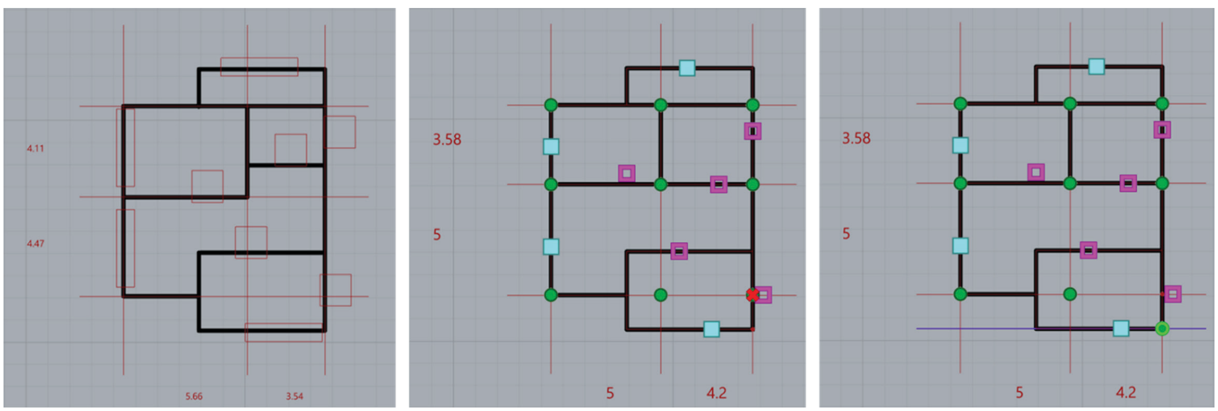


Figure 6. Plan adjustment based on parking performance on an exceptional mode.

29. Park and Kim, '3DPlanNet: Generating 3D Models from 2D Floor Plan Images Using Ensemble Methods.'

30. Sherif et al., 'Automated BIM-Based Structural Design and Cost Optimization Model for Reinforced Concrete Buildings.'

31. Kanyilmaz et al., 'A Genetic Algorithm Tool for Conceptual Structural Design with Cost and Embodied Carbon Optimization.'

32. Ampanavos et al., 'Structural Design Recommendations in the Early Design Phase Using Machine Learning.'

33. Zhang et al., 'End-to-End Generation of Structural Topology for Complex Architectural Layouts with Graph Neural Networks.'

34. Fu et al., 'Dual Generative Adversarial Networks for Automated Component Layout.'

35. Shan et al., 'Integrated Method for Intelligent Structural Design of Steel Frames Based on Optimization and Machine Learning Algorithm.'

36. Meng et al., 'Optimizing Support Locations in the Roof–Column Structural System.'

37. Doukari and Greenwood, 'Automatic Generation of Building Information Models from Digitized Plans.'

38. Urbieta et al., 'Generating BIM Model from Structural and Architectural Plans Using Artificial Intelligence.'

39. Leon-Garza et al., 'A Type-2 Fuzzy System-Based Approach for Image Data Fusion to Create Building Information Models.'

and, where relevant, dependencies on other elements (e.g., the host wall for an opening) were defined.Table3 presents the data inputs required for each element category in the IFC creation process. The proposed workflow for automated IFC model generation is presented as a pseudocode in Algorithm 1.

Using the IfcOpenShell Python library, the JSON data was programmatically transformed into a structured IFC model. IfcOpenShell with Python scripting is now a common, robust approach to programmatic IFC generation that preserves both geometry and semantic attributes. This process ensured that the resulting 3D BIM model was ready for use in downstream design, coordination, and analysis workflows.

4. Results and Discussion

In this study, a three-stage framework was designed and implemented in which an initial architectural floor plan image is first interpreted through a dual approach, followed by the generation of an initial structural column layout based on functional design criteria, and finally, the automatic, programmatic conversion of 2D components into a 3D IFC model—a standard, interoperable format supported by BIM software.

Accordingly, this chapter presents the results and final evaluation of the research, focusing on these three main stages. In the first stage, the proposed hybrid method for floor plan interpretation is assessed in terms of accuracy, execution speed, and computational cost. The second stage evaluates the performance of the column layout recommendation algorithm by applying it to four real-world executed projects and comparing the proposed layouts with the actual column arrangements in the construction drawings. Finally, the third stage analyses the efficiency of IFC model generation in terms of speed and computational cost, comparing the proposed method with conventional IFC modelling approaches.

4. 1. Floorplan Interpretation Evaluation

In the rule-based component, the wall detection algorithm achieved a 100% accuracy rate in correctly identifying wall lines. Compared to the study by, which reported an 86% accuracy, this represents a substantial improvement in extracting precise linear structures from 2D plans.The results of the deep learning module were assessed using four metrics: precision, recall, mAP50, and mAP50:95. Precision measures the proportion of correctly predicted positive instances among all predictions the model labelled as positive. A high precision value indicates fewer false positives.Recall represents the proportion of correctly identified positive instances among all actual positive instances in the dataset. A high recall score means the model rarely misses actual elements, even if it produces some false positives.mAP50 (mean Average Precision at IoU = 0.5) is the average of the precision values calculated across all classes when the Intersection over Union (IoU) threshold is set to 0.5. This metric evaluates whether

Table 3. Data inputs for IFC generation in IfcOpenShell.

Category	Data Inputs
IfcWall	name, length, height, placement, x_axis, z_axis, predefined_type
IfcSlab	slab_outline, slab_depth,
IfcWindow, IfcDoor, IFCOpeningElement	name, type, length, height, thickness, placement, x_axis, z_axis, host_wall_name
IfcColumn	name, size, height, placement, x_axis, z_axis

the predicted bounding boxes sufficiently overlap with the ground truth, providing a measure of overall detection accuracy under a moderate standard.mAP50-95 extends this evaluation by calculating mean Average Precision across multiple IoU thresholds, ranging from 0.5 to 0.95 in steps of 0.05. This stricter metric requires higher overlap between predictions and ground truth to count as correct, and thus offers a more rigorous assessment of a model's localisation accuracy and robustness.

A comparison of the trained YOLO model's performance in this study with results from previous research (Table4) shows that, although the precision value of 0.63 is lower than in some earlier works, the recall of 0.97 is significantly higher[45]. The high recall indicates that the model successfully detected almost all actual doors and windows in the floor plans, whereas the relatively low precision reflects false positivecases where the model predicted the presence of an element that was not actually in the image. This limitation may stem from visual similarities between classes, low-quality or noisy images, or an imbalance in the number of training samples. To address this, several improvements are suggested, such as increasing the diversity and quality of training data, fine-tuning the confidence threshold, applying post-processing filters to remove low-confidence detections, and including training images deliberately devoid of certain classes to improve the model's understanding of 'object absence' and thus reduce false positives. Overall, the model demonstrated better performance in detecting windows compared to doors, likely due to the greater geometric detail in window forms, which makes them more distinguishable from the background.

```
Procedure Build_IFC()

file ← create IFC4 file
assign units + contexts
create site, building, storeys

define surface styles (wall/slab/door/window/column)

# Walls
load Walls.json
for each wall:
  create IfcWall
  assign to Ground Storey
  add wall representation (extrusion)
  place via placement_matrix
  apply wall style

# Slab (as IfcFooting)
load Slab.json
create IfcFooting
assign to Ground Storey
profile + extrude geometry
apply slab style

# Openings (voids only)
load Opening.json
for each opening:
  create IfcOpeningElement
  assign to Ground Storey
  add rectangular void geometry
  place via placement_matrix
  link to host via IfcRelVoidsElement

# Doors / Windows (no Fills relation)
create IfcDoor / IfcWindow from JSON where type matches
assign to Ground Storey
add representation + style

# Columns
load Columns.json
create IfcColumn
assign to Ground Storey
square profile + extrude
place + style

# Multi-storey replication
duplicate_elements_to_storeys(walls, slab, doors, windows,
columns)
duplicate_openings_to_storeys(openings)

file.write(output_path)
```

Algorithm 1. Automated multi-storey IFC model generation from JSON-based architectural and structural data.

40. Zhao et al., 'Reconstructing BIM from 2D Structural Drawings for Existing Buildings.'

41. Leon-Garza et al., 'A Type-2 Fuzzy System-Based Approach for Image Data Fusion to Create Building Information Models.'

42. Myunghyun Jung et al., 'Analysis of the Floor Plan Dataset with YOLOv5,' *Journal of the Korean Society for Industrial and Applied Mathematics* 26 (2023): 311–23, https://doi.org/10.12941/jksiam.2023.27.311; Zhongguo Xu et al., 'Multiscale Object Detection on Complex Architectural Floor Plans,' *Automation in Construction* 165, no. 105486 (2024), https://doi.org/10.1016/j.autcon.2024.105486.

In terms of accuracy metrics, the model achieved an mAP50 of 0.902, indicating precise localisation and sizing of elements at the 0.5 IoU threshold, comparable to prior studies.[46] The mAP50-95 score of 0.816—higher than those reported in previous works.[47]—confirms the model's robustness in maintaining high prediction accuracy under stricter IoU thresholds. These results collectively demonstrate that the proposed dual-approach method effectively combines rule-based precision for wall extraction with deep learning's high recall for opening detection, resulting in a reliable interpretation of floor plan images.In terms of computational efficiency, the optimal model processed each image in an average of 0.0041 seconds, outperforming previous studies utilising YOLO[48] by 99% and Faster R-CNN[49] by 46%.

Notably, since the YOLO model was trained on images with dimensions of 640×640 pixels for opening detection, reducing the input resolution is expected to increase the likelihood of errors in identifying either the presence or the precise location of these elements. In contrast, for wall detection using the rule-based algorithm, a reduction in resolution does not affect performance, provided that the input image is free from noise.

4. 1. 1. Hyperparameter Sensitivity Analysis

In the training process of the YOLO model, the outcomes were significantly influenced by the selected hyperparameter values; however, the model exhibited varying levels of sensitivity to each parameter. In this study, the number of epochs was set to 100. Nevertheless, due to an early stopping threshold of 20 epochs and the absence of improvements in evaluation metrics between epochs 60 and 80, the training process was terminated at epoch 80. This indicates that increasing the number of epochs did not yield meaningful performance gains and merely increased computational costs. The batch size was set to 32, which provided a suitable balance between training speed and gradient stability. Although larger batch sizes could potentially enhance learning, they would also considerably increase computation time and resource requirements. Among all hyperparameters, the learning rate was identified as the most influential factor in determining model performance. A very small initial learning rate of 0.0001 was adopted to ensure stability in reaching optimal weights and to prevent high fluctuations in results, despite reducing the convergence speed. Furthermore, a weight decay value of 0.0005 played a crucial role in controlling model complexity; higher values reduced overfitting, whereas lower values impaired generalisation. Overall, the sensitivity analysis revealed that the final accuracy of the model was most dependent on the learning rate parameters (Lr0 and lrf) and the weight decay.

4. 1. 2. Noise Sensitivity Analysis

To evaluate the impact of image resolution on model performance, the YOLOv11L network was trained on three input sizes: 512, 768, and 864 pixels. The results, illustrated in **Figure 7,** indicate that the total precision reached its peak value of 0.637 at a resolution of 768 pixels, with no further improvement at higher resolutions. Similarly, mAP50 achieved its highest score of 0.902 at 768 pixelsand remained nearly unchanged when the image size was increased to 864. The mAP50–95 metric exhibited a moder-

43. Yaping Lai et al., 'BIM-Based Intelligent Optimization of Complex Steel Joints Using SVM and NSGA-II,' *Journal of Constructional Steel Research* 223, no. 109086 (2024), https://doi.org/10.1016/j.jcsr.2024.109086.

44. *National Building Regulations of Iran, Topic Four: General Building Requirements*, Road, Housing & Urban Development Research Center, 2017.

45. Liu et al., 'Raster-to-Vector: Revisiting Floorplan Transformation'; Zhao et al., 'A YOLO-Based Method to Recognize Structural Components from 2D Drawings'; Zhao et al., 'Reconstructing BIM from 2D Structural Drawings for Existing Buildings.'

46. Urbieta et al., 'Generating BIM Model from Structural and Architectural Plans Using Artificial Intelligence'; Zhao et al., 'Reconstructing BIM from 2D Structural Drawings for Existing Buildings.'

47. Barreiro et al., 'Automatic Reconstruction of Semantic 3D Models from 2D Floor Plans'; Jang et al., 'Indoor Reconstruction from Floorplan Images with a Deep Learning Approach.'

ate rise from 0.755 to 0.816 as the resolution increased from 512 to 768, but did not show further enhancement beyond that point. In contrast, the recall metric demonstrated a steady improvement, increasing from 0.979 to 0.996 as resolution grew. Although the overall detection accuracy metrics were only slightly affected by resolution, the computational speed showed a significant variation—rising from 0.0024 SPI at 512 to 0.0041 SPI at 768 and 0.0051 SPI at 864. These results suggest that a resolution of 768 pixels offers the optimal balance for this dataset, achieving the highest precision and mAP50 values while maintaining a favourable processing speed.

4. 1. 3. Assessment of Geometric Relation Extraction

The performance evaluation of the proposed algorithm for extracting geometric relationships indicates that this method, by relying on the detection of the central point of an opening within the boundaries of its host wall, could accurately identify the corresponding host wall and correctly classify the opening type (door or window) in all experiments. The extracted geometric relationships play a critical role in generating the IFC model, as the accurate creation of elements in the model depends on the precise association of each opening with its host wall.

4. 1. 4. Efficiency Gains through the Hybrid Detection Pipeline

The dual approach proposed in this research, despite relying on a simpler labelling process, achieved enhanced accuracy in opening detection compared to previous studies and reached 100% precision in wall recognition. Notably, the object labelling and dataset generation process—which is typically a labour-intensive and time-consuming task—was automated through the developed data preparation method, achieving an average generation speed of 1.27 images per second. This performance represents an acceptable rate for such computationally demanding processes.

48. Zhao et al., 'A YOLO-Based Method to Recognize Structural Components from 2D Drawings.'

49. Zhao et al., 'Reconstructing BIM from 2D Structural Drawings for Existing Buildings.'

Table 4. Comparison of the Performance of the Proposed Deep Learning Model with Previous Studies.

Method	Precision	Recall	mAP50	mAP50-95	Number of Initial Data	Hardware Capacity (GB)	Computational Speed (seconds/ image)
Proposed model in this study (YOLO)	0.63	0.97	0.902	0.816	9,600	Nvidia A100 (80)	0.0041
YOLO (Zhao et al., 2020)	0.80	0.90	—	—	450	Nvidia GeForce GTX 1080 (8)	0.63
Faster R-CNN(Zhao et al., 2021)	0.94	0.91	0.90	—	4000	GeForce RTX 2080 Ti (11)	0.0077
Faster R-CNN (Barreiro et al., 2023)	—	—	—	0.80	5,000	—	—
Mask R-CNN (Urbieta et al., 2023)	—	—	96.7	—	50,000	Nvidia A100	0.0002
CNN (Jang et al., 2020)	—	—	—	0.78	319	—	—
ResNet (Liu et al., 2017)	0.67	0.91	—	—	870	—	—

4. 2. Evaluation of the Proposed Column Layout Recommendation Algorithm

Beyond producing automated recommendations, the framework is designed to actively involve users in the decision-making process. Designers can define key parameters—such as allowable cantilever length and permissible span range—and adjust the relative hierarchies of penalty functions that guide the sorting procedure. This flexibility enables them to prioritise criteria such as structural regularity, span uniformity, or minimal column count according to project-specific needs. From the ranked set of five optimal layouts generated for each plan, users can then select their preferred arrangement.

To evaluate the effectiveness of this interactive recommendation process, the algorithm was tested on four real-world architectural floor plans from previously executed projects. For each case study, the algorithm produced five candidate solutions, with the configuration exhibiting the highest similarity to the actual built layout selected as the representative output for assessment. A geometric matching criterion was employed to quantify similarity, where a proposed column location was considered a match if it fell within a predefined threshold radius (e.g., 50 cm) of any actual column location. The overall similarity percentage was subsequently calculated as the ratio of matched columns to the total number of proposed columns.

The results demonstrated that, in three out of the four examined cases—including one involving exceptional design conditions—the proposed algorithm achieved complete alignment (100%) with the real-world column arrangement. In the remaining case, the match rate was 83.33%, still indicative of a high degree of agreement with structural engineering decisions. Overall, the algorithm attained an average prediction accuracy of 95.83% across the four case studies, which is competitive with prior research, including[50] (97.79%),[51] (97%), and[52] (93.6%) performance score relative to expert evaluations). A visual comparison between the proposed and actual layouts is presented in Figure8.

In terms of ranking the final valid column layouts, the predefined prioritisation scheme yielded satisfactory results in most experimental cases. However, altering the order of these priorities led to variations in the outcomes: in some cases, producing configurations with higher applicability, while in others reducing the overall quality of the results. Consequently, it was determined that the order of the criteria should be made user-adjustable, thereby allowing practitioners to tailor their influence according to the specific requirements of each project.

4. 3. Performance and Applicability of the Proposed IFC Model Generation Algorithm

Figure 9 illustrates the workflow results on three sample floor plans, from plan interpretation through column recommendation to 3D IFC generation. Visual inspection indicates that the generated IFC model occasionally exhibits minor deficiencies in details—such as the connection of orthogonal walls—which can be resolved through refinement of the input data. Nevertheless, in conventional BIM-authoring software such as Revit, the process of producing an IFC model, even for experienced users, may require several hours to an entire working

50. Ampanavos et al., 'Structural Design Recommendations in the Early Design Phase Using Machine Learning.'

51. Zhang et al., 'End-to-End Generation of Structural Topology for Complex Architectural Layouts with Graph Neural Networks.'

52. Fu et al., 'Dual Generative Adversarial Networks for Automated Component Layout.'

day, depending on the plan's complexity. In contrast, the algorithm developed in this research, implemented using the IfcOpenShell library, is capable of converting vectorised floor plan data into a 3D IFC model—including walls, doors, windows, floors, and columns—in less than one minute. This initial model can subsequently be semantically enriched with additional data for use across a wide range of structural and energy analysis software, such as Simergy and ETABS, as well as other IFC-based modelling tools. Providing such a model in the early design stages, particularly during conceptual phases, can play a significant role in fostering coordination among multidisciplinary teams and facilitating engineering decision-making. Figure 10illustrates the application of the model in Revit for generating preliminary material estimates.

4. 4. Computational Resources

In terms of computational requirements, the deep learning model was trained on an NVIDIA A100 GPU with 80 GB of memory using Google Colab. The complete training process required approximately 2.7 hours,

which is considered efficient given the dataset size and model complexity. The column arrangement recommendation algorithm, in turn, was implemented on an NVIDIA GeForce RTX 3070 GPU with 8 GB of RAM.

5. Limitations

This research introduces a three-stage framework that improves upon previous methods in floor plan interpretation, structural layout suggestion, and automated IFC model generation. By integrating geometric logic with a lightweight YOLO-based detection model, the proposed approach achieves competitive accuracy with high processing speed. The column layout algorithm uniquely incorporates architectural and spatial performance criteria, producing arrangements that are both structurally efficient and spatially coherent. The IFC generation process further accelerates early design workflows and enhances multidisciplinary collaboration by providing a ready-to-use, BIM-compliant 3D model. However, the framework has certain limitations that will be discussed in this section.

5. 1. Limitations in Input Data

In this study, certain architectural features—such as spatial functions, enclosed or semi-open configurations, the presence of voids, staircases, and other specialised elements—were not incorporated into the modelling process. Including these features in future studies could enhance the accuracy of analyses and improve the model's alignment with real-world project conditions. From the perspective of plan image interpretation, the YOLO model employed in

Figure 7. Comparison of achieved metrics (Precision, Recall, mAP@50, mAP@50–95) across three training runs using input resolutions of 512, 768, and 864.

this study is currently capable of detecting only a limited and specific set of door and window symbols. Other elements, such as furniture or uncommon symbols, fall outside the model's recognition scope. Expanding the range of detectable classes could improve the model's accuracy and applicability when processing diverse architectural plans. Furthermore, the current plan geometry analysis algorithm is compatible only with orthogonal structures and straight walls, and does not support plans with unconventional angles, hand-drawn irregular lines, or curved walls. Extending the capability to recognise and model such structures represents a key avenue for future development of this research.

5. 2. Limitations in Column Arrangement Recommendation

In the structural layout recommendation section, the optimisation algorithm is designed solely based on initial geometric and functional criteria, without incorporating technical and structural considerations such as load-bearing capacity, stability, ductility, or code compliance. Integrating structural requirements with functional criteria could enhance the accuracy and reliability of the proposed algorithm. Another limitation of the algorithm is the extended processing time required for complex initial plans; as plan complexity increases, the algorithm's runtime grows nonlinearly. Figure 11illustrates the execution time of the algorithm for plans

(a) 100% alignment in the fourth recommendation

(b) 100% alignment in the first recommendation

(c) 83.33% alignment in the fourth recommendation

(d) 100% alignment in the first recommendation on exceptional mode

Figure 8. Results of the Algorithm for Initial Column Placement in Four Built Plan Case Studies.

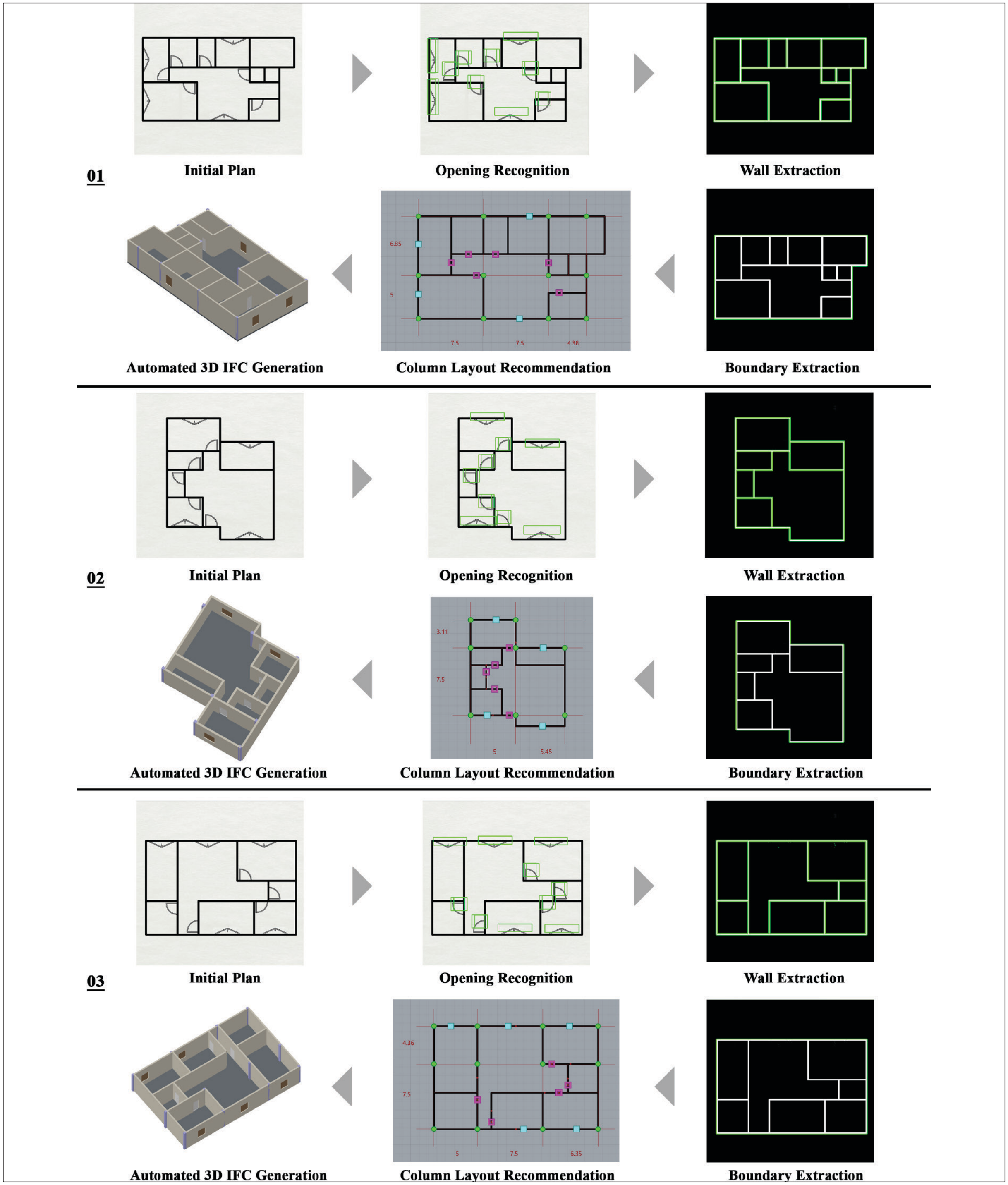


Figure 9. Overall workflow on sample plans.

with varying numbers of layouts. Modifications in the solution generation structure and the filtering of invalid solutions, combined with the adoption of metaheuristic approaches, could reduce processing time.

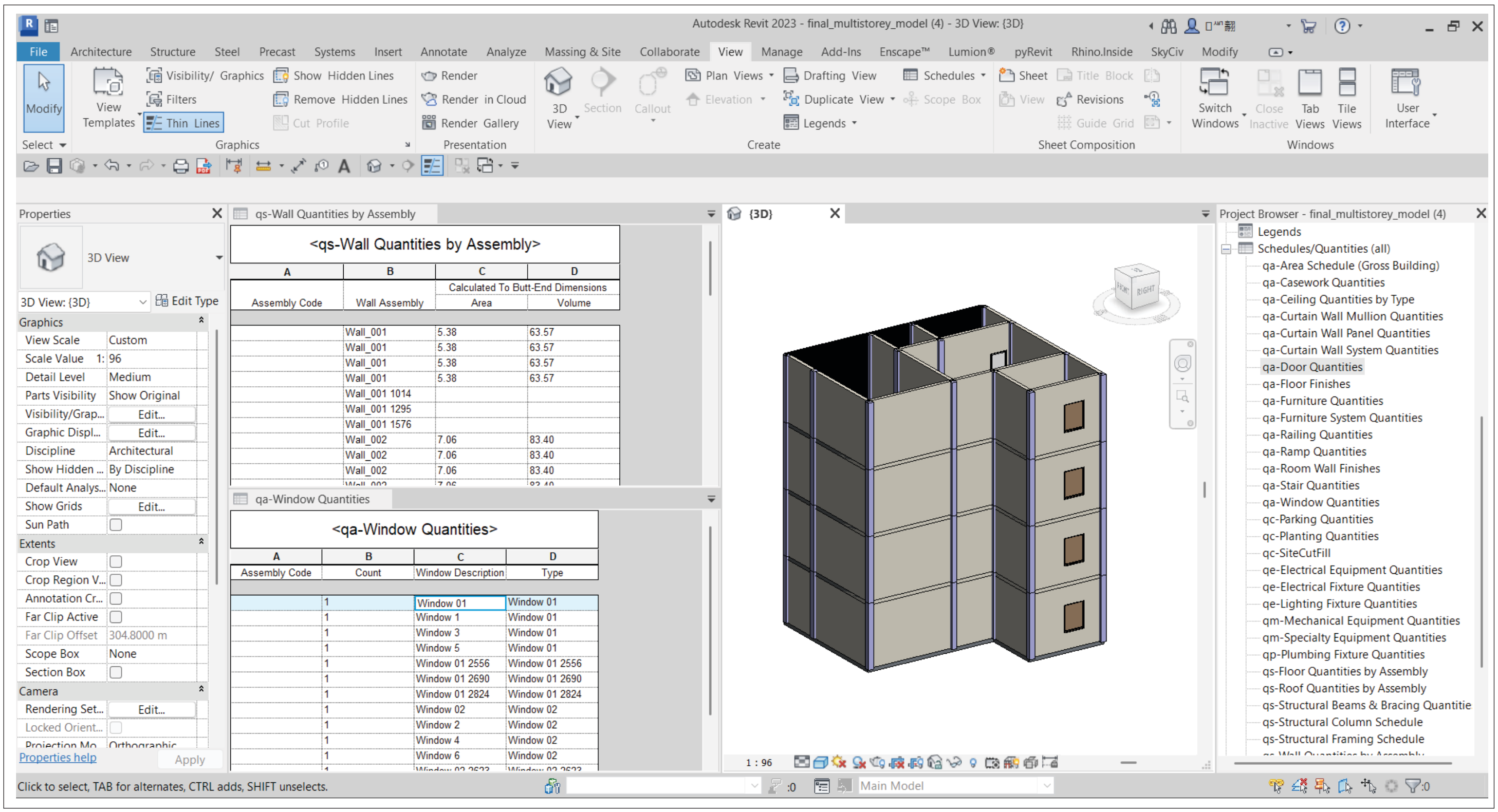
5. 3. Limitations in IFC Model Generation

In the IFC generation algorithm, horizontal structural elements, such as beams, have not been considered. Incorporating these components in future developments could significantly enhance the completeness of the structural model and its analytical capabilities. Moreover, to improve the compatibility of the generated 3D model with structural and energy analysis software, it is essential to enrich the model across multiple dimensions, enabling accurate and comprehensive analyses.

6. Conclusion and Further Work

The developed framework introduces an automated three-stage pipeline for converting simplified 2D architectural floor plans into fully BIM-compliant IFC models, integrating hybrid drawing interpretation, a performance-oriented column layout suggestion algorithm, and a structured IFC generation strategy. In this workflow, the purpose of IFC creation is not merely

Figure 10. Application of the IFC model in Revit for generating preliminary material estimates.



E-ISSN: 2645-5900 *Soffeh Journal*, Shahid Beheshti University, Vol. 36, Issue 2, No. 113, 2026

the geometric reconstruction of building components; instead, it establishes a reliable data backbone that enables direct interoperability with widely adopted multidisciplinary platforms. The resulting IFC models seamlessly support architectural modelling and coordination in Revit, Archicad, Navisworks, and BlenderBIM; structural design and assessment in Robot Structural Analysis, Tekla Structures, and ETABS; and energy-performance simulation in environments such as Simergy, IES-VE, and RIUSKA.

By enabling these analytical pathways at the conceptual design stage, the proposed pipeline significantly accelerates early decision-making, facilitates communication among architects, engineers, and energy analysts, and strengthens the foundation for performance-driven design exploration long before detailed modelling typically begins. The evaluation results confirmed its competitive accuracy, high computational efficiency, and adaptability to practical design contexts, highlighting its potential to enhance collaboration across disciplines and reduce rework in downstream phases.

However, current constraints—including the limited scope of recognised architectural symbols, the assumption of orthogonal geometries, the absence of certain structural elements in the output, and increased processing time in highly intricate plan configurations—still restrict the full generalisation of the approach. Future extensions focused on enriched object recognition, support for non-standard and free-form geometries, deeper incorporation of structural and energy-performance metrics, and improved semantic modelling can further elevate the applicability and intelligence of this framework. With these advancements, the pipeline can evolve into a domain-ready solution that effectively bridges the gap between early-stage design intent and comprehensive BIM-based analytical workflows, ensuring that IFC generation becomes a catalyst for the integrated, high-quality development of the built environment.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

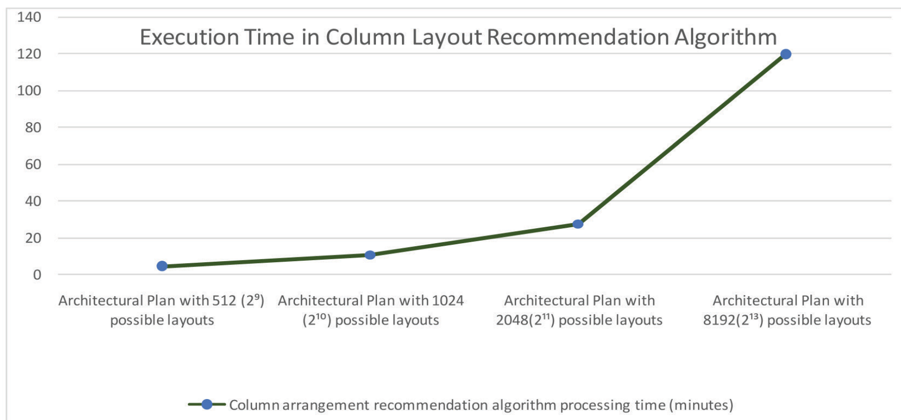
Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work, the authors used ChatGPT (OpenAI) as a writing assistant in order to improve the clarity, coherence, and readability of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Data Availability

Data will be made available on request.

Figure 11. Execution Time in Column Layout Recommendation Algorithm.



References

Lim, J. O. I. E., P. A. T. R. I. C. K. Janssen, and R. U. D. I. Stouffs. '*Automated generation of BIM models from 2D CAD drawings*.' (2018) *(CAADRIA)*. https://doi.org/10.52842/conf.caadria.2018.2.061

Liu, Chen, Jiajun Wu, Pushmeet Kohli, and Yasutaka Furukawa. 'Raster-to-vector: Revisiting floorplan transformation.' *In Proceedings of the IEEE international conference on computer vision*, pp. 2195-2203. 2017. https://doi.org/10.1109/ICCV.2017.241

Lu, Qiuchen, Long Chen, Shuai Li, and Michael Pitt. 'Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings.' *Automation in Construction* 115 (2020): 103183. https://doi.org/10.1016/j.autcon.2020.103183

Lu, Zhengda, Teng Wang, Jianwei Guo, Weiliang Meng, Jun Xiao, Wei Zhang, and Xiaopeng Zhang. 'Data-driven floor plan understanding in rural residential buildings via deep recognition.' *Information Sciences* 567 (2021): 58-74. https://doi.org/10.1016/j.ins.2021.03.032

Meng, Xianchuan, Ting-Uei Lee, Yulin Xiong, Xiaodong Huang, and Yi Min Xie. 'Optimizing support locations in the roof–column structural system.' *Applied Sciences* 11, no. 6 (2021): 2775. https://doi.org/10.3390/app11062775

National Building Regulations of Iran. *Topic four: General building requirements*. Road, Housing & Urban Development Research Centre. (2017)

Padwal, P. Building Information Modelling (BIM) market size & outlook. *Leadership and Management Engineering*. (2024). https://www.researchgate.net/publication/377969878_Building_Information_Modelling_BIM_Market_Size_Outlook

Park, Sungsoo, and Hyeoncheol Kim. '3DPlanNet: generating 3D models from 2D floor plan images using ensemble methods.' *Electronics* 10, no. 22 (2021): 2729. https://doi.org/10.3390/electronics10222729

Shan, Wenchen, Jiepeng Liu, and Junwen Zhou. 'Integrated method for intelligent structural design of steel frames based on optimization and machine learning algorithm.' *Engineering structures* 284 (2023): 115980. https://doi.org/10.1016/j.engstruct.2023.115980

Sherif, Mohamed, Khaled Nassar, Ossama Hosny, Sherif Safar, and Ibrahim Abotaleb. 'Automated BIM-based structural design and cost optimization model for reinforced concrete buildings.' *Scientific Reports* 12, no. 1 (2022): 21616. https://doi.org/10.1038/s41598-022-26146-6

Urbieta, Martin, Matias Urbieta, Tomas Laborde, Guillermo Villarreal, and Gustavo Rossi. 'Generating BIM model from structural and architectural plans using Artificial Intelligence.' *Journal of Building Engineering* 78 (2023): 107672. https://doi.org/10.1016/j.jobe.2023.107672

Wang, Dejiang, Jinzheng Liu, Haili Jiang, Panpan Liu, and Quanming Jiang. 'Existing Buildings Recognition and BIM Generation Based on Multi-Plane Segmentation and Deep Learning.' Buildings 15, no. 5 (2025): 691. https://doi.org/10.3390/buildings15050691

Xu, Zhongguo, Naresh Jha, Syed Mehadi, and Mrinal Mandal. 'Multiscale object detection on complex architectural floor plans.' *Automation in Construction* 165 (2024): 105486. https://doi.org/10.1016/j.autcon.2024.105486

Zhang, Chong, Mu-Xuan Tao, Chen Wang, and Jian-Sheng Fan. 'End-to-end generation of structural topology for complex architectural layouts with graph neural networks.' *Computer-Aided Civil and Infrastructure Engineering* 39, no. 5 (2024): 756-775. https://doi.org/10.1111/mice.13098

Zhao, Yunfan, Xueyuan Deng, and Huahui Lai. 'A YOLO-based method to recognize structural components from 2D drawings.' In Construction Research Congress 2020, pp. 753-762. Reston, VA: *American Society of Civil Engineers*, 2020. https://doi.org/10.1061/9780784482865.080

Zhao, Yunfan, Xueyuan Deng, and Huahui Lai. 'Reconstructing BIM from 2D structural drawings for existing buildings.' *Automation in Construction* 128 (2021): 103750. https://doi.org/10.1016/j.autcon.2021.103750