Automatic BIM detailing using deep features of 3D views

Sou-Han Chen[1] and Fan Xue[2]

Abstract

Building information modeling (BIM) detailing, the process of adding the level of graphical and non-graphical details, is required in many BIM stages and applications; however, manual BIM detailing is a resource-intensive and costly process. This study proposes an automatic BIM detailing method based on deep features (DFs) of BIM 3D views in three steps. First, a BIM's 3D view and semantics were extracted automatically. Then, machine learning (ML) algorithms learned the DFs to predict the target BIM's invisible details. Finally, the details were automatically added to BIM by a Dynamo program. A case study of motion-bearing component detailing for 86 doors through three DFs and five ML algorithms revealed that DFs improved the automatic detailing results comprehensively (29 out of 32 scenarios) and significantly. This paper's contribution includes an effective, novel approach for automatic BIM detailing as well as quantified experimental evidence about the effectiveness of DFs for BIM applications.

Highlights

- An automatic BIM detailing method using machine learning (ML) and deep features (DFs).
- A case study of motion-bearing component detailing for 86 door types in NBS BIM Library.
- Experiments conducted for 7 detailing tasks using 5 ML algorithms and 3 DFs.
- Comprehensiveness: 29 out of 32 ML-task combinatorial scenarios improved by DFs.
- Significance: averagely 47.7% error reduction for classification and 18.8% for regression.

Keywords

Building information modeling; BIM detailing; deep features; motion-bearing components; NBS BIM library

[1] Sou-Han CHEN, Research Assistant
Department of Real Estate and Construction, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
Email: souhan@hku.hk, ORCiD: https://orcid.org/0000-0003-0511-7518

[2] Fan XUE, Assistant Professor, Ph.D.
Department of Real Estate and Construction, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
Email: xuef@hku.hk, ORCiD: https://orcid.org/0000-0003-2217-3693

*: Corresponding author, Tel: +852 3917 4174, Fax: +852 2559 9457, Email: xuef@hku.hk

1 Introduction

The architecture, engineering, construction, and operation (AECO) sector is pursuing a digital transformation to evolve the traditional workflow (NIC 2017; CIC 2021). Building information modeling (BIM) is ‘a comprehensive digital representation of physical and functional characteristics of a facility’ (NIBS 2015). In addition to the geometric 3D model, BIM can store various types of non-geometric data, relationships, and documentation. The benefits and impacts of BIM adoption have been agreed upon and discussed by researchers worldwide (Barlish & Sullivan 2012; Ghaffarianhoseini et al. 2017). BIM has had a positive impact on many construction applications, such as workflow standardization, information integration, project management, and data analytics. Therefore, the AECO industry has advocated and promoted the importance of mandating BIM around the world (Chan et al. 2019) and is gearing up with online and offline BIM libraries (Kim et al. 2015).

BIM detailing refers to the process of increasing the level of detail (LOD) for construction projects. Unlike the level of development (LoD), referring to the development of reliable and concrete BIM at different lifecycles of the project, LOD refers to the accurate details of BIM. In general, the development of the LOD of BIM will increase progressively as the project moves on. The design stage will primarily have BIM with lower LOD, while during the construction period, BIM will be further enhanced and detailed by engineers from different disciplines. The process of BIM detailing is labor-intensive and time-consuming, leading to a need for an automation process. Currently, the performance and applications of BIM heavily depend on the quality, semantics, and LOD of BIM. Meanwhile, research also emphasized the importance of information depth and pointed out the consequences of information loss (Borrmann et al. 2018).

BIM detailing is vital to the BIM lifecycle. The insufficient LOD of BIM can lead to the failure of performing time dynamic simulation for digital twin applications. A digital twin is a virtual representation of a physical object or system across its lifecycle, using real-time data to enable understanding, learning, and reasoning (NIC 2017). The International Council for Building also suggests that the digital twin is a prominent topic in the AECO sector as it provides a cyber-physical integration of the dynamic building status, including the construction and operation phase (Seaton et al. 2022). For example, a door without detailed hinges falls rather than opens after a simulated push, as shown in Figure 1. In contrast, the constraint from hinges leads to a correct motion behavior in a digital twin. Nevertheless, due to the low added value of manually created high LOD of BIM objects, many BIM objects might not have adequate components of information needed in the digital twin applications. One distinct example would be the motion-bearing components. Without the motion-bearing components, the digital twin might not be able to accurately present and visualize the collected real-time data. Therefore, an efficient method for automatic BIM detailing should be developed.

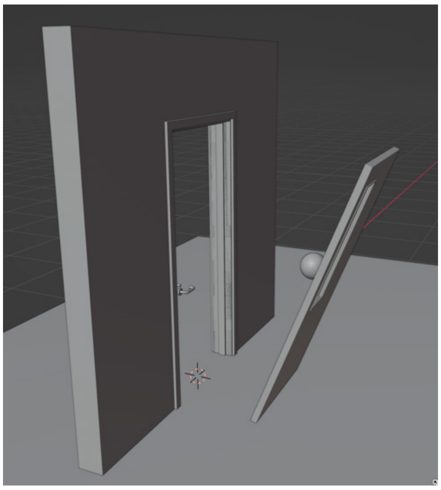

Figure 1. A failed door simulation resulting in falling without hinge details in the BIM

Machine learning (ML), including deep learning, has been demonstrated to be a set of effective approaches to facilitate and automate BIM detailing and enrich construction information. ML and deep learning can be used to automate and optimize BIM applications (Zabin et al. 2022) and predict missing construction data (Yang et al. 2021). Recent studies (Dargan et al. 2020) showed that deep features (DFs) could be successfully integrated into many applications (e.g., image recognition, semantic image segmentation, and object detection). The results of such applications can be further applied in assisting BIM parametric design through various tools. For example, Dynamo from Autodesk Revit provides a visual programming environment to quickly perform a data-driven parametric BIM design process (Kensek 2015).

This study presents an automatic BIM detailing approach through the exploitation of the target BIM's DFs. To validate the presented approach, we collected a set of training BIM objects from the NBS National BIM Library (NBS 2022). Based on the learning results of DFs in the training of BIM objects, the missing BIM components can automatically be predicted as three classification tasks and four regression tasks; and created via a Dynamo program. The contribution of this paper is two-fold. First, from the perspective of knowledge, the outcomes of this paper confirmed that DFs could considerably facilitate the ML-based BIM detailing; for example, all the classification tasks were improved by 30.4~58.7% on average, while four regression tasks were improved by 9.5~33.6%. For BIM practitioners, automatic detailing based on the BIM sketch design can escalate productivity and save costs in terms of human resources and machine hours.

Following this introduction, Section 2 provides a literature review of BIM, detailing deep learning applications for BIM. Section 3 presents a proposed three-step BIM detailing approach, that is, feature extraction, component parameters prediction, and parametric design. Section 4 shows a case study of detailing door hinges with a training set of 86 BIM objects, together with evaluations and analyses of various deep features and ML methods. Section 5 discusses the findings and potential impacts, along with the limitations. Section 6 concludes this study and points out future works.

2 Literature review

2.1 BIM detailing

BIM has been increasingly adopted in the AECO sector. On the one hand, BIM provides a systematic information hub for various project practitioners to collaborate at the same pace. On the other hand, BIM is considered a 'blind and deaf' model as the information stored inside cannot be easily synchronized with actual built assets as the project progresses (Chen et al. 2015). To alleviate such static, outdated, and lacking detail issues in BIM, the concepts of as-built BIM and digital twin building emerged (Wu et al. 2021). The as-built BIM provides up-to-date information on the current built assets (Xue 2022), while the digital twin provides a systematic and dynamic virtual environment for retrieving, simulating, and manipulating different levels of information. However, there are two main gaps between BIM and digital twin: the LOD of the model and the capability of handling the dynamic data collected from reality (Opoku et al. 2021).

To bridge the gaps between a 'blind and deaf' BIM and digital twin, academic research and industrial applications in the literature have reported three aspects of BIM detailing, as contrasted in Table 1. An example group of research is BIM integration with other technologies, such as light detection and ranging (LiDAR) (Xue et al. 2019), Internet of Things (IoT) (Tang et al. 2019), geographic information system (GIS) (Wang et al. 2019), and machine learning (ML) (Zabin et al. 2022); BIM data exchange and interoperability (Lou et al. 2020), and optimization (Lou et al. 2020); BIM interactive application through virtual reality (VR) (Zhang et al. 2021) and gaming technology (Potseluyko et al. 2022). On the other way, as the guidelines and standards of BIM have been developed by both international and local institutions (CIC 2020; ISO 2018), the application of BIM in the industry is booming and based mostly on the stage of the project: the design stage focuses on the simulation and planning of the models; the construction stage focuses on BIM detailing, 4D BIM, and project monitoring; and the operation stage focuses on the documentation and maintenance of the project. As construction projects by nature are dynamic and contain lots of variables and uncertainties, the industry is seeking a more dynamic BIM to reveal an accurate and instant project status, such as motion and load analysis during the design and construction stage.

Table 1. BIM detailing research direction

BIM detailing aspects	Research direction	Technologies adoption
Semantic enrichment	BIM integration	LiDAR, IoT, GIS, ML
Semantic constitution	BIM interoperability & optimization	BIM
Semantic implementation	BIM simulation & visualization	VR, gaming technology

The detail of the model is critical to the successful implementation of BIM in managing dynamic projects. A higher LOD of BIM enables different applications to perform more dynamically and store more semantic information in the models, which can further affect and assist stakeholders in the decision-making process (Boje et al. 2020) as well as provide engineers a comprehensive project status (Lu et al. 2020). However, as BIM detailing is time-consuming and requires interdisciplinary engineers to create detailed components and collaborate throughout the project, an appropriate LOD of BIM would be preferred and suggested based on different stages of the project (Fai & Rafeiro 2014). Similarly, researchers suggested different BIM applications should imply different LOD to achieve cost savings (Hong et al. 2019). Therefore, some components and items are often neglected during BIM detailing, as they will not affect the construction process or technical analysis or conflict with any stakeholder's interest.

2.2 Machine learning and deep learning applications for BIM

Machine learning (ML) refers to the scientific study of algorithms and models that enable computer systems performing certain tasks without predefined and explicitly programmed processes (Han et al. 2011). These algorithms and models can be categorized into different groups based on how they are being trained and their applications. For applications in the AECO sector, there are three categories, instance-based ML, rule/tree-based ML, and function-based ML (Hall et al. 2009) that are commonly used for data prediction, image processing, simulation, and modeling. In all, the conventional ML techniques require structured and well-organized input data.

Deep learning (DL) techniques, in contrast, allow models to learn representations of raw data with multiple levels of abstraction (LeCun et al. 2015). The capabilities of deep learning to analyze data, provide insights, optimize workflows, and make predictions have become a new method for the AECO sector to process BIM data across the BIM lifecycle. For the design stage, deep learning has been proven to be an effective method to perform generative design (Bianconi et al. 2019; Chen et al. 2022), clash prediction along with resolution (Hu & Castro-Lacouture 2019), and compliance checking (Song et al. 2018) by manipulating BIM data. For the construction stage, deep learning can provide cost estimation (Banihashemi et al. 2022) and optimize project schedules (Torres-Calderon et al. 2019) by fully utilizing the information stored in BIM. Lastly, during the operation stage, deep learning can be used to classify the building type (Lomio et al. 2018), enrich semantic information in the as-built BIM (Xue et al. 2021; Xue 2022), automatically conduct the property valuation (Su et al. 2021), and assist the defect surveying (Valero et al. 2018). As deep learning progressively develops, more potential applications for BIM will be explored.

Since current technologies of deep learning cannot directly use native BIM-format data as the training data and output data, an appropriate data exchange method should be adopted for BIM. Currently, many BIM plug-ins have been developed and can fill this gap, such as Grasshopper for Rhinoceros, PARAM-O for ArchiCAD, and Dynamo for Revit. These plug-ins enable users to use the external data for parametric design and export the needed data for analyses (Natanian et al. 2019). Take Dynamo as an example; it provides a graphical interface for users to perform parametric-related functions of BIM objects (Kensek 2015). Research also shows that Dynamo can effectively facilitate complex model generation (Yang et al. 2020) and extract data for operational usage (Sadeghi et al. 2019; Thabet et al. 2022). Another illustrated the adoption of Dynamo to facilitate the process of 3D concrete printing (Weng et al. 2021).

In summary, as the adoption of BIM increases in the AECO sector, various BIM applications have been explored and adopted rapidly, including the digital twin of the project. However, the current industrial practice requires great resources to achieve a higher LOD of BIM to perform dynamic and accurate project status. Meanwhile, the rapid development and application of deep learning technologies in BIM have proven to be effective in facilitating BIM-related works. As a result, inspired by previous research, this project will utilize deep features to fill in the missing motion components in BIM objects to increase the interactive and dynamic performance of BIM applications.

3 The proposed automatic BIM detailing method

3.1 Overview

The proposed method aims to automatically create detailed BIM components with the deep features of 3D views. Figure 2 shows the developed three-step method for the automated generation of BIM details. The three steps are (1) characteristic 3D views of BIM, (2) deep learning of BIM details, and (3) parametric BIM detailing. Each step has a set of inputs, which in part depend on the prior step.

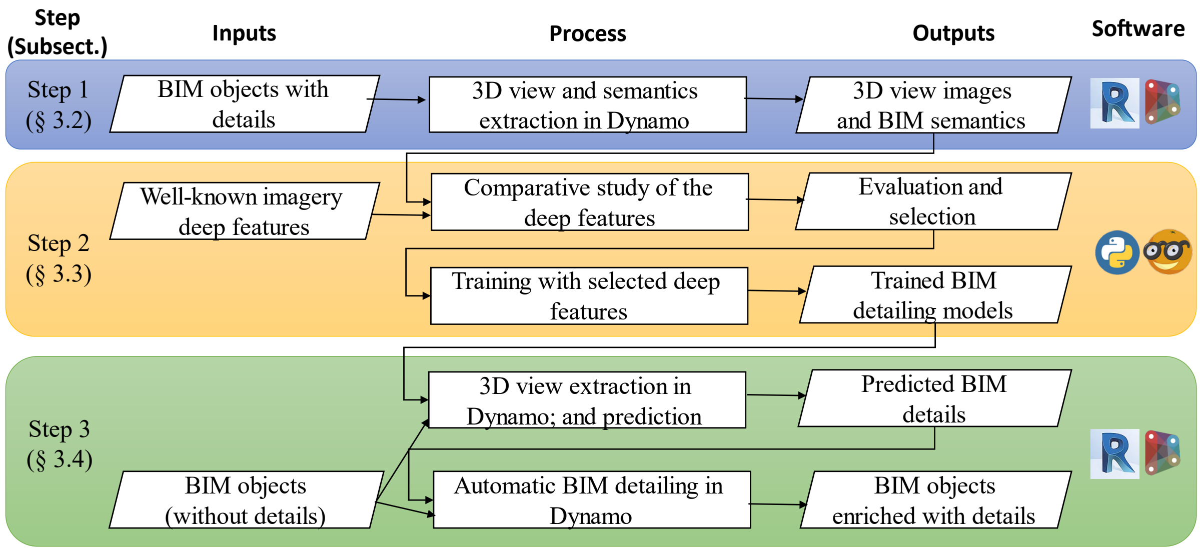

Figure 2. Method of the proposed three-step automatic BIM detailing using deep features

3.2 BIM 3D views and semantics extraction

The first step extracts the semantics and 3D views of an input BIM for the subsequent deep learning processes. BIM can be considered a well-structured information hub (Heaton et al. 2019). However, the BIM software ecosystem has unsatisfactory interoperability with existing ML and deep learning algorithms in the computer science field. Therefore, the semantics and 3D view imagery of an input BIM must be converted from the BIM object to machine learning-friendly formats. Given a BIM object with expected details in BIM software, users can use the built-in functions or plug-ins or self-programming to generate the data for machine learning use. Using Dynamo, a Revit plug-in for extracting data from the BIM family, as an example, the general steps are shown in figure 3. First, the Dynamo scripts enable Revit to load family documents in batches. The 3D views of the BIM objects can be generated by image export Revit, or users can predefine the angle and export images through Dynamo or self-programming in C#. Meanwhile, the semantic data can be filtered and retrieved through Dynamo or using Revit API. Finally, 3D views in PNG format and parameters in CSV format are extracted, stored, and ready for the next step.

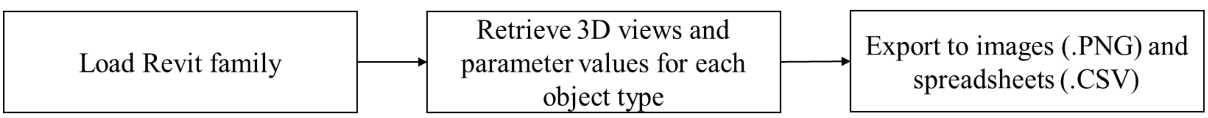

Figure 3. Data conversion from BIM to ML

3.3 DF-based BIM details prediction

This step aims to train the ML algorithms for the capability of predicting the parameter values of BIM objects. To achieve this target, the adoption of DFs was proposed during the ML training process. Three well-known DFs, VGG19, InceptionV3, and SqueezeNet, are investigated in this study. DFs of the images can represent the unstructured image in a tabular format. VGG19 is a typical 19-layer deep convolutional neural network (CNN) proposed in 2014 and widely used in the medical sector for disease diagnosis (Alhindi et al. 2018). InceptionV3 (InceptV3) is a 48-layer deep CNN proposed in 2015 that uses a splitting method to effectively extract the DFs (Dong et al. 2020). SqueezeNet is a 48-layer deep CNN proposed in 2016 that uses fewer parameters for the feature extraction with competitive accuracy and smaller model sizes (Lee et al. 2019).

As shown in Figure 2, a comparative study is conducted to select the most suitable model. The DF-based learning process of the model includes two types of features, 3D views' DFs and BIM semantic features. The prediction targets of BIM details can be broadly classified into two types: discrete values for classification and continuous values for regression. Examples of classification are types of family and materials of parts, while regression is the locational values of the components. This study adopts five representative ML algorithms from three ML categories, as listed in Table 2. K-nearest neighbors (KNN) algorithm is selected as a representative in the instance-based ML category. From the rule/tree-based category, decision tree (Tree) is a typical and well-known ML algorithm, while the eXtreme Gradient Boosting (XGBoost) is selected as a more advanced algorithm that exploits the second order gradient statistics of the loss function. From the function-based category, linear regression (LR) and support vector machine (SVM) are selected for their linear relationship and kernel function models.

Table 2. Selected ML algorithms for BIM details classification and regression

ML category	ML algorithm	Classification	Regression	Reference
Instance-based	KNN	Y	Y	(Zhang & Zhou 2007)
Rule/tree-based	Tree	Y	Y	(Han et al. 2011)
	XGBoost	Y	Y	(Chen & Guestrin 2016)
Function-based	LR	N	Y	(Han et al. 2011)
	SVM	Y	Y	(Hearst et al. 1998)

Both classification and regression training processes are based on 5-fold cross-validation. To evaluate the performance of the trained models, F_1 will be used for the comparison of classification algorithms, while RMSE will be used for the comparison of regression algorithms. Table 3 lists the error metrics and selection in the two processes.

Table 3. Error metrics for BIM details classification and regression

Task	Name	Definition	Selected?
Classification	Precision	the proportion of true positives among instances classified as positive	N
	Recall	the proportion of true negatives among all negative instances	N
	F_1	the weighted harmonic mean of precision and recall	Y
Regression	RMSE	square root of the arithmetic mean of the squares of a set of numbers	Y

A series of comparisons and evolutions of trained models is conducted based on the error metrics. First, whether or not the model is improved can be gleaned by comparing the best F_1 value for classification or best root mean square error (RMSE) value for regression between the control group (i.e., without DF) and the experimental group (i.e., with DF). If the model is improved, the CNN DF and ML algorithm are the best settings for the corresponding tasks. In contrast, if the model is not improved, the adoption of CNN DF might not be suitable for the tasks, and the setting of the ML algorithm becomes the best setting. Once the best training settings are selected, users can input all the existing features – both 3D views and BIM semantics – to train the model. The output of this step is a trained BIM detailing model, which can predict the target BIM detail using 3D views and BIM semantics.

3.4 Parametric BIM detailing

This step aims to enrich the BIM objects by adopting the trained model and the predicted information of the target details. Therefore, the input of this step is a BIM object without target details. Firstly, the 3D views and semantics of the BIM object without details need to be extracted as described in Section 3.2. Then users can apply the trained model in Section 3.3 to predict the discrete or continuous values of the target BIM details.

The predicted values are used to create the BIM objects parametrically, which can be done by programming and plug-ins as shown in Figure 4. For example, Dynamo provides a node where users can create a cylinder by inputting the radius and height; Grasshopper provides a node where users can create a cone by inputting the base plane, the radius at the cone base, and the cone height. Such parametric design can achieve the same goals with less time and effort than manual drawing, especially with many duplicate tasks. Eventually, the output of this step is the enriched BIM objects with details.

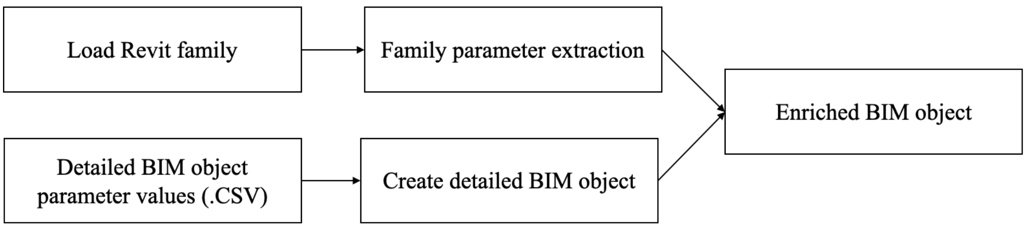

Figure 4. General framework of Dynamo program

4 A study of NBS doors

4.1 Case study

A case study of invisible hinges for swing door families in the UK National Building Specification (NBS) BIM Library was conducted to advance the proposed method. Doors are one of the most common and movable object types in BIM projects, and hinges are the critical motion-bearing details. In the NBS BIM Library, 48 of the 94 door families, consisting of 86 door types according to the parametric settings, conveyed critical details of hinges. The proposed method was tested for predicting and detailing the motion-bearing components, i.e., invisible hinges. The training BIM object set thus was the collection of all 86 door types with hinges. It was a challenging task to detail the hinges correctly for various door types automatically; it was even more challenging in this case since hinges are invisible from the training external 3D views.

Table 4 lists the target details, including quantity, depth, radius, and different positions, to predict hinges. ML algorithms were trained using native BIM semantics and 3D view's deep features. Table 5 lists the three general parameters, NominalHeight, NominalLength, and NominalWidth, of doors defined by NBS BIM Object Standard v2.1 for training the ML algorithms.

Table 4. List of targets to predict for hinge detailing

Task Type	Task name	Definition
Classification	Quantity	The quantity of hinge. (either 3-hinge or 4-hinge)
	Depth	The height of the hinge (cylinder).
	Radius	The radius of the hinge (cylinder).
Regression	Position (Top)	The ratio of the highest hinge Z-position to the height of the door.
	Position (Bottom)	The ratio of the lowest hinge Z-position to the height of the door.
	Position ($Extra_1$)	The ratio of the 2nd-lowest hinge Z-position to the height of the door.
	Position ($Extra_2$)*	The ratio of the 3rd-lowest hinge Z-position to the height of the door.

*Only applicable to the 4-hinge door.

Table 5. Selected BIM parameters

Property name	Property requirements
NominalHeight	A numerical value of the nominal height (typically the vertical characteristic dimension of the product) in millimeters.
NominalLength	A numerical value of the nominal length (typically the primary or larger of the two perpendicular horizontal dimensions of the product) in millimeters.
NominalWidth	A numerical value of the nominal width (typically the secondary or smaller of the two perpendicular horizontal dimensions of the product) in millimeters.

4.2 Experimental settings

The experiments were conducted on a MacBook Pro M1 with a CPU (3.2 GHz, 8-core), 16 GB memory, and MacOS (ver. 12.4). The test software included Autodesk Revit (ver. 2022), Dynamo (ver. 2.12), and Orange data mining (ver. 3.31), as listed in table 6.

Table 6. Experimental environment setting

Software	Version	Functionality
Revit	2022	Loading Revit family documents (rfa.) file and supporting dynamo
Dynamo	2.12	BIM semantics and 3D view image extraction; parametric design for hinge creation

Orange	3.31	Deep feature extraction, machine learning model training, evaluation, and prediction

Figure 5 shows the Dynamo scripts for extracting the door types' 3D views and semantics, respectively. Figure 5(a) shows the script for exporting 3D views. The view angle was set as the top right corner of the exterior side of the door. The image resolution was set to be 1024 x 1024 pixels at 600 DPI. For standardized qualities of the view images, the 'detail level' on the Revit view control bar was selected as 'fine,' the view scale was set as 1:10, and the visual style was set as 'hidden line.' Also, all the annotation categories and the walls model categories were disabled inside the visibility/graphic setting. The final output of the image is the 3D view of the exterior side in PNG format. As shown in Figure 5(b), the three standard NBS parameters, NominalHeight, NominalLength, and NominalWidth, were exported into an index file in the CSV format. Meanwhile, hinge-related BIM parameters (i.e., the quantity, depth, radius, and relative locations) were also extracted into the index file.

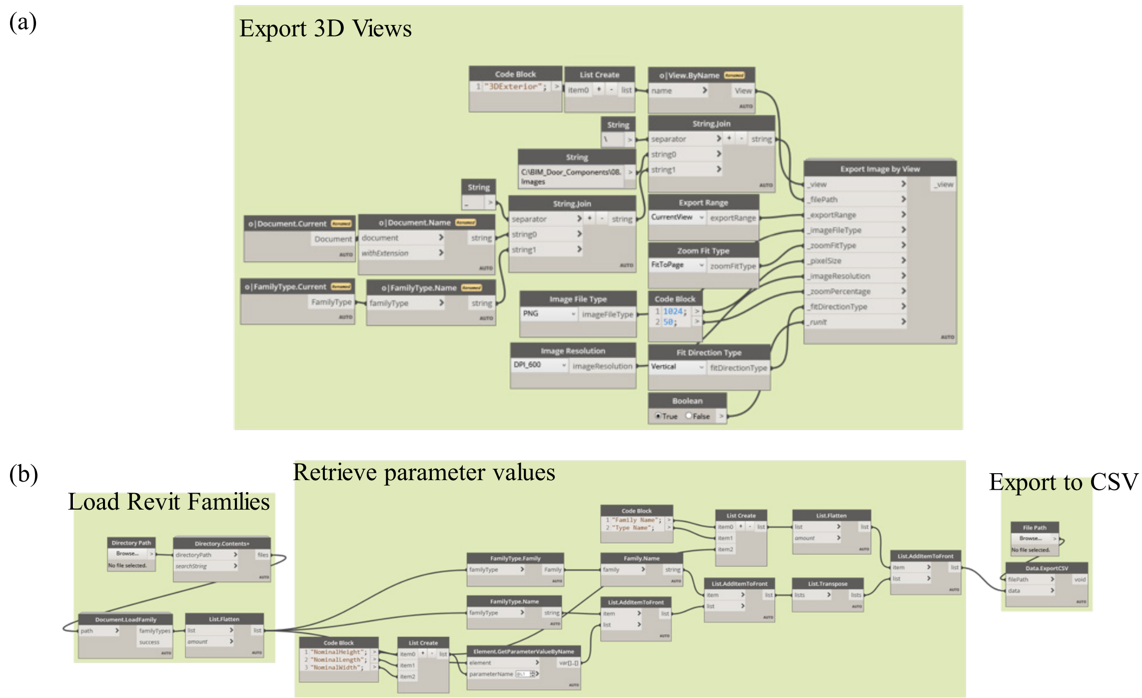

Figure 5. Dynamo scripts in Step 1. (a) Exporting 3D views; (b) Parameter extraction

Figure 6 shows the workflow diagrams of the software Orange for extracting the DFs of the 3D view images and comparisons of ML settings. For classification tasks of the quantity, radius, and depth of the hinge, Figure 6(a) shows four ML algorithms were involved, while all five ML algorithms were tested for regression, as shown in Figure 6(b). The left dashed box in blue color represents the extraction of the DFs from 3D view images, while the right dashed box in orange represents the ML process.

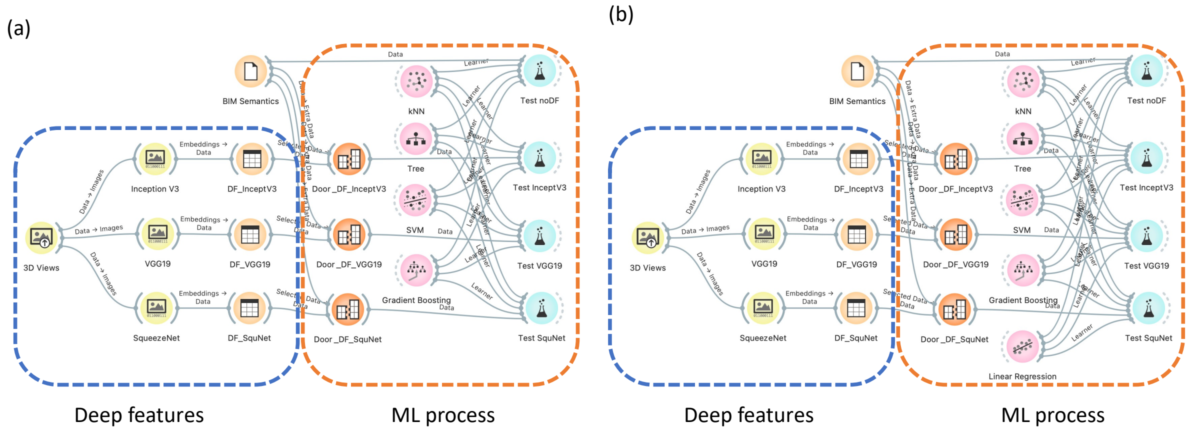

Figure 6. Orange program of deep features extraction and ML for predicting hinge details. (a) Classification; (b) Regression

Once the ML model is trained, it can be used to predict the hinge-related information, as shown in Table 3. The predicted data can be stored in a CSV file and used for the parametric design task. Figure 7 illustrates a Dynamo script retrieving the predicted data stored in a CSV file to perform the parametric design task for auto-enhancing the BIM object.

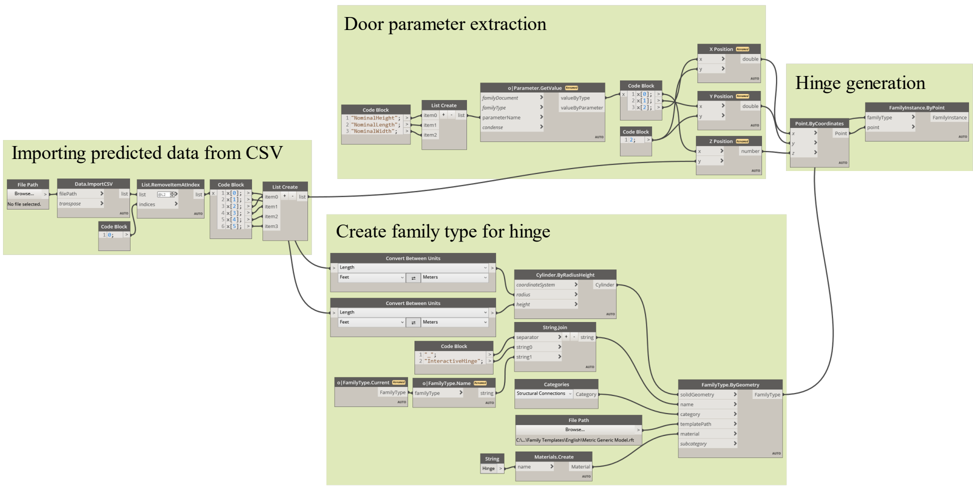

Figure 7. Dynamo script of adding and detailing new hinge instances to the door families

4.3 Results

4.3.1 Results of Step 1

The results of Step 1, as shown in Figure 2, contained 86 view image files (.PNG format) and an index file (.CSV format). The naming system of the PNG files is ‘Document name_family type – 3D view – 3DExterior’, as shown in Figure 8. The resolution of each PNG file was 1024 × 1024 pixels, and the average file size was around 10KB, while the disk size of the index file was about 20 KB. All the 3D views were shot from outside of the doors, as shown in Figure 8; therefore, the hinges – three or four instances – were invisible in the 3D view photos. Plus, the variety of the designs of the test door families, that is, all the three-hinge and four-hinge doors with hinges at NBS had well covered the common door types in real projects.

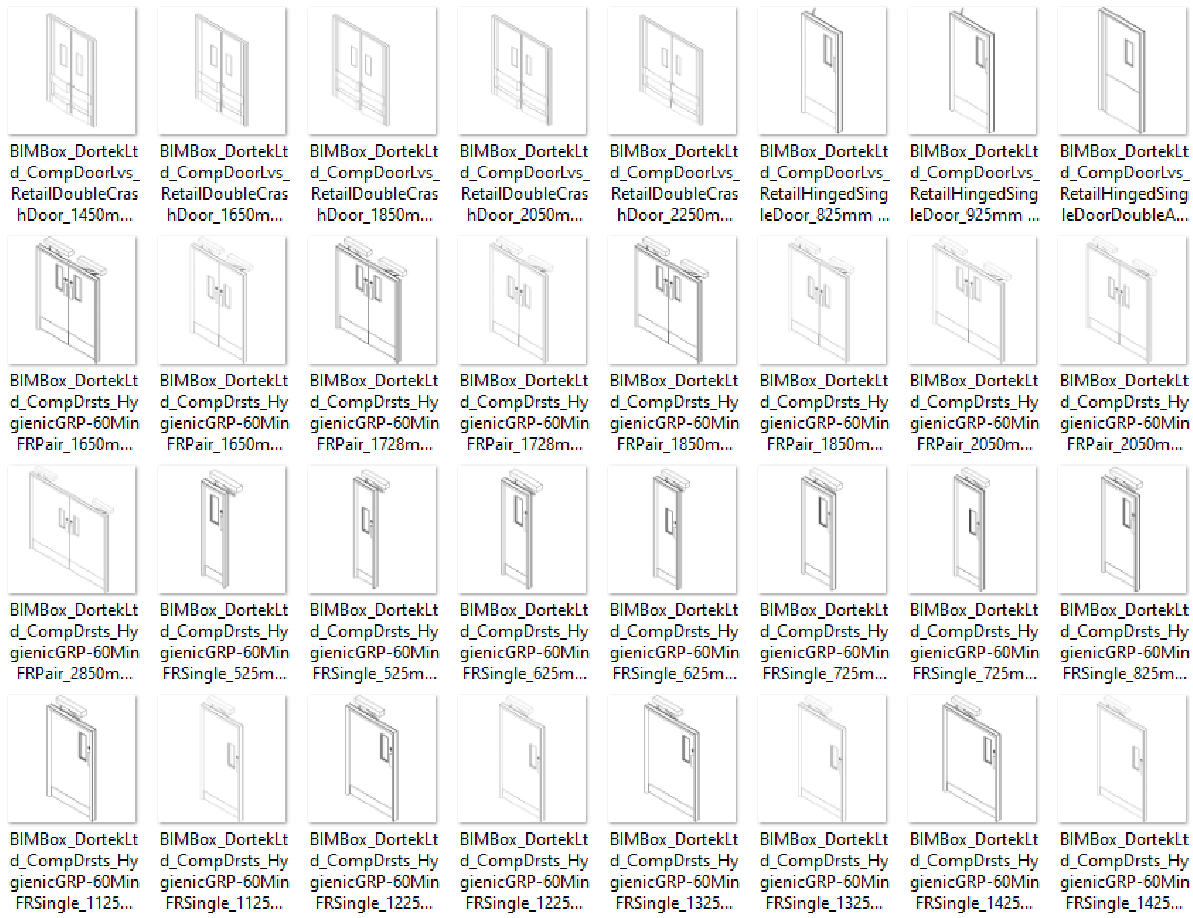

Figure 8. Screenshots of sample extracted 3D view images

Table 7 lists an excerpt of the decision parameters and target values extracted by the Dynamo script automatically. In general, there were three main categories: NBS standard parameters, discrete hinge details to classify, and continuous hinge detail values to regress. An example of discrete hinge detail was the radius in millimeters, where the four possible values were 3.5, 6, 7.5, and 8. For the continuous hinge details to regress, the values were normalized between 0 and 1, for example, as the relative Z-position against the NominalHeight.

Table 7. Excerpt of the training data for door features and prediction results

NBS standard parameters			Classification results			Regression			
Nominal Height	Nominal Length	Nominal Width	Quantity	Radius in mm	Depth in mm	Top	Bottom	Extra1	Extra2
2079	96	1125	3	3.5	100	0.0981	0.8389	0.4685	N/A
2479	96	1125	3	3.5	100	0.0823	0.8649	0.5543	N/A
2079	96	1925	4	3.5	100	0.0981	0.8389	0.4685	0.7186
2479	96	1925	4	3.5	100	0.0823	0.8649	0.5543	0.7640
…	…	…	…	…	…	…	…	…	…
2110	55	1004	4	8	100	0.1123	0.8517	0.3967	0.6242

4.3.2 Results of Step 2

Table 8 summarizes the results of utilizing DFs in the hinge prediction process. The uniform check marks for the three classification tasks show that DFs 100% improved the BIM details predictions in terms of F_1 scores against every baseline without DF (w/o DF). In the four regression tasks, 17 out of 20 DF-ML combinations showed accuracy improvement (indicated as check marks) in terms of reduced RMSE values. Thus, it was confident to conclude that DFs can improve the prediction of the hidden BIM details.

Table 8. Deep feature's improvement matrix under different ML algorithm

ML algorithm	Classification prediction tasks			Regression prediction tasks				Sub total
	Quantity	Depth	Radius	Top	Bottom	$Extra_1$	$Extra_2$	
KNN	✓	✓	✓	✓	✓	✓	✓	7
Tree	✓	✓	✓	×	✓	✓	×	5
XGBoost	✓	✓	✓	×	✓	✓	✓	6
LR	N/A	N/A	N/A	✓	✓	✓	✓	4
SVM	✓	✓	✓	✓	✓	✓	✓	7
Sub total	4	4	4	3	5	5	4	

"✓" indicates the result is improved compared to w/o DF; "N/A" indicates not applicable

The rightmost column in Table 8 shows that three ML algorithms (KNN, LR, and SVM) received consistent prediction improvements in all the possible tasks. In contrast, decision tree and XGBoost show encouraging – though not as consistent as others – results in receiving improvements in five and six out of the seven tasks. Based on our observation and experience during the experiments, the main reason would be the quantity (thousands of columns) of DFs sometimes exceeded the capacity of logical conditioning/branching combinations for rule/tree-based ML algorithm, leading to convergence at "local optima."

Table 9 shows the average and the best performances of classification tasks in terms of F_1 values, while the detailed result data is available in the data set in the appendix. In general, the adoption of DFs can increase the F_1 values considerably, by 15.4~58.7% on average. For the classification tasks, ML models with DFs have considerably higher best and mean F_1 scores than those w/o DF. And when comparing the mean F_1 score from all ML models using different CNNs to the mean score w/o DF, the error reduction can be calculated. In the classification tasks, all the values of error reduction are positive, especially for the depth of the hinge prediction task, for which the error reduction can reach 58.7%.

Table 9. Deep feature performance matrix of the classification tasks

Tasks	w/o DF		Performance of four ML algorithms using three DFs					Best setting
	Best F_1	Avg F_1	Best F_1	%*Imp**	Avg F_1	%*Imp**	*Stdev*#	
Quantity	0.8252	0.7346	0.8951	40.0	0.8152	30.4	0.0581	InceptV3+SVM
Depth	0.8963	0.8024	0.9656	66.9	0.9185	58.7	0.0356	VGG19+SVM
Radius	0.8983	0.8168	0.9654	66.0	0.9139	15.4	0.0389	VGG19+SVM
			Average	57.6		47.4		

* %*Imp*: Improvement in percentage calculated as error reduction in percentage, where *Error* = 1 − F_1

\# Stdev: Standard deviation of the sixteen F_1 scores from the combinations of all ML models and DFs

Table 10 shows the average and the best performances of regression tasks in terms of F_1 values. For the regression tasks, the adoption of DFs reduced the RMSE values of most ML models. However, an exception was the "Top" hinge locational prediction task, where the best RMSE w/o DF (predicted by XGBoost) was better. The main reason should lie in the limited capability of XGBoost in processing thousands of columns. Other models still have a better performance, especially for the bottom hinge locational prediction task (see "Bottom" in Table 10); the error reduction can reach 33.6% on average.

Table 10. Deep feature performance matrix of the regression tasks

Tasks	w/o DF		Performance of Five ML algorithms using three DFs					Best setting
	Best RMSE	Avg RMSE	Best RMSE	%*Imp*#	Avg. RMSE	%*Imp*#	*Stdev*^	
Top	0.0074	0.0213	0.0081	-10.2	0.0191	10.1	0.0191	InceptV3+XGBoost
Bottom	0.0093	0.0168	0.0079	15.7	0.0112	33.6	0.0112	VGG19+LR
Extra1	0.0623	0.0752	0.0388	37.7	0.0585	22.2	0.0585	InceptV3+LR
Extra2	0.0374	0.0416	0.0280	25.2	0.0376	9.5	0.0376	SquNet+SVM
			Average	17.1		18.8		

\# %*Imp*: Improvement in percentage calculated as error reduction in percentage, where *Error* = RMSE

^ *Stdev*: Standard deviation of the twenty RMSE scores from the combinations of all ML models and DFs

In addition to the performance, the time cost of DFs adoption was also considered and evaluated. In this experiment, there are mainly two parts of the time cost, DFs preparation and ML training time. For the preparation of DFs, Dynamo was used to export the 3D views of the BIM doors. The development of the Dynamo script would take around one hour for experienced developers or two to three hours for those not familiar with Dynamo. After exporting the images, Orange will be used for the DFs generation and ML training. The time performances of each DF are listed in Table 11. When comparing the effectiveness of the ML performance enhancement and the time consumption of the training process, SqueezeNet would be the most cost-effective CNN.

Table 11. Time performance in seconds for adopting different DFs on a laptop computer

DF	DF generation (s)	Avg. classification training (s)	Avg. regression training (s)
InceptV3	2.88	9.92	24.36
SqueezeNet	2.84	5.76	13.30
VGG19	2.94	10.72	31.18

4.3.3 Results of Step 3

Figure 9 shows the results of Step 3 for two test BIM doors for illustration. In each subfigure, the left part is the door before automatic hinge detailing, while the right half shows the final results of the proposed method. It can be seen that the newly-added hinges in red color were successfully added to the original BIM door family through the presented DF-facilitated automatic BIM detailing process.

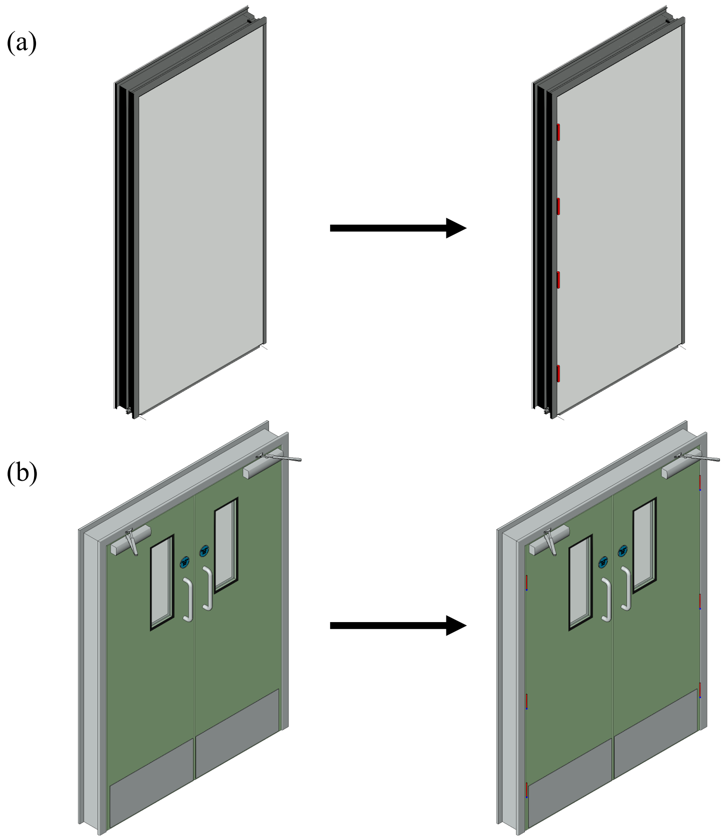

Figure 9. The automatic BIM detailing results for hinges. (a) a single door; (b) a double door

Figure 10 compares the detailing results between different methods, including ground truth data in black, predicting data from best ML settings with DFs (in red) and best ML settings without DFs (in blue). In Figure 10, the DFs here did not change the quantity of the hinges, while the differences were in the regression tasks of the Z-positions of the hinges. Since the figures were token in the elevation view, the correct parts of the hinges were overlapped by the ground truth. As a result, the red areas represent the errors from BIM detailing with DFs, while the blue areas represent the errors from BIM detailing without DF. In both cases, the considerably smaller red areas indicate that deep features led to more accurate BIM detailing of the hinges.

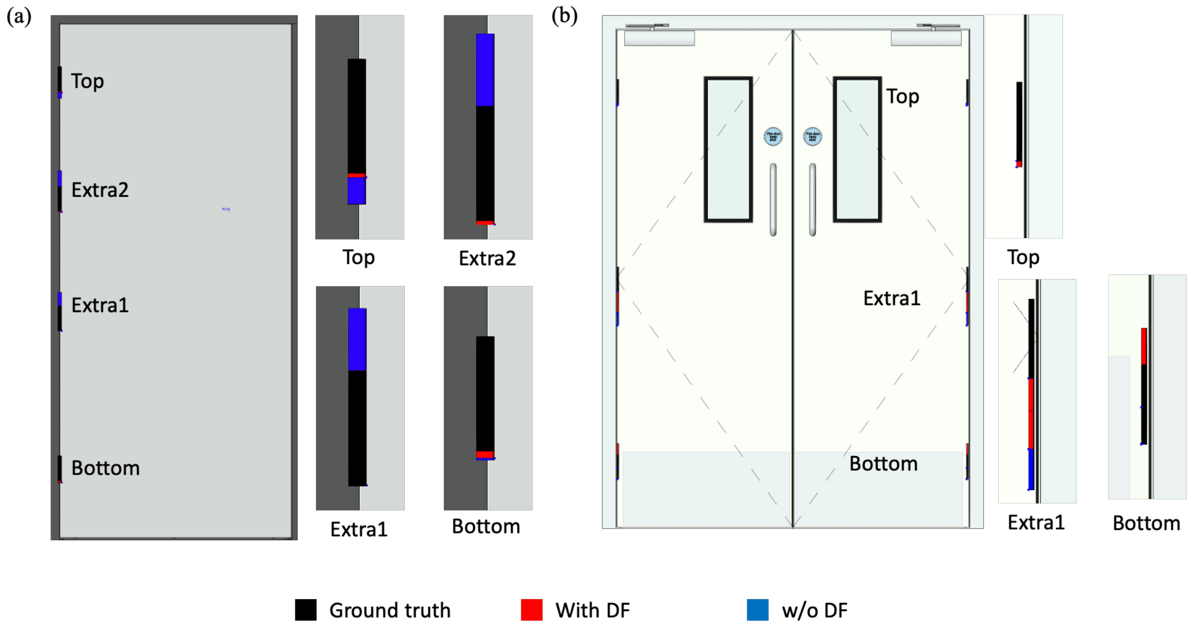

Figure 10. Errors of hinges' Z-positions by different ML methods. (a) a single door; (b) a double door

4.4 A behavior simulation of the detailed doors as postprocessing

Figure 11 illustrates how the newly-detailed hinges can create more realistic scenes for BIM simulations and visualizations. The BIM doors with newly-detailed hinges were exported from Revit (as FBX files) to Blender through an add-on. Physics engine-based interactions and simulations can then be created by assigning the hinges as the rigid body constraint. When external forces are applied to the door in a BIM, hinges can prevent doors from dropping out of the frame and can swing as expected, as shown in Figure 11. In comparison with Figure 1, with the newly detailed motion-bearing components inside BIM, the behavior simulations became more realistic, and the simulation results could provide the opportunity to open the door in the virtual environment (e.g., for digital twins).

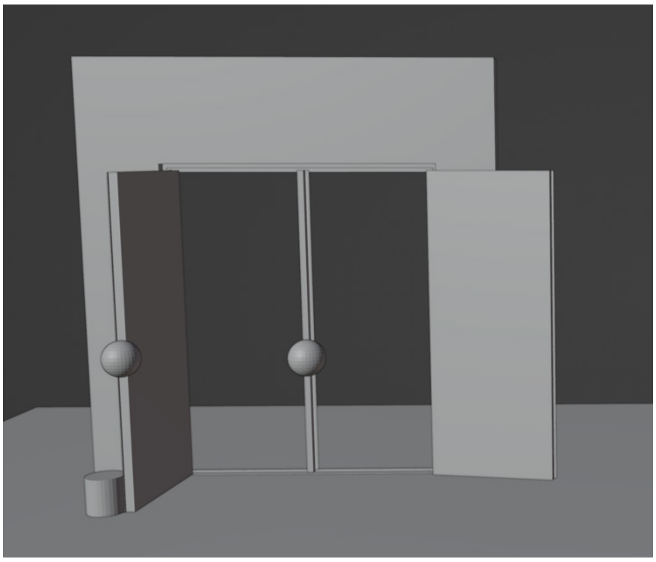

Figure 11. Door simulation with detailed motion-bearing components using Blender

5 Discussion

The method presented in this paper provides a novel workflow of the automatic BIM detailing process, including extracting existing BIM data, adopting DFs and ML models for predicting missing BIM components, and using parametric design to detail the missing components automatically. The proposed method has three advantages. First, compared with conventional manual BIM detailing, the method shortens the time of BIM detailing, especially for complex BIM projects and hidden details, through up-to-date DFs, ML algorithms, and parametric designs. DFs are proven successful for automatic BIM detailing (i.e., an average of 47.4% error reduction in classification tasks and 18.8% error reduction in regression tasks). The main reason should be the BIM design features, i.e., the door design features like glass panels, kick plates, and handles in Figure 8, that are represented as DFs (Han et al. 2019). These DFs can be recognized and analyzed by different ML algorithms, and eventually improving the automatic BIM detailing process. Secondly, the general methods are applicable to all BIM objects. Since DFs can capture the design features of the objects, which supplement more information than typical parameters, such as Width, Height, and Depth, to any ML algorithms, the significance of this study is not only applied to motion-bearing components of the door but also other BIM components with design features in 3D views. Lastly, the predicted parametric component details are concise and compatible with BIM versioning such as on blockchains (Xue & Lu 2020; Zhao et al. 2023). The effects of deep features on BIM detailing also shed light on new directions of BIM semantic enrichment and digital twin building (Xue et al. 2021).

However, there are two main limitations in this paper, i.e., the quantity of test BIM objects and the diversity of DFs. Although all the BIM doors annotated with nominal sizes were downloaded from NBS BIM Library for the tests (see Table 5), the diversity of test BIM objects was still limited. Future researchers are encouraged to test more BIM objects.

Furthermore, all three DFs in this paper were CNN DFs. Therefore, another research direction is to examine the effectiveness of more DL/DFs, such as deep transfer learning or transformer, for automatic BIM detailing processes.

6 Conclusion

The insufficient LOD hinders BIM applications for dynamic and interactive digital twins. This paper presents an automatic BIM detailing method based on the adoption of DFs. Experiments on 86 BIM doors from the NBS BIM Library were conducted to test the proposed methodology. According to the experimental results, the proposed automatic BIM detailing can achieve a satisfactory performance for all classification regression tasks with the facilitation of DFs. The results showed that DFs contributed an average of 47.4% (up to 58.7%) of error reduction in the classification tasks and an average of 18.8% (up to 33.6%) in the regression tasks. According to the performance and time cost in the experimental results, SqueezeNet was the most cost-effective deep feature extraction architecture for the proposed method.

This paper contributes to the research in two ways. From the industrial practitioner's perspective, the presented method demonstrates an automatic approach to BIM detailing. BIM modelers can thus focus on the general designs, while the detailed BIM components – even as invisible as hinges in the study case – can be created automatically. From BIM researchers' perspectives, the DFs of BIM 3D views and ML algorithms open new avenues for BIM semantic enrichment and digital twin building. With a higher LOD of BIM, digital twin constructions become more realistic and applicable for a variety of analyses.

Several research directions can extend this study. First, industrial specifications and standards and manufacturing semantics can be incorporated to reinforce the BIM detailing process. Furthermore, the non-geometry semantics in BIM can be further investigated if the semantic properties, such as the weight or material proprieties, are consistently available. Also, the applications of high LOD BIM objects and projects can be explored, such as interactive AECO educational games and digital twins.

Acknowledgment

The work present in this paper was financially supported by Hong Kong Research Grants Council (RGC) (Nos. 27200520, T22-504/21-R) and in part by The University of Hong Kong (No. 202111159233).

References

Alhindi, T. J., Kalra, S., Ng, K. H., Afrin, A. & Tizhoosh, H. R. (2018). Comparing LBP, HOG and deep features for classification of histopathology images. In *Proceedings of* *the 2018 International Joint Conference on Neural Networks.* IEEE. doi:10.1109/IJCNN.2018.8489329

Banihashemi, S., Khalili, S., Sheikhkhoshkar, M. & Fazeli, A. (2022). Machine learning-integrated 5D BIM informatics: Building materials costs data classification and prototype development. *Innovative Infrastructure Solutions, 7*(3), 215. doi:10.1007/s41062-022-00822-y

Barlish, K. & Sullivan, K. (2012). How to measure the benefits of BIM — A case study approach. *Automation in Construction, 24*, pp. 149-159. doi:10.1016/j.autcon.2012.02.008

Bianconi, F., Filippucci, M. & Buffi, A. (2019). Automated design and modeling for mass-customized housing. A web-based design space catalog for timber structures. *Automation in Construction, 103*, pp. 13-25. doi:10.1016/j.autcon.2019.03.002

Boje, Y., Guerriero, A., Kubicki, S. & Rezgui, Y. (2020). Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction, 114*, 103179. doi:10.1016/j.autcon.2020.103179

Borrmann, A., König, M., Koch, C. & Beetz, J. (2018). *Building Information Modeling.* Springer, ISBN 9783319928623. doi:10.1007/978-3-319-92862-3

Chan, D. W., Olawumi, T. O. & Ho, A. M. (2019). Perceived benefits of and barriers to Building Information Modelling (BIM) implementation in construction: The case of Hong Kong. *Journal of Building Engineering, 25*, 100764. doi:10.1016/j.jobe.2019.100764

Chen, J., Li, S., Liu, D. & Lu, W. (2022). Indoor camera pose estimation via style-transfer 3D models. *Computer-aided Civil and Infrastructure Engineering, 37*(3), pp. 335-353. doi:10.1111/mice.12714

Chen, K., Lu, W., Peng, Y., Rowlinson, S. & Huang, G. Q. (2015). Bridging BIM and building: From a literature review to an integrated conceptual framework. *International Journal of Project Management, 33*(6), pp. 1405-1416. doi:10.1016/j.ijproman.2015.03.006

Chen, T. & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). New York, NY, US: ACM. doi:10.1145/2939672.2939785

CIC. (2020). *CIC BIM Standards - General (Version 2).* Hong Kong, China: Construction Industry Council. Retrieved June 21, 2022, from https://www.bim.cic.hk/en/resources/publications_detail/85

CIC. (2021). *Construction Digitalisation Roadmap for Hong Kong.* Hong Kong, China: Construction Industry Council. Retrieved January 25, 2023, from https://www.bim.cic.hk/en/resources/publications_detail/95

Dargan, S., Kumar, M., Ayyagari, M. R. & Kumar, G. (2020). A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering, 27*(4), pp. 1071-1092. doi:10.1007/s11831-019-09344-w

Dong, N., Zhao, L., Wu, C.-H. & Chang, J.-F. (2020). Inception v3 based cervical cell classification combined with artificially extracted features. *Applied Soft Computing, 93*, 106311. doi:10.1016/j.asoc.2020.106311

Fai, S. & Rafeiro, J. (2014). Establishing an appropriate level of detail (LoD) for a building information model (BIM) – West Block, Parliament Hill, Ottawa, Canada. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, II-5*, pp. 123-130. doi:10.5194/isprsannals-II-5-123-2014

Ghaffarianhoseini, A., Tookey, J., Ghaffarianhoseini, A., Naismith, N., Azhar, S., Efimova, O. & Raahemifar, K. (2017). Building Information Modelling (BIM) uptake: Clear benefits, understanding its implementation, risks and challenges. *Renewable and Sustainable Energy Reviews, 75*, pp. 1046-1053. doi:10.1016/j.rser.2016.11.083

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009). The WEKA data mining software: An update. *Association for Computing Machinery Special Interest Group on Knowledge Discovery in Data Explorations Newsletter, 11*(1), pp. 10-18. doi:10.1145/1656274.1656278

Han, J., Kamber, M. & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Cambridge, MA, US: Morgan Kaufmann, ISBN 978-0-12-381479-1. doi:10.1016/C2009-0-61819-5

Han, X.-F., Laga, H. & Bennamoun, M. (2019). Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(5), pp. 1578-1604. doi:10.1109/TPAMI.2019.2954885

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications, 13*(4), pp. 18-28. doi:10.1109/5254.708428

Heaton, J., Parlikad, A. K. & Schooling, J. (2019). Design and development of BIM models to support operations and maintenance. *Computers in Industry, 111*, pp. 172-186. doi:10.1016/j.compind.2019.08.001

Hong, Y., Hammad, A. & Akbarnezhad, A. (2019). Forecasting the net costs to organisations of Building Information Modelling (BIM) implementation at different levels of development (LOD). *Journal of Information Technology in Construction, 24*, pp. 588-603. doi:10.36680/j.itcon.2019.033

Hu, Y. & Castro-Lacouture, D. (2019). Clash Relevance Prediction Based on Machine Learning. *Journal of Computing in Civil Engineering, 33*(2), 4018060. doi:10.1061/(ASCE)CP.1943-5487.0000810

ISO. (2018). *ISO 19650-1:2018 - Organization and Digitization of Information about Buildings and Civil Engineering Works, including Building Information Modelling (BIM) — Information Management using Building Information Modelling — Part 1: Concepts and Principles.* Geneva, Switzerland: ISO, ISBN 9789267110127. Retrieved January 25, 2023, from https://www.iso.org/standard/68078.html

Kensek, K. (2015). Visual programming for building information modeling: Energy and shading analysis case studies. *Journal of Green Building, 10*(4), pp. 28-43. doi:10.3992/jgb.10.4.28

Kim, J. B., Jeong, W., Clayton, M. J., Haberl, J. S. & Yan, W. (2015). Developing a physical BIM library for building thermal energy simulation. *Automation in Construction, 50*, pp. 16-28. doi:10.1016/j.autcon.2014.10.011

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), pp. 436-444. doi:10.1038/nature14539

Lee, H. J., Ullah, I., Wan, W., Gao, Y. & Fang, Z. (2019). Real-time vehicle make and model recognition with the residual SqueezeNet architecture. *Sensors, 19*(5), 982. doi:10.3390/s19050982

Lomio, F., Farinha, R., Laasonen, M. & Huttunen, H. (2018). Classification of Building Information Model (BIM) structures with deep learning. In *Proceedings of the 2018 7th European Workshop on Visual Information Processing.* IEEE. doi:10.1109/EUVIP.2018.8611701

Lou, J., Lu, W. & Xue, F. (2020). A review of BIM data exchange method in BIM collaboration. In *Proceedings of the 25th International Symposium on Advancement of Construction Management and Real Estate* (pp. 1329-1338). Springer Nature. doi:10.1007/978-981-16-3587-8_90

Lu, Q., Parlikad, A. K., Woodall, P., Don Ranasinghe, G., Xie, X., Liang, Z., Konstantinou, E., Heaton, J. & Schooling, J. (2020). Developing a digital twin at building and city levels: Case study of West Cambridge campus. *Journal of Management in Engineering, 36*(3), 5020004. doi:10.1061/(ASCE)ME.1943-5479.0000763

Natanian, J., Aleksandrowicz, O. & Auer, T. (2019). A parametric approach to optimizing urban form, energy balance and environmental quality: The case of Mediterranean districts. *Applied Energy, 254*, 113637. doi:10.1016/j.apenergy.2019.113637

NBS. (2022). *NBS National BIM Library.* Newcastle, UK: National Building Specification. Retrieved July 7, 2022, from https://www.nationalbimlibrary.com

NIBS. (2015). *National BIM Standard - United States Version 3.* Washington, DC, US: National Institute of Building Sciences. Retrieved June 10, 2022, from https://www.nationalbimstandard.org/files/NBIMS-US_V3_3_Terms_and_Definitions.pdf

NIC. (2017). *Data for the Public Good.* London, UK: National Infrastructure Commission. Retrieved July 16, 2022, from https://nic.org.uk/app/uploads/Data-for-the-Public-Good-NIC-Report.pdf

Opoku, D.-G. J., Perera, S., Osei-Kyei, R. & Rashidi, M. (2021). Digital twin application in the construction industry: A literature review. *Journal of Building Engineering, 40*, 102726. doi:10.1016/j.jobe.2021.102726

Potseluyko, L., Pour Rahimian, F., Dawood, N., Elghaish, F. & Hajirasouli, A. (2022). Game-like interactive environment using BIM-based virtual reality for the timber frame self-build housing sector. *Automation in Construction, 142*, 104496. doi:10.1016/j.autcon.2022.104496

Sadeghi, M., Elliott, J. W., Porro, N. & Strong, K. (2019). Developing building information models (BIM) for building handover, operation and maintenance. *Journal of Facilities Management, 17*(3), pp. 301-316. doi:10.1108/JFM-04-2018-0029

Seaton, H., Cristina, S., Sepasgozar, S. M. & Sawhney, A. (2022). *Digital Twins from Design to Handover of Constructed Assets.* London, UK: Royal Institution of Chartered Surveyors, ISBN 9781783214648. Retrieved Jul 18, 2022, from https://www.rics.org/globalassets/wbef-website/reports-and-research/digital-twins-from-design-to-handover-of-constructed-assets.pdf

Song, J., Kim, J. & Lee, J. K. (2018). NLP and deep learning-based analysis of building regulations to support automated rule checking system. In *Proceedings of the 35th International Symposium on Automation and Robotics in Construction.* IAARC. doi:10.22260/ISARC2018/0080

Su, T., Li, H. & An, Y. (2021). A BIM and machine learning integration framework for automated property valuation. *Journal of Building Engineering, 44*, 102636. doi:10.1016/j.jobe.2021.102636

Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P. & Gao, X. (2019). A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Automation in Construction, 101*, pp. 127-139. doi:10.1016/j.autcon.2019.01.020

Thabet, W., Lucas, J. & Srinivasan, S. (2022). Linking life cycle BIM data to a facility management system using Revit Dynamo. *Organization, Technology & Management in Construction, 14*(1), pp. 2539-2558. doi:10.2478/otmcj-2022-0001

Torres-Calderon, W., Chi, Y., Amer, F. & Golparvar-Fard, M. (2019). Automated mining of construction schedules for easy and quick assembly of 4D BIM simulations. In *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation* (pp. 432-438). ASCE. doi:10.1061/9780784482421.055

Valero, E., Forster, A., Bosché, F., Renier, C., Hyslop, E. & Wilson, L. (2018). High level-of-detail BIM and machine learning for automated masonry wall defect surveying. In *Proceedings of the 35th International Symposium on Automation and Robotics in Construction.* IAARC. doi:10.22260/ISARC2018/0101

Wang, H., Pan, Y. & Luo, X. (2019). Integration of BIM and GIS in sustainable built environment: A review and bibliometric analysis. *Automation in Construction, 103*, pp. 41-52. doi:10.1016/j.autcon.2019.03.005

Weng, Y., Mohamed, N. A., Lee, B. J., Gan, N. J., Li, M., Jen Tan, M., Li, H. & Qian, S. (2021). Extracting BIM information for lattice toolpath planning in digital concrete printing with developed dynamo script: A case study. *Journal of Computing in Civil Engineering, 35*(3), 5021001. doi:10.1061/(ASCE)CP.1943-5487.0000964

Wu, Y., Shang, J. & Xue, F. (2021). RegARD: Symmetry-based coarse registration of smartphone's colorful point clouds with CAD drawings for low-cost digital twin buildings. *Remote Sensing, 13*(10), 1882. doi:10.3390/rs13101882

Xue, F. & Lu, W. (2020). A semantic differential transaction approach to minimizing information redundancy for BIM and blockchain integration. *Automation in Construction, 118*, 103270. doi:10.1016/j.autcon.2020.103270

Xue, F. (2022). As-built building information modeling: joint effort of 3D reconstruction and semantic enrichment. In *Research Companion to Building Information Modeling* (pp. 190-211). Northampton, MA, USA: Edward Elgar Publishing. doi:10.4337/9781839105524.00019

Xue, F., Lu, W., Webster, C. J. & Chen, K. (2019). A derivative-free optimization-based approach for detecting architectural symmetries from 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing, 148*, pp. 32-40. doi:10.1016/j.isprsjprs.2018.12.005

Xue, F., Wu, L. & Lu, W. (2021). Semantic enrichment of building and city information models: A ten-year review. *Advanced Engineering Informatics, 47*, 101245. doi:10.1016/j.aei.2020.101245

Yang, L., Cheng, J. C. & Wang, Q. (2020). Semi-automated generation of parametric BIM for steel structures based on terrestrial laser scanning data. *Automation in Construction, 112*, 103037. doi:10.1016/j.autcon.2019.103037

Yang, Z., Xue, F. & Lu, W. (2021). Handling missing data for construction waste management: machine learning based on aggregated waste generation behaviors. *Resources, Conservation and Recycling, 175*, 105809. doi:10.1016/j.resconrec.2021.105809

Zabin, A., González, V. A., Zou, Y. & Amor, R. (2022). Applications of machine learning to BIM: A systematic literature review. *Advanced Engineering Informatics, 51*, 101474. doi:10.1016/j.aei.2021.101474

Zhang, M., Ke, J., Tong, L. & Luo, X. (2021). Investigating the influence of route turning angle on compliance behaviors and evacuation performance in a virtual-reality-based experiment. *Advanced Engineering Informatics, 48*, 101259. doi:10.1016/j.aei.2021.101259

Zhang, M.-L. & Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition, 40*(7), pp. 2038-2048. doi:10.1016/j.patcog.2006.12.019

Zhao, R., Chen, Z. & Xue, F. (2023). A blockchain 3.0 paradigm for digital twins in construction project management. *Automation in Construction, 145*, 104645. doi:10.1016/j.autcon.2022.104645