Enhancing Deep Learning-based BIM Element Classification via Data Augmentation and Semantic Segmentation

Y. Yu[a], K. Lee[a], D. Ha[a] and B. Koo[a*]

[a]Department of Civil Engineering, Seoul National University of Science and Technology, Republic of Korea
E-mail: youngsu@seoultech.ac.kr, leekoeun@seoultech.ac.kr, daemok@seoultech.ac.kr, bonsang@seoultech.ac.kr

Abstract –

A critical aspect of BIM is the capability to embody semantic information about its element constituents. To be interoperable, such information needs to conform to the Industry Foundation Classes (IFC) standards and protocols. Artificial intelligence approaches have been explored as a way to verify the semantic integrity of BIM to IFC mappings by learning the geometric features of individual BIM elements. The authors through previous studies also investigated the use of geometric deep learning to automatically classify individual BIM element classes. However, such efforts were limited in the number of training data and restricted to subtypes of BIM elements. This study has significantly expanded the training set, to include a total of 46,746 elements representing 13 types of BIM elements. The magnitude of the data set is considered to be the first of this kind. Furthermore, Conditional Random Fields as Recurrent Neural Networks (CRF-RNN), a deep learning algorithm for semantic segmentation, was deployed to enhance the quality of individual input images. Deploying the dataset and segmentation improved the performance of previous model (Multi-View CNN) by 4.37% and achieve an overall performance of 95.38%.

Keywords –

BIM; IFC; Semantic Integrity; MVCNN; CRF-RNN

1 Introduction

Building Information Modeling (BIM) has become the mainstream medium for integrating and disseminating information during the project life cycle of construction projects. Multiple stakeholders today employ a variety of specialized BIM tools tailored to their various needs in the design, engineering, and construction management of their projects.

The Industrial Foundation Classes (IFC), a neutral and open data format, is a critical component in ensuring interoperability within the BIM centric work process. The absence of such a standard would require local data protocols for each and every pair of applications, quickly making BIM based project execution intractable.

Working under the IFC protocol requires BIM elements, relationships, and their properties to be represented in conformance to its standards. However, due to the lack of logical rigidity of the IFC schema, IFC model instances are prone to misrepresentations and misinterpretations, resulting in a lack of semantic integrity [4].

Such issues continue to be addressed in the domain of ‘semantic enrichment’, which reasons about the relations and meaning implicit in the geometry and topology of BIM models to check and rectify semantic inaccuracies.

In particular, a subset of these studies investigated ways to check the correct mapping of individual BIM elements to their corresponding IFC entities [1-3, 12, 16].

For example, [12] formalized sets of inference rules to check mappings, and subsequently automate inaccurate associations.

More recently, artificial intelligence approaches have been employed as an alternate approach to check the integrity of BIM-element to IFC-entity mappings. This approach has been conducted by extracting geometric features existing in the IFC or using 2D image or 3D shape information of each element for learning model training. [3] conducted a study to classify space in the BIM model using geometric features-based machine learning, and demonstrated that the classification accuracy of machine learning approach was superior to the inference rule-based approach. [11] used images extracted from BIM models to classify building types, and [6] classified BIM furnishing elements using 2D CNN.

The authors also explored the use of different machine learning and deep learning models to determine their applicability [7-9]. Mainly, we trained learning models based on the geometric features of individual BIM elements. In particular, we attained promising results from incorporating Multi-View CNN(MVCNN), a geometric deep learning model that learns from multiple panoramic images of a 3D artifact to learn and distinguish its shape [7].

However, despite its relative high performance, MVCNN still was limited in correctly classifying specific BIM elements despite their distinct geometric differences. The classification errors were attributed mainly to two factors. The first factor was the limited number of training data, as well as the imbalance in the number of samples per element class. Secondly, and more technically, was the lack of definition between the boundary of objects in the individual images, which made it difficult to detect the geometric detail of the elements.

Commensurately, novel approaches were newly explored to rectify these limitations. First, the data set for training was increased by applying data augmentation based on parametric modeling. Secondly, as a data preprocessing step prior to training MVCNN, Conditional Random Fields as RNN (CRF-RNN), a deep learning-based semantic segmentation technique, was applied to sharpen the geometric features of the images for each and every BIM element.

13 types of BIM elements with high utility in the architectural field were collected, and an expanded data set was constructed by performing data augmentation via parametric modeling. Afterward, the deep learning model was applied to the dataset to compare their performances. The first step was to learn the MVCNN algorithm, and the second step was to learn the MVCNN algorithm after applying semantic segmentation based on the CRF-RNN algorithm. Finally, by testing the two deep learning models on a BIM model excluded from the dataset, the classification performance of the developed model was compared to quantitatively verify the degree of performance improvement.

2 Research Background

2.1 Multi-View CNN (MVCNN)

With the recent development of computing technology, it is possible to directly utilize 3D data, and object recognition research using a 3D model-based CNN algorithm is increasing. However, 3D raw data is very large in size to be directly applied to deep learning algorithms, so it must be accompanied by a lightweight process to reduce it to a size suitable for learning [13, 14, 17]. Furthermore, the problem of performance degradation due to the loss of detailed features of objects in the lightweight process also makes it difficult to directly use the 3D CNN model.

MVCNN, designed specifically to resolve this issue, has been proven to provide better performance than directly utilizing existing 3D data. MVCNN utilizes the multi-angled images of a 3D model, while using multiple CNN layers which in effect prevents loss of geometric details [15].

Figure 1 below shows the architecture of MVCNN. A 3D object is rendered as 12 images taken panoramically. Each image is then input to individual Convolutional Neural Networks (CNN_1), which extracts compact shape descriptors representing the characteristics of each image. All extracted shape descriptors are reduced to one shape descriptor in the view-pooling layer, and they are transmitted to CNN (CNN_2) for final classification through the softmax classifier. Each CNN used in MVCNN utilized a VGG-M network composed of five convolution layers (CNN_1), two fully connected layers and, a softmax classification layer (CNN_2).

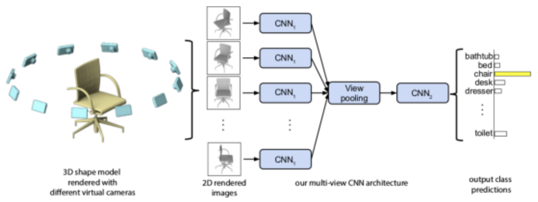


Figure 1. Multi-view CNN for 3D Shape recognition [15]

2.2 Conditional Random Field as Recurrent Neural Networks (CRF-RNN)

2D image-based deep learning models such as MVCNN have a disadvantage that their performance is highly dependent on the resolution and sharpness level of the data. That is, to improve the classification performance of the learning model, it is necessary to collect high-quality image data for training or to improve the resolution and clarity of the previously collected images. Accordingly, this study aims to improve the classification performance of MVCNN models by applying semantic segmentation to images of individual elements collected.

In this study, the CRF-RNN algorithm was used as a method for semantic segmentation. CRF (Conditional Random Field) refers to an undirected probabilistic graph model used for pattern recognition and structural prediction by labeling and segmenting consecutive pixels in an image. The simple CRF is composed of a lattice form in which adjacent nodes (pixels) in an image are connected by an edge, and due to this, exquisite segmentation was impossible during image segmentation. Accordingly, a Fully Connected CRF methodology that enables exquisite image segmentation by connecting all pixels of an image in pairs has been proposed, but there is a limitation in that the computation time is very long. Afterward, a method was devised to reduce the time required for label inference to 0.2 seconds by simplifying the complex structure by applying mean field approximation to the fully connected CRF.

CRF-RNN is a method of reconstructing two models into one framework based on a Recurrent Neural Network (RNN) to utilize weights output via CRFs with mean-field approximation as parameters in CNN learning. Due to this, it is possible to reduce the computation time and secure high semantic segmentation accuracy, and for this reason, it was adopted as a methodology for semantic segmentation in this study.

The formula below is the operating structure of the CRF-RNN algorithm for one iteration of the mean field. In the equation below, T denotes mean-field iteration, Q_{in} and Q_{out} denote input and output according to respective one average field iteration, and Q_{final} denotes the final prediction result of CRF-RNN. $Softmax(U)$ is the output value of the CNN operation, U is the unary potential value, $f_\theta(U, Q_{in}, I)$ is the weight inferred by Q_{in}, I is the image, and θ is the parameter of the CRF.

$$Q_{in}(t) = \begin{cases} softmax(U), & t = 0 \\ Q_{out}(t-1), & 0 < t \leq T \end{cases} \tag{1}$$

$$Q_{out}(t) = f_\theta(U, Q_{in}, I), \quad 0 \leq t \leq T \tag{2}$$

$$Q_{final}(t) = \begin{cases} 0, & 0 \leq t < T \\ Q_{out}(t), & t = T \end{cases} \tag{3}$$

After integrating the structure into one deep neural network, end-to-end learning is possible by implementing a back propagation algorithm. Through this process, semantic segmentation was performed on panoramic 12 images for each element, and an example of the result is presented in figure 2.

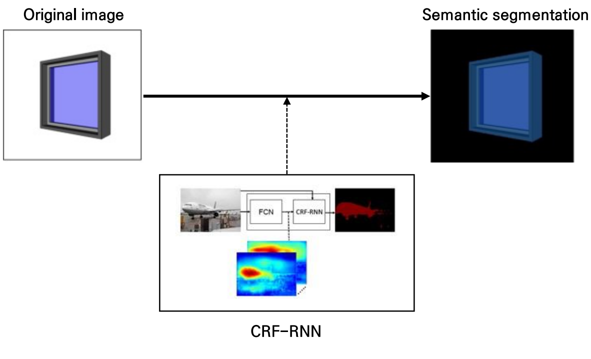


Figure 2. CRF-RNN-based semantic segmentation

3 Research Methodology

3.1 Parametric Modeling-based Data Augmentation

A goal of this study was to build a sufficiently large BIM element data set to train MVCNN. Ideally, BIM element samples need to be collected from open-source libraries or existing BIM models. However, such avenues were limited due to copyright issues or lack of high-quality data from attainable BIM models.

Thus, a data augmentation process that creates new data based on existing data was required. In this study, parametric modeling, an element technology of BIM was used.

Parametric modeling applies the concept of an independent parent/child to the composition factors (line, point, spline, plane, etc.) of individual BIM elements and connects them in a mutually related structure [10]. In other words, by defining the dimensions of the element as parameters and expressing their relationship as a function, the user can convert it into a shape suitable for the purpose through parameter setting. Since the operating principle, range, and limit of the result are directly affected by the modeling method in this process, it is necessary to clearly set the shape control criteria [5].

To establish the shape control criteria in this research, 13 types of major building elements were collected from four IFC standard building models and three online BIM libraries (KBIMS Library, NBS, bim object), and their shapes were investigated and analyzed. As a result, 45 parameters and their ranges were extracted from 13 elements, and the results are shown in Table 1. Subsequently, parametric modeling was performed using Revit Dynamo software based on the parameters. Figure 3 shows an example of parametric modeling of a beam element using this method.

Table 1. Parameters and range for each BIM element

Element	Parameter	Range(mm)
Beam	Width	250-600
	Height	400-1,100
	H (web width)	100-900
	B (flange width)	75-400
Column	Width	4-19
	Height	7-37
	H (web width)	200-1,000
	B (flange width)	200-1,200
	t1 (web thickness)	100-900
	t2 (flange thickness)	75-400
	Diameter	4-19
Double door	Width	7-37
	Height	500-1,000
	Frame width	1,800-2,400
	Panel thickness	1,800-2,400
	Opening width	40-60
Single door	Width	15-35
	Height	40-60
	Frame width	900-1,100
	Panel thickness	1,800-2,400
	Opening width	50-70
Slab	Width	20-40
	Length	90-110
Covering	Width	100-400
	Length	100-400
Wall	Length	3,000-11,000
Window	Width	800-1,800
	Height	900-1,600

	Frame width	80-120
	Panel thickness	10-30
	Frame thickness	30-70
	Glass thickness	2-22
Revolving door	Width	1900-2,500
	Height	2100-2,800
	Opening width	500-2,000
Curtain wall	Length	12000-1,7000
	Height	4000-6,000
	Vertical grid	12,100-16,900
	Horizontal grid	4,100-5,900
Stair flight, Member	Shape	U-shape with middle
		U-shape straight
		Straight with land
		Spiral
		L-shape winder
		U-shape winder
		Straight
	Type	General
		Reinforced concrete
		Wood
		Metal
		Steel
		Stainless steel
		Aluminum
	Width	1,000-3,000
	Radius	1,200-1,500
	Step	3-10
Railing	Type	Wood
		Metal
		Steel
		Stainless steel
		Glass
		Toughened glass

*Stairs include stair flight, member, and railing, so when parametric modeling is applied to stairs, these elements are applied at the same time.

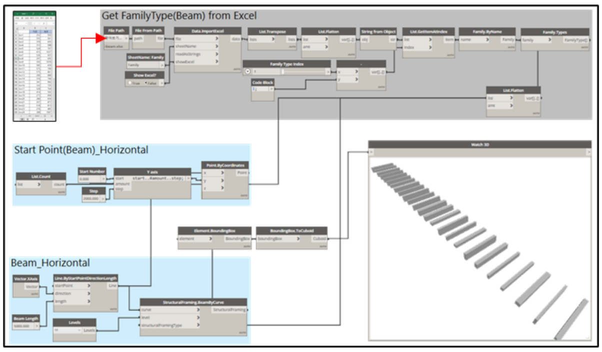


Figure 3. Parametric modeling using Revit Dynamo

The research team named the expanded data set built through this process 'ArchShapesNet', and released this data set in an open source form on the i3LAB homepage (http://i3l.seoultech.ac.kr) for other researchers to use.

3.2 ArchShapesNet Overview

12,672 elements of 13 types were collected from the four IFC standard architectural models (Table 2) and three BIM libraries mentioned in Section 3.2. Parametric modeling-based data augmentation was then performed to finally construct the 'ArchShapesNet' data set consisting of 46,746 elements. The data distribution for each element is presented in Table 3. After constructing an element classification model by applying a deep learning algorithm to the constructed ArchShapesNet data set, the performance was verified for 'Sejong city stadium', which was not used for learning.

Table 2. Five models for training and test

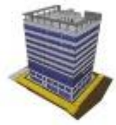
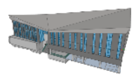
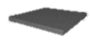
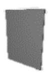
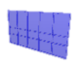
IFC model	Train/Test	Image
Office building	Train	
Cultural and assembly facilities		
Educational research facilities		
Single house		
Sejong city stadium	Test	

Table 3. Collected and augmented train data set status

Type	Beam		Column		Slab	
	Original	Augmented	Original	Augmented	Original	Augmented
No. of element	1,908	3,882	848	3,086	2,562	4,002
Rendering						
Type	Wall		Window		Stair flight	
	Original	Augmented	Original	Augmented	Original	Augmented
No. of element	3,731	4,917	781	3,353	140	5,407
Rendering						
Type	Member		Railing		Curtain Wall	
	Original	Augmented	Original	Augmented	Original	Augmented
No. of element	55	2,451	62	3,729	195	4,195
Rendering						
Type	Covering		Single door		Double door	
	Original	Augmented	Original	Augmented	Original	Augmented
No. of element	364	2,364	1,022	3,568	990	2,674

Rendering			
Type	Revolving Door		
	Original	Augmented	
No. of element	14	3,178	
Rendering			

3.3 Data Preparation for Deep Learning Implementation

Each element in ArchShapesNet were rendered into panoramic 12 images for MVCNN training. 'KBIM Assess-Lite', an automatic IFC model checking software, was used for this conversion process. The images consist of 10 side images taken panoramically at 36° intervals and 2 images taken from top and bottom.

Table 4 shows the data distribution for each element of the verification model (Sejong city stadium). In other words, model training is performed with the data constructed in Section 3.2, and the performance of the model is evaluated by testing the elements presented in this section. However, the verification model was composed of curtain walls without windows due to the characteristics of sports facilities, and the member and revolving door elements were not included, so verification of those elements were excluded.

Table 4. Status of test data set (Sejong city stadium)

Type	No. of element
Beam	164
Column	171
Slab	61
Wall	308
Window	0
Stair flight	19
Member	0
Railing	24
Curtain wall	63
Covering	12
Single door	19
Double door	4
Revolving door	0
Total	**845**

3.4 MVCNN Implementation

Figure 4 shows the MVCNN model construction process using the panoramic 12 image data for each element. When the 12 images of individual elements pass through the neural network (CNN_1), the features of the elements in the image are extracted. Afterward, the features of each extracted image are integrated in the view-pooling layer, which is again passed through the secondary recurrent neural network (CNN_2). Here, CNN_2 is composed of the softmax layer, and through this, the classification results for individual elements are output. The MVCNN implementation utilized python-based Tensorflow, and through this, a first step learning model (baseline) for 13 building elements was established.

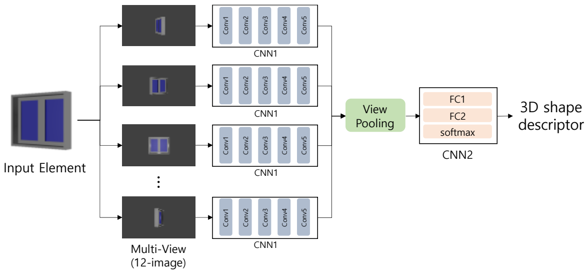


Figure 4. MVCNN architecture

3.5 CRF-RNN+MVCNN Implementation

Figure 5 shows the second step MVCNN model learning process using images with improved clarity by applying semantic segmentation. In this step, the MVCNN model was trained after applying the CRF-RNN algorithm so that the model can focus on the shape of the element by clearly dividing the region where the element exists. In this process, CRF-RNN was applied using python-based Keras, and a CRF-RNN+MVCNN model was built through the above-mentioned series of processes.

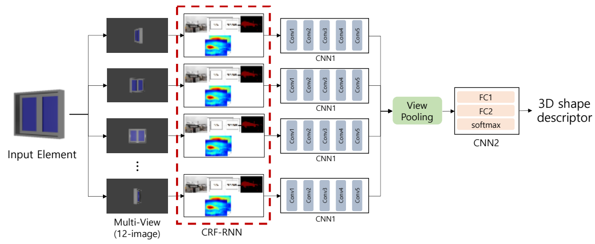


Figure 5. CRF-RNN + MVCNN architecture

4 Results

Table 5 and Figure 6 below show the results of the MVCNN model validation based on the verification data presented in section 3.3. As a result of verification, the MVCNN model was confirmed to recognize and classify elements in the actual BIM model with an accuracy of 91.01%. However, in the case of slab, covering, and stair flight elements, the classification accuracy were notably lower than that of other elements.

Table 5. Validation results (MVCNN)

Element	Precision	Recall	F1 score	Accuracy (%)
Beam	0.86	0.99	0.92	99.39
Column	1.00	0.99	0.99	98.83
Slab	0.77	0.33	0.46	32.79
Wall	0.99	0.94	0.96	93.51
Window	-	-	-	-
Stair flight	0.88	0.79	0.83	78.95
Member	-	-	-	-
Railing	1.00	0.92	0.96	91.67
Curtain wall	1.00	0.94	0.97	93.65
Covering	0.11	0.83	0.20	83.33
Single door	1.00	1.00	1.00	100.00
Double door	1.00	1.00	1.00	100.00
Revolving door	-	-	-	-
Total	**0.78**	**0.87**	**0.83**	**91.01**

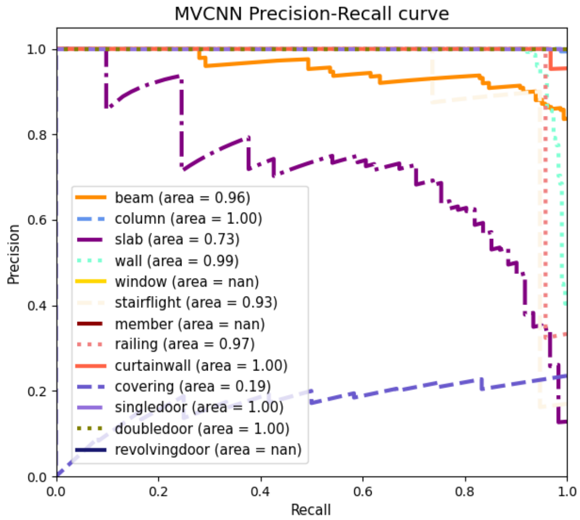


Figure 4. Precision-recall curve (MVCNN)

Table 6 and Figure 7 show the results of the CRF-RNN+MVCNN model verification. This hybrid model recognized and classified elements in the actual BIM model with high accuracy of 95.38% and achieved overall accuracy of over 90%. The model has difficulty for slab elements, with an accuracy of 68.85%. Yet, it still retained a higher classification accuracy compared to the standalone MVCNN model.

Table 6. Validation results (CRF-RNN + MVCNN)

Element	Precision	Recall	F1 score	Accuracy (%)
Beam	0.94	1.00	0.97	100.00
Column	0.99	0.99	0.99	99.42
Slab	0.88	0.69	0.77	68.85
Wall	1.00	0.96	0.98	96.10
Window	-	-	-	-
Stair flight	1.00	0.95	0.97	94.74
Member	-	-	-	-
Railing	1.00	1.00	1.00	100.00
Curtain wall	1.00	1.00	1.00	100.00
Covering	0.12	0.50	0.19	50.00
Single door	0.95	1.00	0.97	100.00
Double door	0.80	1.00	0.89	100.00
Revolving door	-	-	-	-
Total	**0.87**	**0.90**	**0.88**	**95.38**

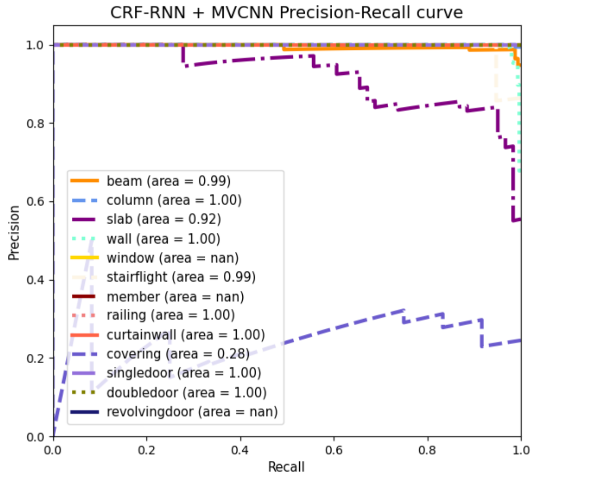


Figure 7. Precision-recall curve (CRF-RNN + MVCNN)

5 Discussion

5.1 Discussion of the Results

The ACC values of the previously presented MVCNN and CRF-RNN models were 91.01% and 95.38%, respectively, and the F_1 *scores* were 0.83 and 0.88, respectively. Through this, when CRF-RNN-based semantic segmentation was applied to the MVCNN learning process, the ACC improved by 4.37% and the F_1*score* improved by 0.05, so it was possible to confirm that applying semantic segmentation improved MVCNN's ability to classifying the BIM elements.

The 4.37% improvement is meaningful as the increase in accuracy resulted from specific elements that MVCNN by itself had trouble in distinguishing correctly. As shown in the confusion matrix (Table 7), the slab element, which was previously misclassified as either covering or beam, was classified properly, resulting in a significant ACC improvement. Moreover, the classification accuracy of elements with complex geometric features such as stair flight, railing, and curtain wall also increased. Thus, employing CRF-RNN to sharpen images was conducive to enhancing MVCNN's most relevant shortcomings.

Table 7. Confusion matrix of delta values between MVCNN and CRF-RNN+ MVCNN

Actual \ Predicted	Beam	Column	Slab	Wall	Window	Stair flight	Member	Railing	Curtain wall	Covering	Single door	Double door	Revolving door	Total
Beam	164(▲1)	0	0	0(▼1)	0	0	0	0	0	0	0	0	0	164
Column	0	170(▲1)	0(▼1)	0(▼1)	0	0	0	0	0	0	1(▲1)	0	0	171
Slab	0(▼7)	0	42(▲22)	0	0	0(▼1)	0	0	0	19(▼14)	0	0	0	61
Wall	10(▼9)	1(▲1)	0	296(▲8)	0	0(▼1)	0	0	0	0	0	1(▲1)	0	308
Window	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stairflight	0	0	1(▼3)	0	0	18(▲3)	0	0	0	0	0	0	0	19
Member	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Railing	0	0	0	0(▼1)	0	0	0	24(▲2)	0	0(▼1)	0	0	0	24
Curtainwall	0	0	0	0	0(▼4)	0	0	0	63(▲4)	0	0	0	0	63
Covering	1	0	5(▲4)	0	0	0	0	0	0	6(▼4)	0	0	0	12
Singledoor	0	0	0	0	0	0	0	0	0	0	19	0	0	19
Doubledoor	0	0	0	0	0	0	0	0	0	0	0	4	0	4
Revolvingdoor	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	175(▼15)	171(▲2)	48(▲22)	296(▲5)	0(▼4)	18(▲1)	0	24(▲2)	63(▲4)	25(▼19)	20(▲1)	5(▲1)	0	845

5.2 Limitations

Although applying CRF-RNN-based semantic segmentation improved the classification accuracy, for elements of similar shapes, namely slabs and covering elements, were still misclassified. This is because many the two elements are indistinguishable in their shape, apart from their thickness. So, it is difficult to distinguish them due to the characteristics of the MVCNN model, which considers only geometric shapes in the training process. When considering the two elements as a single type, the classification accuracy of the CRF-RNN+MVCNN model increases to 98.22%. The higher value indicates that our model performs is competent in distinguishing geometric features of individual elements. Nevertheless, from a practical point of view, it is necessary to recognize slab and covering as separate elements when verifying the semantic integrity of architectural BIM models, thus improving this limitation is planned by adding additional attribute variables or utilizing relational information in conjunction with the learning process in the future.

6 Conclusion

This study aimed to construct a BIM element classifier using a 3D geometric deep learning algorithm and improve its performance by increasing the definition of individual images using semantic segmentation.

ArchShapesNet, an expanded data set, was constructed by applying parametric modeling-based data augmentation so that learning for each element class was sufficiently performed, and CRF-RNN-based semantic segmentation was applied to properly reflect the geometric features of the BIM element in the learning process. As a result, the ACC of the MVCNN model applying CRF-RNN prior to the learning process was found to be high at 95.38%, which is a 4.37% improvement in ACC compared to the baseline MVCNN model. In detail, the classification accuracy of slab and elements with complex geometry was increased, and through this, it was confirmed that the application of CRF-RNN improved MVCNN performance. However, there was a limitation in that it was not possible to distinguish similar geometric shapes (slab and covering). Future works will be conducted to solve this issue through model training using additional property variables or relational information between elements that can distinguish between the two elements.

Despite the limitation and future work, the results of this study are then encouraging as the test was performed on an entirely separate BIM model that was not used in the training. Whereas, existing studies conducted testing using a designated portion of the dataset, this study was more stringent in using a BIM model previously not seen by the deep learning models. ACC's of 95.38% suggests that our model can be deployed to classify and check the classifications of newly created architectural BIM models, albeit it be for 13 elements trained in this study.

7 Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NO. NRF-2020R1A2C110074112)

References

[1] Bassier M., Vergauwen M. and Van Genechten B. Automated classification of heritage buildings for as-built BIM using machine learning techniques. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(2W2):25–30, 2017.

[2] Belsky M., Sacks R and Brilakis I. Semantic enrichment for building information modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(4):261–274, 2016.

[3] Bloch T. and Sacks R. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. *Automation in Construction*,

91(21):256–272, 2018.

[4] Eastman C., Lee J. M., Jeong Y. S., and Lee J. K. Automatic rule-based checking of building designs. *Automation in construction*, 18(8):1011–1033, 2009.

[5] Kim J. H., Jang P. G., and Jean B. H. A parametric modeling methodology optimized for Korean traditional house. *Journal of the Architecture Institute of Korea Planning & Design*, 28(2):105–112, 2012.

[6] Kim J. S., Song J. Y. and Lee J. K. Recognizing and classifying unknown object in BIM using 2D CNN. In *International Conference on Computer-Aided Architectural Design Futures*, pages 47–57, Daejeon, Republic of Korea, 2019.

[7] Koo B. S., Jung R. K., and Yu Y. S. Automatic classification of wall and door BIM element subtypes using 3D geometric deep neural networks. *Advanced Engineering Informatics*, 47:101200, 2021.

[8] Koo B. S., Jung R. K., Yu Y. S. and Kim I. H. A geometric deep learning approach for checking element-to-entity mappings in infrastructure building information models. *Journal of Computational Design and Engineering*, 8(1):239–250, 2021.

[9] Koo B. S., La S. M., Cho N. W., and Yu Y. S. Using support vector machines to classify building elements for checking the semantic integrity of building information models. *Automation in Construction*, 98:183–194, 2019.

[10] Lim J. T. and Kim N. U. A Study on the Design Process by Parametric Modeling Method. In *The 60th Anniversary and Annual Conference of the Architectural Institute of Korea*, pages 523–526, Seoul, Republic of Korea, 2005.

[11] Lomio F., Farinha R., Laasonen M., and Huttunen H. Classification of Building Information Model (BIM) Structures with Deep Learning. In *2018 7th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6, Tampere, Finland, 2018.

[12] Ma L., Sacks R. and Kattell U. Building model object classification for semantic enrichment using geometric features and pairwise spatial relations. In *2017 Lean and Computing in Construction Congress (LC3)*, pages 373–380, Heraklion, Greece, 2017.

[13] Maturana D. and Scherer S. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 922–928, Hamburg, Germany, 2015.

[14] Qi C. R., Su H., Mo K., and Guibas L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, Honolulu, Hawaii, 2017.

[15] Su H., Maji S., Kalogerakis E., and Learned-Miller E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, Santiago, Chile, 2015.

[16] Venugopal M., Eastman C. M. Sacks R., and Teizer J. Semantics of model views for information exchanges using the industry foundation class schema. *Advanced engineering informatics*, 26(2):411–428, 2012.

[17] Wu Z., Song S., Khosla A., Yu F., Zhang L., Tang X., and Xiao J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, Bonston, Massachusetts, 2015.