electronics

MDPI

Article

3DPlanNet: Generating 3D Models from 2D Floor Plan Images Using Ensemble Methods

Sungsoo Park [1,2] and Hyeoncheol Kim [2,*]

[1] Korea Virtual Reality Inc., Seoul 05719, Korea; littlepu@naver.com
[2] Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea
* Correspondence: hkim64@gmail.com

Abstract: Research on converting 2D raster drawings into 3D vector data has a long history in the field of pattern recognition. Prior to the achievement of machine learning, existing studies were based on heuristics and rules. In recent years, there have been several studies employing deep learning, but a great effort was required to secure a large amount of data for learning. In this study, to overcome these limitations, we used 3DPlanNet Ensemble methods incorporating rule-based heuristic methods to learn with only a small amount of data (30 floor plan images). Experimentally, this method produced a wall accuracy of more than 95% and an object accuracy similar to that of a previous study using a large amount of learning data. In addition, 2D drawings without dimension information were converted into ground truth sizes with an accuracy of 97% or more, and structural data in the form of 3D models in which layers were divided for each object, such as walls, doors, windows, and rooms, were created. Using the 3DPlanNet Ensemble proposed in this study, we generated 110,000 3D vector data with a wall accuracy of 95% or more from 2D raster drawings end to end.

Keywords: deep learning; 2D floor plan; 3D model; data based methods; rule based methods; ensemble methods

Citation: Park, S.; Kim, H. 3DPlanNet: Generating 3D Models from 2D Floor Plan Images Using Ensemble Methods. *Electronics* **2021**, *10*, 2729. https://doi.org/10.3390/electronics10222729

Academic Editor: Eva Cernadas

Received: 28 September 2021
Accepted: 3 November 2021
Published: 9 November 2021

1. Introduction

The history of architecture and architectural drawings is so long that it cannot be separated from human culture. However, until the second half of the 20th century, when the spread of PCs made the use of computer-aided design (CAD) software [1] possible, architectural drawings were completed on paper and archived only on paper or as scanned raster images. Even today, when CAD is widely used, CAD data are employed only for professional purposes, and most architectural drawings are used and distributed in an image format. Because there is a limit to the information that can be checked in these 2D images, there has always been a need for transforming them into vector data and 3D shape models. However, much time and effort is required to manually create numerous images as 3D vector data.

For example, we can quickly identify room extents or areas, door layouts, and shapes of drawings. Recently, some large-scale buildings use building information modeling (BIM) data for 3D building object information to systematically manage the life cycle, but most buildings do not use it for reasons, such as cost or construction period. This study can be the basis for BIM data [2]. Ultimately, this study can be utilized as a basic research for automatically generating architectural drawings using artificial intelligence.

Converting raster images into vector-based drawings is a challenge for researchers, and, accordingly, numerous studies have been conducted. An architectural plan consists of various structural architectural objects and auxiliary information objects, such as walls, doors/windows, rooms, furniture, materials, door/window rotation radius, dimensions, and various texts. Since an architectural plan represents the shape of a horizontal section at a specific height, its three-dimensional shape must also be predicted, for example, a wall

that is covered by the door/window image. These various objects must be classified and extracted, and their sizes should be meaningful.

Most of the problems tackled by existing studies were mainly solved through heuristic-based algorithms [3–15], and the applied drawings were kept in a small amount of samples. In recent years, deep learning has been used to extract objects from images, and the accuracy of the objects has been improved [16–21]. However, to accomplish this, much effort was required to prepare a large amount of training data and for annotating a floorplan dataset [22].

In order to overcome this limitation, we conducted learning with only a small amount of training data (30 floor plan images), and proposed an end-to-end 3DPlanNet Ensemble model that incorporates a complementary rule-based heuristic method. Despite the small amount of training data, a wall was created with an accuracy of 95% or more, and the 3D object was automatically created with an accuracy close to that of a previous study [16]. Additionally, using the drawing's area information, a 2D drawing without scale was created with an accuracy of 97% or more. As a result of this study, about 110,000 drawing images were converted into 3D vector data. In order to service to the actual system [23], the insufficient parts were corrected manually, and a questionnaire survey on the performance of the 3DPlanNet Ensemble methods was also conducted with the corresponding modifiers.

2. Related Work

The task of converting 2D architectural drawing images into vector data has a long history in the field of pattern recognition. For example, many studies have analyzed image drawings using various heuristic algorithms, such as Hough transform, image vectorization, and generic methods [3,4,6–13]. In [5,14,15], architectural semantic information was used with heuristics to convert AutoCAD 2D vector data into 3D models. The research in [24–29] was conducted to generate and analyze 3D models and drawings directly from data captured by 3D scans. However, their results show limitations in performance, such as accuracy and scalability, and could not be applied to the creation of a large number of 3D models.

Recently, there have been studies [16–21,30,31] that recognize 2D architectural plan images using deep learning based on data. For example, [17–21,30,31] analyzed image drawings using graph neural network (GNN) [17,18], generative adversarial network (GAN) [17,20], convolutional neural network (CNN) [19,21], global convolutional network (GCN) [30], and fully convolutional network (FCN) [31]. One study [16] showed the results of generating 3D models with high accuracy by extracting junction data through a convolutional neural network (CNN) modified with ResNet-152. A junction layer and a primitive layer were created through the junction data to create a wall and a closed curve. However, the method of creating a wall through junction data reduces the accuracy of the wall's creation. For example, even in a location where there is no wall, a faulty wall can be created through two collinear junction points. In order to solve this limitation, we focused on recognizing the wall line itself, not the junction data, and we were able to create walls with high accuracy.

Although a large amount of training data are required for data-based learning, the reality is that there are little training data. For deep learning, training/test data (870 [16], 319 [30], 500 [31]) were directly generated in the previous studies, but a great deal of time and effort was required. To solve this problem, we trained with 30 data (floor plan images) and used ensemble methods incorporating rule-based heuristic methods. As a result of this study, 110,000 3D models with a wall accuracy of more than 95% were produced and are currently being used as a web service [23].

3. Methods

Converting raster images into vector-based drawings is a challenge for researchers, and, consequently, numerous studies have been conducted. Architectural drawings consist of various structural architectural objects and auxiliary information objects, such as walls,

doors, windows, rooms, furniture, material, door/window rotation radius, dimensions, and various text. Recently, several studies [16–21,30,31] have recognized 2D architectural drawing images using deep learning based on data, but a large amount of learning data was required for learning. To solve this problem, we propose 3DPlanNet Ensemble methods that combine data-based methods with 30 pieces of learning data and rule-based heuristic methods.

As shown in Figure 1, the 3DPlanNet is composed of pattern recognition, object detection data (learning) based methods, node + edge generation, and object generation rule (heuristic) based methods. It can generate 3D vector models through sequential end-to-end steps. Each step is described in more detail below.

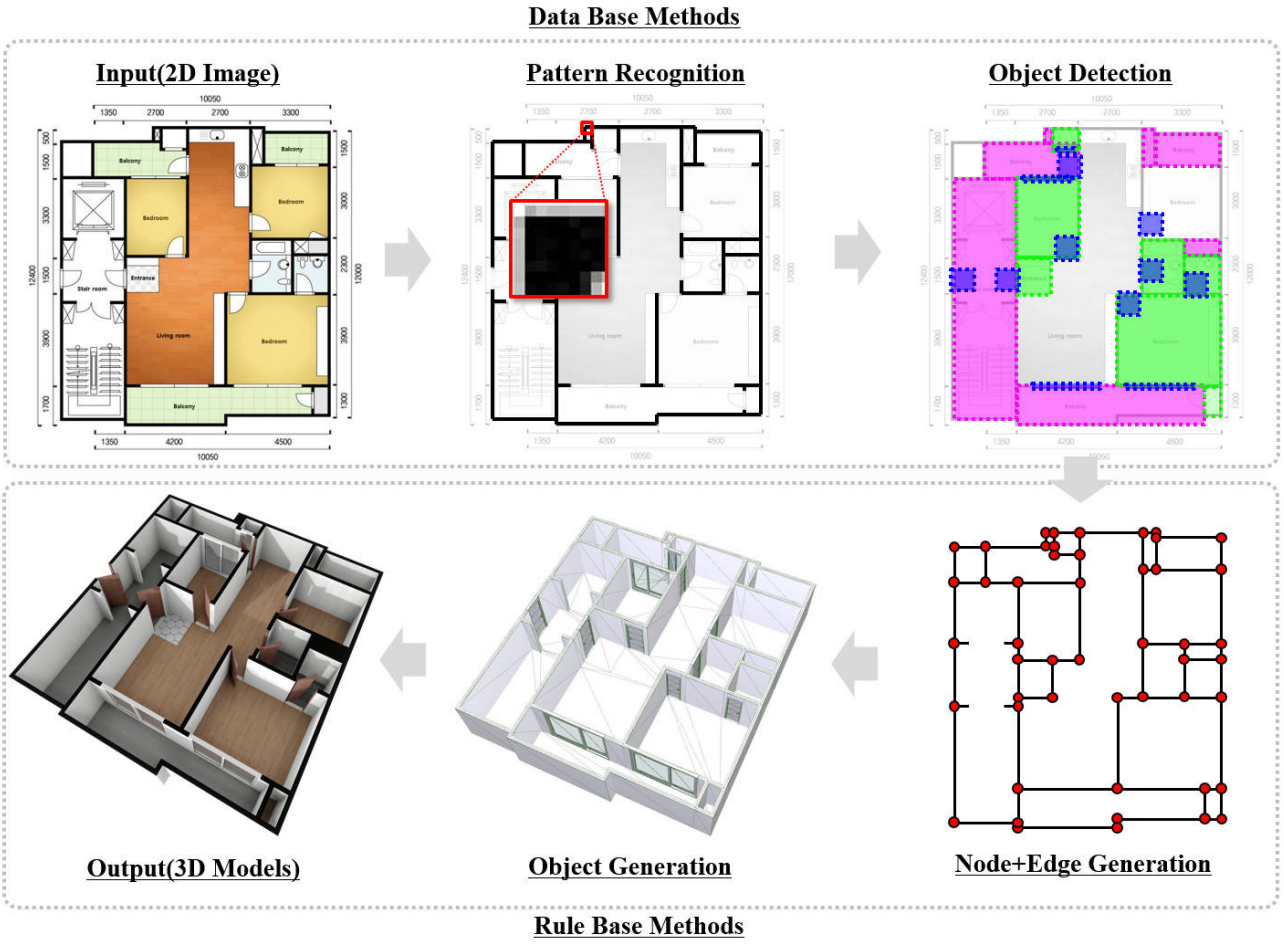

Figure 1. To convert a 2D raster floorplan (input) into 3D vector models (output), we propose 3DPlanNet Ensemble methods that combine data (learning) based methods and rule (heuristic) based methods.

3.1. Pattern Recognition

The task of converting 2D architectural plan images into vector data has a long history in the field of pattern recognition. For such conversions, the task of recognizing walls is the most important and difficult part of the research. In an architectural drawing, the division of the space and the layout of the doors/windows are all made through the composition of the wall. Since wall patterns are diverse, it becomes more difficult to determine the wall due to the shape of the doors/windows. We decided that it would be much easier to identify the wall by excluding the door/window area. Accordingly, it was possible to recognize the wall simply by learning it without the door/window area using a logistic regression model in units of a certain area (5×5 pixels).

Just as in [30], which restored the wall thickness using the moving kernel method, we restored the wall center using the same method. However, unlike [30], which determined

whether the wall was heuristic, we determined the wall center point by discriminating the logistic regression learning result for a certain area (5 × 5 pixels). In this study, focusing on the point that most architectural drawings consist of horizontal/vertical walls, only horizontal/vertical walls, excluding any diagonal walls, were recognized. As shown in Figure 2, when the wall was identified through the kernel method moving from left to right and the kernel method of 5 × 5 pixels moving from top to bottom, the center point of 5 × 5 pixels was restored to the wall center. Later, a vector-based wall line was created through an image vectorization technique based on the pixels stored in the node + edge generation stage. Section 3.4 will explain the node + edge generation in more detail.

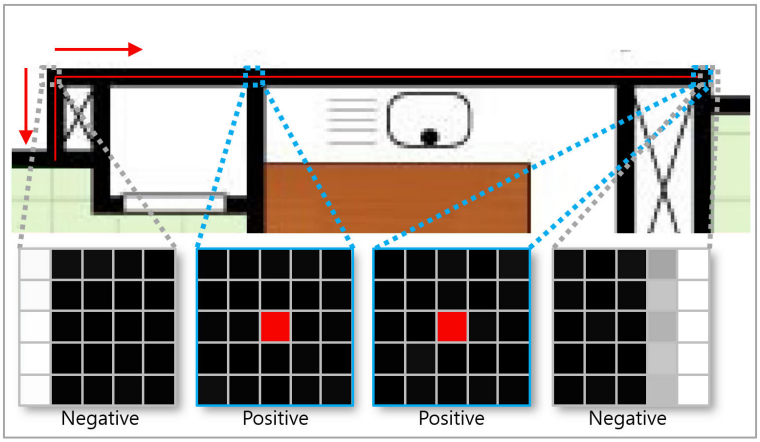

Figure 2. Restoring the wall center line using the moving kernel method.

As shown in Figure 3, the finally restored wall center line was able to recognize the center line in almost all the wall areas except for the door/window area. In the drawing image, several parallel center lines were restored for walls with a wall thickness of 5 pixels or more, but they were corrected in the node + edge generation step.

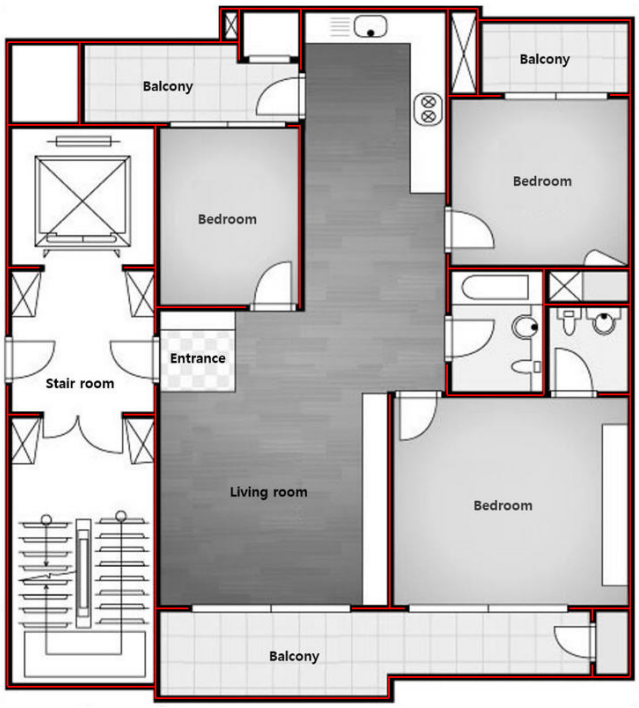

Figure 3. Restoring the wall center line (red pixels) of a 2D plan image resize.

3.2. Object Detection

In addition to wall information, 2D architectural drawings include various information, including doors, windows, rooms, furniture, materials, dimensions, and text. Naturally, how accurately the information can be extracted greatly affects the performance of the 3D vector models transformation. In the wall pattern recognition step, since recognition was performed excluding the door/window area, we needed to recognize the door/window area in order to increase accuracy. Using TensorFlow object detection API [32], we classified each object into wall junction, opening, and room, and then performed object detection.

Wall Junction Detection. In order to extract the wall junction from the image, we divided The wall junction into 4 types I (w2), L (w1), T (w3), and X (w4), as in [16], and each type, except for X (w4), was divided into 4 types (-1, -2, -3, -4). It was then classified into a total of 13 types (Figure 4). In order to minimize the indirection of other types of images, the training data area of the wall junction were minimized to less than 40 pixels, and the wall was planned to be connected through the extracted junction data. However, as we will discuss again in Section 4. Experiments, the accuracy of detection was low, so we had to exclude Wall Junction Detection from the actual experiment.

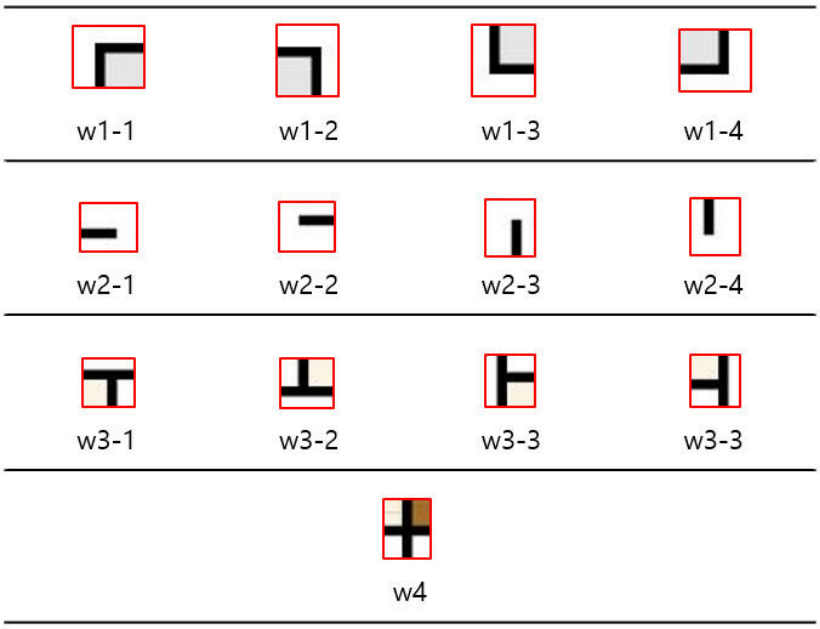

Figure 4. Wall junction types.

Openings (door/window) Detection. The shapes of the doors/windows are divided into a hinged type and a sliding type. In architectural drawings, the door is generally expressed as a hinged type (d1/d2/d4), and the window is expressed as a sliding type (d3) (Figure 5). Exceptionally, sliding doors can be used in architectural drawings, but they cannot be distinguished in the image. We extracted information about the type, location, size, and direction of the sliding door through detection.

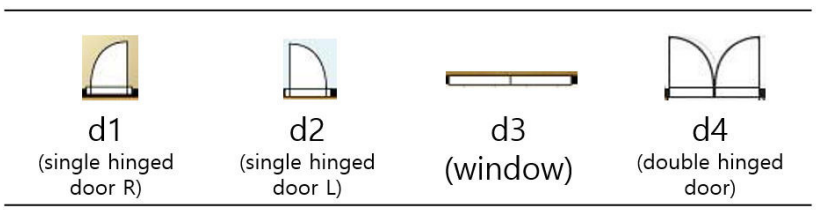

Figure 5. Opening types.

Room Detection. There are various room shapes, even for rooms having the same purpose, making detection very difficult. However, room types were essential information to improve the accuracy and to understand the use of space in the rule-based node + edge generation stage. The space is divided into 8 types: r1 (bedroom), r2 (restroom), r3 (entrance), r4 (balcony), r5 (stairs room), r6 (closet), r7 (duct), and r8 (living room) (Figure 6). Since the wall line was not directly generated through the room detection information, an error of several pixels was allowed, and it was only used to determine the purpose and location of the space.

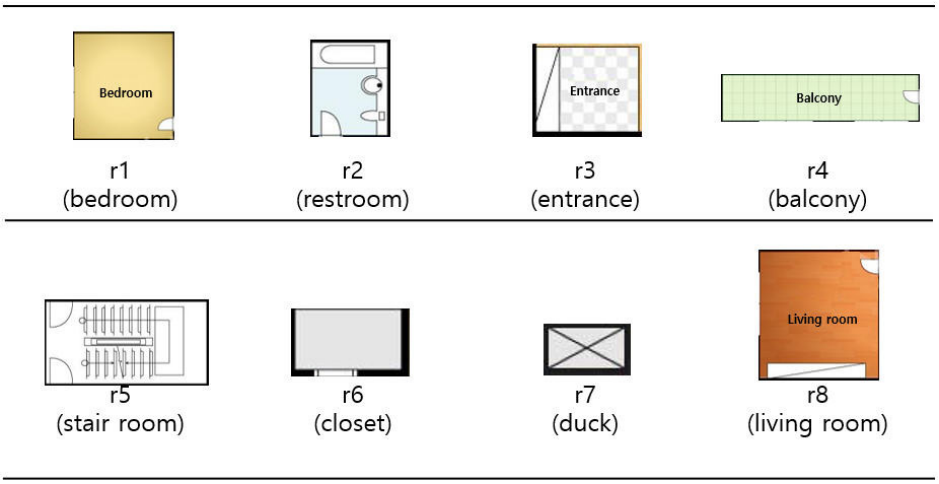

Figure 6. Room types.

3.3. Node/Edge Generation

In this step, nodes and edges are created through information analyzed based on data so far. Nodes and edges are created as wall junctions and walls in the subsequent object generation stage. In order to increase the accuracy, the wall center information restored in the pattern recognition step and the opening data/room data extracted in the object detection step were used. The restored wall center information was converted into a vector line through an image vectorization technique based on the stored pixels. In the node + edge generation step, a wall with high accuracy was created with the following generation rules.

The restored wall center information was converted into a vector line through a image vectorization technique based on the stored pixels. In the Node/Edge Generation step, a wall with high accuracy was created with the following generation rules.

Generation Rule: In order to correctly create node + edge, the following rules are basically proposed (Figure 7):

- Orthogonality: All edges are composed of horizontal/vertical walls only. Drawings with diagonal lines are excluded from this study;
- Substitution: The area from which the opening object (door/window) is extracted is replaced with a wall, and walls that are separated from each other are connected. Even if the walls that are separated from each other are not in a straight line, they are connected within the allowable distance (3 pixels);
- Duplication: Lines overlapping parallel within a certain distance (5 pixels) of the wall are deleted, leaving only the longest wall. If they are arranged alternately, the wall is extended to the longest length;
- Chamfer: According to the rule that the walls are all connected by a closed curve, a shape that is slightly separated is chamfered in a right angle shape;
- Merge/Extension: Several shortly separated lines on a straight line are merged into one line and extended if there is no other intersecting line.

Through this generation rule, an edge (wall) was created, and a node (wall junction) was created at a location where the edge ends or crosses (Figure 1).

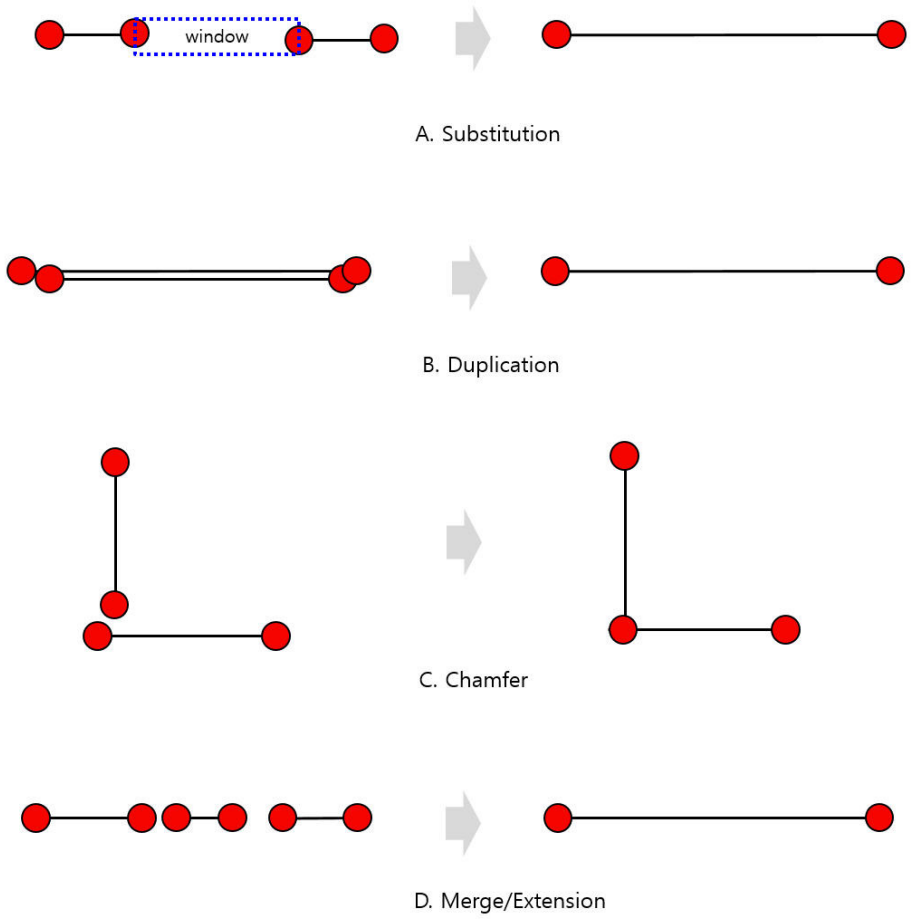

Figure 7. Node/Edge generation rules.

3.4. Object Generation

Object generation is the final step in generating 3D models in the form of 3D mesh objects after the node + edge information is generated. In order to create a more realistic shape of the walls, doors, windows, floor, and ceiling, we created an object as a face composed of a solid shape using the constructive solid geometry (CSG) algorithm. Object creation was performed according to the following procedure. The floor/ceiling were created by classifying the types for each purpose by using room detection information and a heuristic algorithm (Figure 1).

- Wall/Wall Junction: The wall is made of 6-sided boxes of a certain width (150 mm) and height (2400 mm), and each wall is created as an object separated by wall junctions;
- Opening (Door/Window): First, a hole is made in the wall where the door/window is placed using a CSG algorithm, and then a pre-modeled door/window object is placed at that location;
- Floor/Ceiling: These were created by calculating the face in the form of a closed curve along the inner line of the wall.

3.5. Plan Scaling

All architectural drawings have scales and dimensions, but there are many drawings that do not include dimensions in 2D raster images. Yet, it is impossible to directly extract size information from a 2D drawing image without dimensions. In order to solve this problem, we calculated the size of the image by using the exclusive area information of the

drawing. The exclusive area is the apartment household area excluding balconies, ducts, and common spaces. The exclusive area of the model created in the object generation step was calculated and the size was adjusted by comparing it with the actual area value.

$$S^* = \sqrt{A_{gt}/A_c} \tag{1}$$

The first created 3D model was scaled with S^*, which is the ratio of the ground truth exclusive area (A_{gt}) to the calculated exclusive area (A_c) (Equation (1)). The accuracy of the scale of the drawing depends on how accurately the space of the exclusive area is recognized and created (Figure 8 green area). In this manner, it was possible to create drawings without dimensions with an accuracy of more than 97% (Table 1).

Table 1. Object generation accuracy/recall.

Method	Wall		Wall Junction		Opening		Room		Scale
	Acc.	Recall	Acc.	Recall	Acc.	Recall	Acc.	Recall	Acc.
Ahmed et al. [3]	N/A	N/A	74.9	57.5	61.3	48.7	N/A	N/A	N/A
Liu et al. [16]	N/A	N/A	94.7	91.7	91.9	90.2	84.5	88.4	N/A
Kim et al. [20]	81.0	N/A	N/A	N/A	70.0	N/A	66.0	N/A	N/A
Lu et al. [21]	87.0	N/A	N/A	N/A	78.0	N/A	77.7	N/A	N/A
Ours (rule base)	72.3	72.3	64.7	94.3	N/A	N/A	N/A	N/A	N/A
Ours (data base)	72.3	72.3	49.0	49.0	84.4	84.4	82.5	82.5	N/A
Ours (ensemble)	**95.3**	**96.2**	92.2	**97.1**	84.4	84.4	**85.5**	86.5	**97.1**

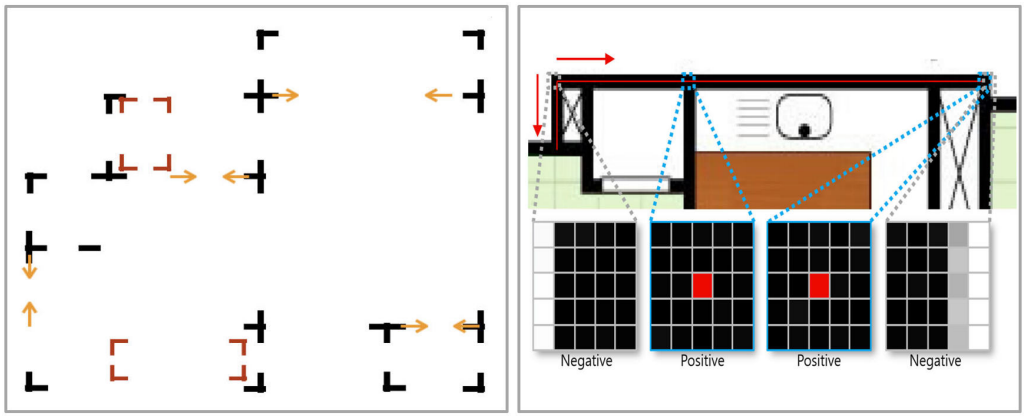

Figure 8. Detection methods of Liu et al. [16] (**left**), detection methods used in our study (**right**).

3.6. Comparison

We propose an ensemble method to convert a 2D floor plan image into a 3D model with similar accuracy to [16], even with a small amount of training data. We reduced the need for training data by combining a data-driven approach with a rule-based method. We believe that other important causes of this achievement are the method of annotation in the training data and the order of wall detection.

In the study of [16], as shown in Figure 8, wall junctions are first extracted and the walls are inferred from them, but we extract the walls first and then find the wall junctions. We thought that the method of finding the wall junction from the wall was much more accurate due to the structural elements of the architectural drawing, and we proved this through our experiments. The typical failure cases of the wall junction method mentioned in study [16] would also appear to support our choice of method.

4. Experiments

In the experiment, we compared the results of the vector data generated by the rule-based methods, the results of the vector data generated by the data-based methods, and the results of the ensemble method that combines the two methods. For the dataset, 30 training data and 30 test data were randomly selected from 110,000 2D apartment drawing images (resolution: 923 × 676) [33]. This dataset consists of about 110,000 2D apartment drawing images with an area between 20 m^2 and 500 m^2 collected from all over the country. As shown in Figure 1 Input (2D Image), the 2D Floor Plan Image involves random dimension lines, text and patterns according to room use, walls, and door/window symbols as raster data. Since it is not the same dataset, it cannot be accurately compared, but for the final performance comparison it was compared with Ahmed et al. [3], Liu et al. [16], Kim et al. [20], Lu et al. [21]. The experiment environment was tested on a GPU-based i7-875H 2.20 GHz notebook.

4.1. Rule Based Methods

Only a pure heuristic method, without data-based pattern recognition and object detection, was used to extract the walls and create junctions accordingly. Unlike the existing method [16], which analyzes the wall junction and then creates a wall from it, we first analyzed the wall and then created the wall junction. Accordingly, as shown in Table 1, it was confirmed that the recall value was larger than that of Liu et al. [16]. Moreover, because the shape of the wall was very clear, there was no difference in the data-based method and performance.

4.2. Data Based Methods

Pattern Recognition. In one flat image, a 5 × 5 pixel area was defined as one instance, and the model was trained through a total of $(923 - 4) \times (676 - 4)$ datasets and logistic data on whether or not the wall was located. An MLP-based logistic regression model that had two hidden layers composed of 500 nodes was used. The recognition of walls, excluding the door/window area, was almost 99.9% or more accurate, but when the door/window area was included, an accuracy of 72.28% was confirmed (Table 1).

Object Detection. TensorFlow object detection API [32] was used as an experimental model for object detection, and each object detection was performed by dividing it into wall junction, opening, and room. As mentioned above, 30 training data and 30 test data were selected and studied at random. As shown in Figure 9 and Table 1, the wall junction had a very low accuracy, with an average of 49.0%. The reason for this is that the number of training data and the recognition area itself were very small (40 × 40 pixels or less). As shown in Figures 9–11 and Table 1, for opening and room detection, a stable recognition of more than 80% was possible. In addition, room detection was divided into an exclusive area (green) and a common area (pink) for scaling using a dedicated area, as shown in Figure 9.

4.3. Ensemble Methods

As shown in Table 1, compared to the performance of each of the rule-based methods and data-based methods, the ensemble method that combines the two methods shows superior performance. In particular, the wall was created with a remarkable performance of 95.3% accuracy and 96.2% recall. When compared to [16], which was trained with a large amount of training data, it can be seen that the accuracy (Room accuracy : 85.5%) of other objects is almost similar. Unlike [16], which generates walls through wall junction information, it is interesting to note that the recall performance (97.1%) of wall junctions is high as a result of generating wall junction information through wall information, contrary to [16]. In addition, the scale of the drawings without dimension information could be converted to a size with an accuracy of 97.1%.

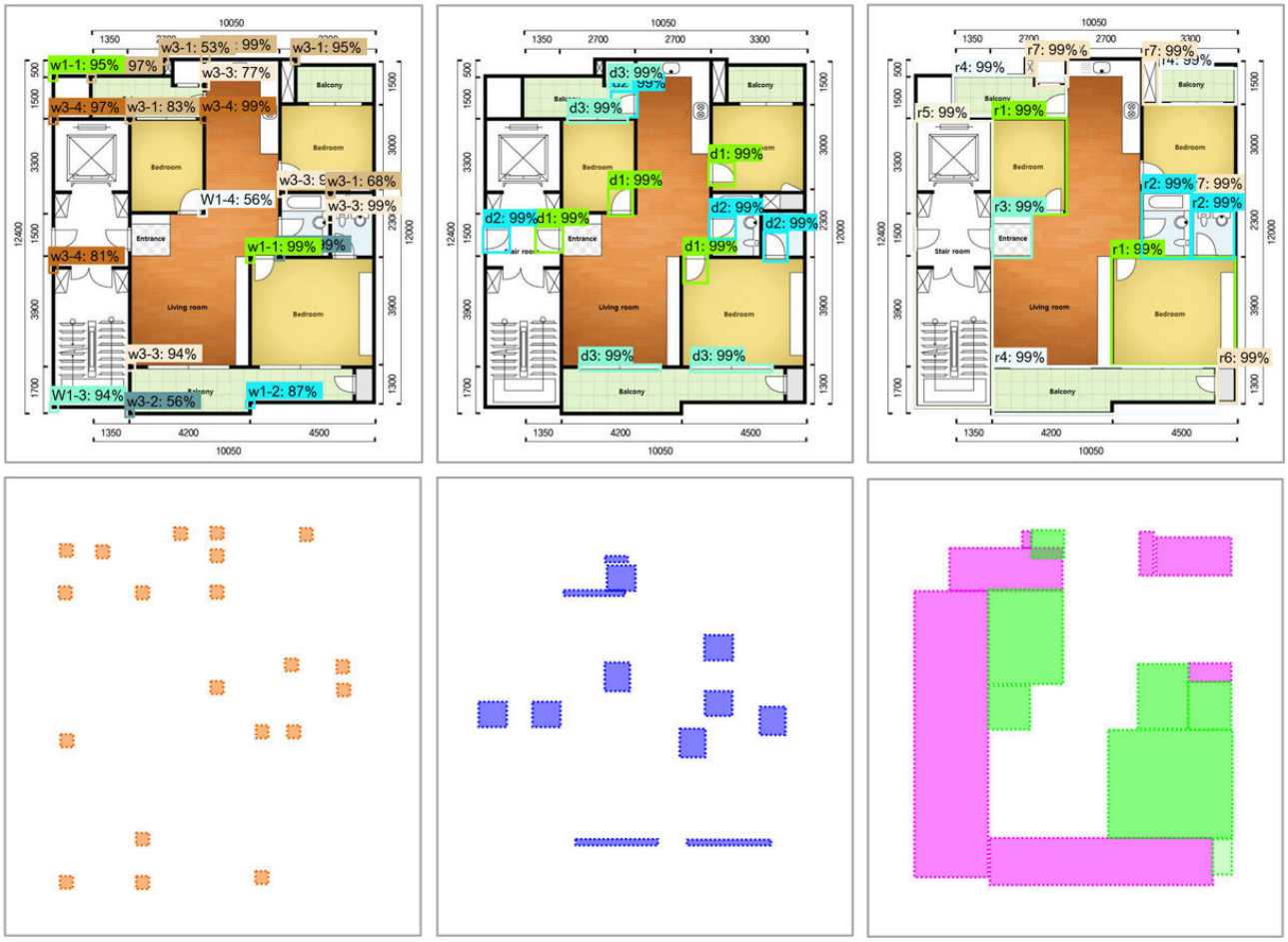

Figure 9. wall junction (**left**)/opening (**center**)/room (**right**) detection result (exclusive area: green, common area: pink).

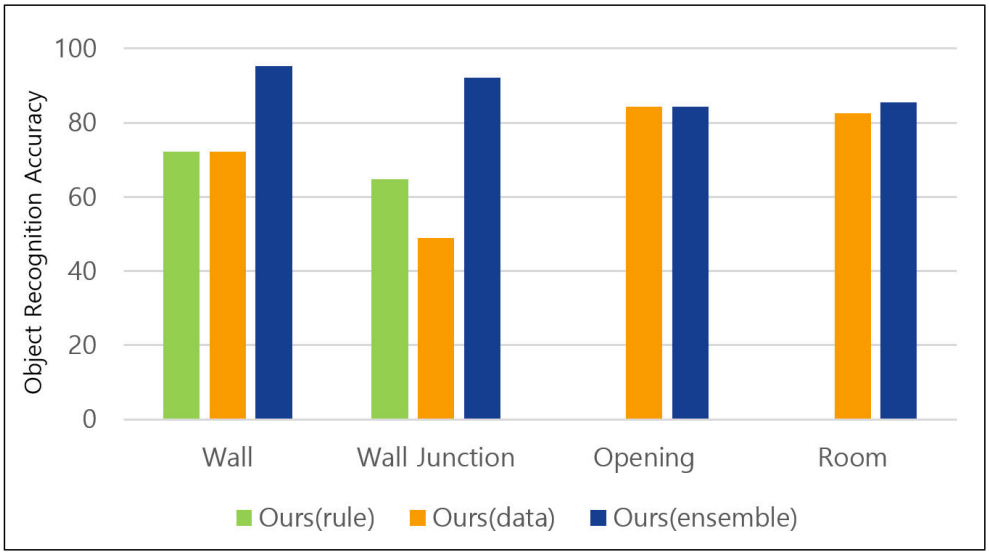

Figure 10. Object recognition accuracy of each class.

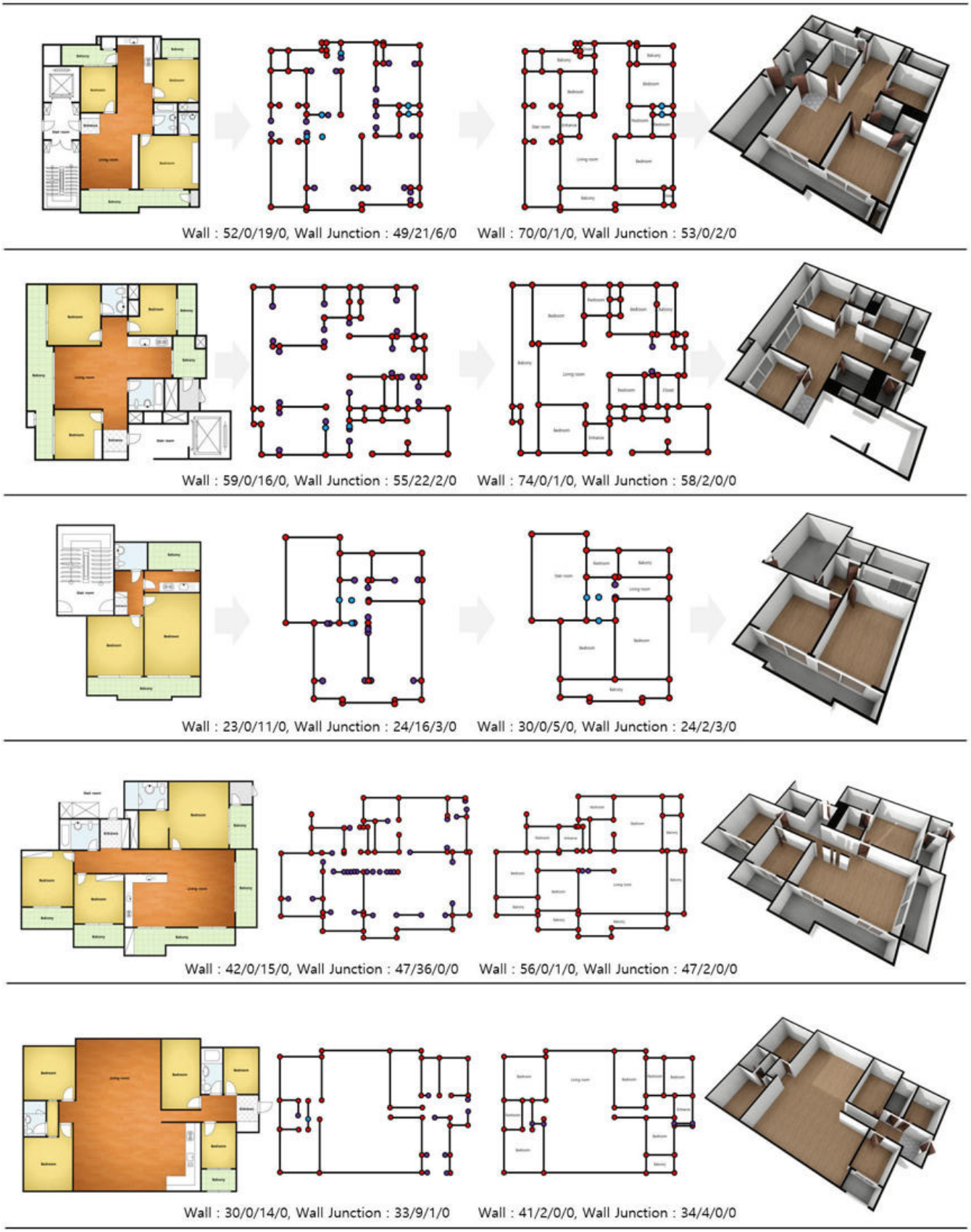

Figure 11. Result of generating various 2D Floorplan images. From the left, input data, data based methods result (TP/FP/FN/TN), ensemble method results (TP/FP/FN/TN), output data.

4.4. Metrics

In order to accurately measure the performance results, it was assumed that the walls were all separated by wall junctions, as shown in Figure 12 (left image), and the location, direction, and length of each wall were compared with the ground truth. In order to accurately measure the performance, a wall with a lower height between the entrance and the living room was also included in the comparison. For the wall junction, as shown in Figure 12 (right), the intersection and start points of all walls were checked.

The accuracy was calculated by counting the number of nodes (wall, wall junction, opening, room) existing in the ground truth image and the generated 3D model for 30 randomly selected apartment drawing images. In this experiment, performance was compared through the values of Accuracy = (TN + TP)/(TN + TP + FN + FP) and Recall = TP/(TP + FN).

- True Positive (TP): Exists in the floorplan 2D image and exists in the created 3D model;
- True Negative (TN): If it does not exist in the floorplan 2D image and does not exist in the created 3D model;
- False Positive (FP): If it does not exist in the floorplan 2D image and exists in the created 3D model;
- False Negative (FN): Exists in the floorplan 2D image and does not exist in the created 3D model.

Figure 12. Ground truth wall count: 71 (red dots in the **left** image), wall junction: 55 (blue dots in the **right** image).

4.5. Object Generation

Visual C++ and a real-time rendering engine [34] were used to create 110,000 3D mesh objects and to capture 3D model images. First, a volume object (wall) with a basic thickness was created between the wall junction and the wall junction. The wall mesh was removed at the door/window position using the constructive solid geometry (CSG) algorithm, and the 3D model was completed by placing the 3D basic door/window. Finally, a 3D model image was captured through a real-time rendering engine expressing global illumination. From the first step of extracting the wall and object information from a 2D image through a pre-trained model, to the last step of creating 3D mesh objects and capturing 3D images by combining data-based extraction information and heuristic methods, we designed 3DPlanNet Ensemble, which proceeds end to end. A total of 110,000 3D mesh objects images created by the 3DPlanNet Ensemble are currently being serviced in [23].

4.6. Robustness

We created 110,000 3D models that we collected through this study and are currently used as internet website services in [23]. As serviced in real estate site, it is applicable to many datasets and has proven its robustness. In addition, Figure 13 shows that high-accuracy wall centerline detection is possible, even from images from the internet.

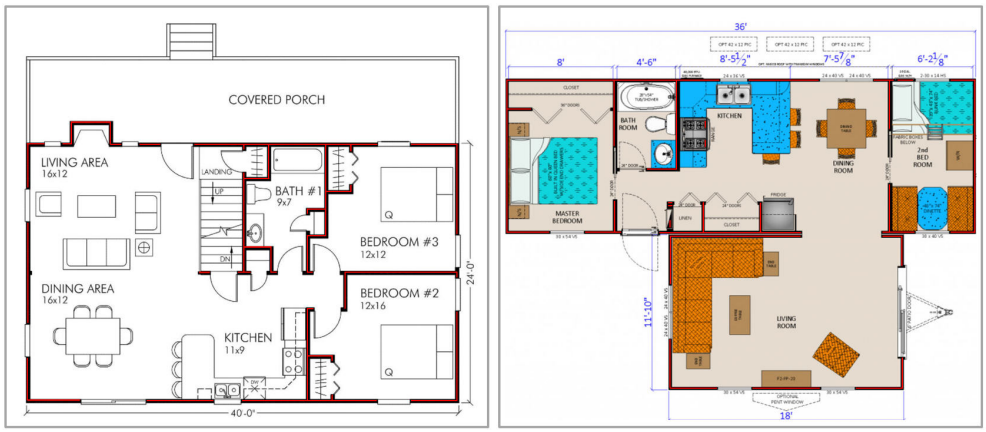

Figure 13. Floorplan vectorization results (red line) on an image from the internet (without object detection).

4.7. Survey

Despite the high accuracy in this study, 100% accurate information must be provided for use in actual industrial fields. In order to compensate for the lack of accuracy in this study, 10 modifiers supplemented the 3D vector data for several months. The results of the survey (Figure 14) show that the modifiers using 3DPlanNet Ensemble in this study felt qualitatively rather than quantitatively. Considering this an answer to the situation where additional modifications were made to the sloping walls and furniture arrangements excluded from this study, it can be seen that the qualitative evaluation of the 3DPlanNet Ensemble methods is very high.

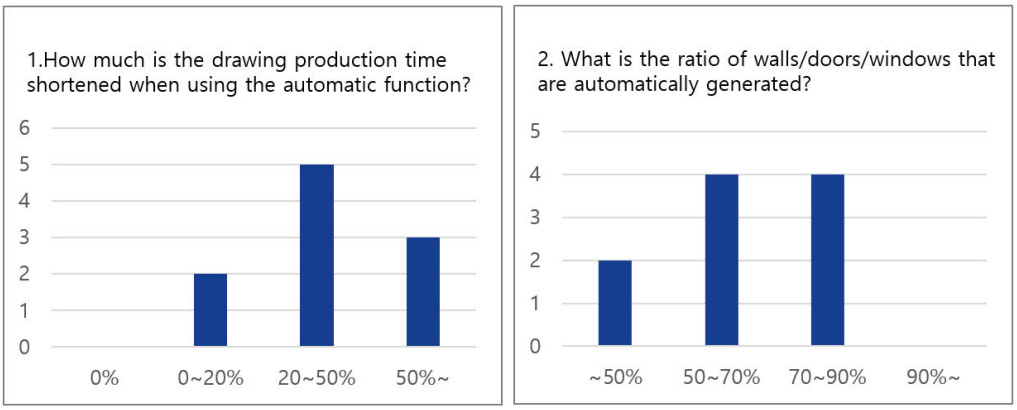

Figure 14. Ensemble methods usability survey results.

5. Conclusions and Future Work

In this study, 2D raster drawings were automatically converted to 3D vector models. Unlike previous studies that required large amounts of data to perform deep learning, this study proposed 3DPlanNet Ensemble methods that combine data-based models learned with 30 training data and rule-based heuristic methods. Through this study, the remarkable achievement of restoring the wall with an accuracy of 95% or more was shown, and a drawing without dimensions was created with a size accuracy of 97% or more.

The contributions of this study are as follows:

- The 3DPlanNet Ensemble method, which combines data-based methods and rule-based methods, was proposed;
- A wall object was created with an accuracy of more than 95% from a 2D drawing image;
- By combining models trained with less than 30 data and heuristic methods, the accuracy was significantly improved;
- Drawings without dimensions were created with a size accuracy of 97% or more.

Future Work. This study was limited to drawings with vertical/horizontal walls. In addition, 1.1 million 2D apartment drawings with clear wall shapes and similar patterns were created. Even if the node + edge was accurately restored from the 2D image, the step of creating a 3D mesh with volume was a completely different issue, and there was a limit to creating a perfect mesh. Beyond these limitations, it is necessary for continued research concerning the performance and perfect 3D modeling for various types of images.

Author Contributions: Conceptualization, S.P. and H.K.; writing—original draft preparation, S.P. and H.K.; writing—review and editing, S.P. and H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-0-01405) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Autocad. Available online: http://www.autodesk.com/products/autocad/overview (accessed on 27 September 2021).
2. Bryde, D.; Broquetas, M.; Volm, J.M. The project benefits of building information modelling (BIM). *Int. J. Proj. Manag.* **2013**, *31*, 971–980. [CrossRef]
3. Ahmed, S.; Liwicki, M.; Weber, M.; Dengel, A. Improved automatic analysis of architectural floor plans. In Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; pp. 864–869.
4. Macé, S.; Locteau, H.; Valveny, E.; Tabbone, S. A system to detect rooms in architectural floor plan images. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, Boston, MA, USA, 9–11 June 2010; pp. 167–174.
5. Lewis, R.; Séquin, C. Generation of 3D building models from 2D architectural plans. *Comput.-Aided Des.* **1998**, *30*, 765–779. [CrossRef]
6. Or, S.h.; Wong, K.H.; Yu, Y.k.; Chang, M.M.y.; Kong, H. Highly Automatic Approach to Architectural Floorplan Image Understanding & Model Generation. *Pattern Recognit.* **2005**, 25–32. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.4263&rep=rep1&type=pdf (accessed on 27 September 2021).
7. de las Heras, L.P.; Ahmed, S.; Liwicki, M.; Valveny, E.; Sánchez, G. Statistical segmentation and structural recognition for floor plan interpretation. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2014**, *17*, 221–237. [CrossRef]
8. Cabrera Vargas, D.A. Wall Extraction and Room Detection for Multi-Unit Architectural Floor Plans. Ph.D. Thesis, University of Victoria, Victoria, BC, Canada, 2018.
9. Moloo, R.K.; Dawood, M.A.S.; Auleear, A.S. 3-phase recognition approach to pseudo 3D building generation from 2D floor plan. *arXiv* **2011**, arXiv:1107.3680.
10. de las Heras, L.P.; Valveny, E.; Sanchez, G. Combining structural and statistical strategies for unsupervised wall detection in floor plans. In Proceedings of the 10th IAPR International Workshop on Graphics Recognition, Bethlehem, PA, USA, 20–21 August 2013; pp. 123–128.
11. Kashlev, D. Efficient 3D Building Model Generation from 2D Floor Plans. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2008.
12. Berkhahn, V.; Kinkeldey, C.; Schleinkofer, M.; Rank, E. Re-Engineering Based on Construction Drawings-From Ground Floor Plan to Product Model. In Proceedings of the International Conference on Computing in Civil and Building Engineering, Weimar, Germany, 2–4 June 2004.
13. Okorn, B.; Xiong, X.; Akinci, B.; Huber, D. Toward automated modeling of floor plans. In Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission, Paris, France, 17–20 May 2010; Volume 2.
14. Xu, D.; Jin, P.; Zhang, X.; Du, J.; Yue, L. Extracting indoor spatial objects from CAD models: A database approach. In Proceedings of the International Conference on Database Systems for Advanced Applications, Hanoi, Vietnam, 20–23 April 2015; pp. 273–279.
15. Zhu, J.; Zhang, H.; Wen, Y. A new reconstruction method for 3D buildings from 2D vector floor plan. *Comput.-Aided Des. Appl.* **2014**, *11*, 704–714. [CrossRef]

16. Liu, C.; Wu, J.; Kohli, P.; Furukawa, Y. Raster-to-vector: Revisiting floorplan transformation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2195–2203.
17. Dong, S.; Wang, W.; Li, W.; Zou, K. Vectorization of Floor Plans Based on EdgeGAN. *Information* **2021**, *12*, 206. [CrossRef]
18. Vidanapathirana, M.; Wu, Q.; Furukawa, Y.; Chang, A.X.; Savva, M. Plan2Scene: Converting Floorplans to 3D Scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 10733–10742.
19. Goyal, S.; Chattopadhyay, C.; Bhatnagar, G. Knowledge-driven description synthesis for floor plan interpretation. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2021**, 24, 19–32. [CrossRef]
20. Kim, S.; Park, S.; Kim, H.; Yu, K. Deep Floor Plan Analysis for Complicated Drawings Based on Style Transfer. *J. Comput. Civ. Eng.* **2021**, *35*, 04020066. [CrossRef]
21. Lu, Z.; Wang, T.; Guo, J.; Meng, W.; Xiao, J.; Zhang, W.; Zhang, X. Data-driven floor plan understanding in rural residential buildings via deep recognition. *Inf. Sci.* **2021**, *567*, 58–74. [CrossRef]
22. Cruz, S.; Hutchcroft, W.; Li, Y.; Khosravan, N.; Boyadzhiev, I.; Kang, S.B. Zillow Indoor Dataset: Annotated Floor Plans With 360deg Panoramas and 3D Room Layouts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2133–2143.
23. Available online: m.land.naver.com/ (accessed on 27 September 2021).
24. Turner, E.; Zakhor, A. Floor plan generation and room labeling of indoor environments from laser range data. In Proceedings of the 2014 International Conference on Computer Graphics Theory and Applications (GRAPP), Lisbon, Portugal, 5–8 January 2014; pp. 1–12.
25. Liu, C.; Wu, J.; Furukawa, Y. Floornet: A unified framework for floorplan reconstruction from 3d scans. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 201–217.
26. Phalak, A.; Badrinarayanan, V.; Rabinovich, A. Scan2Plan: Efficient Floorplan Generation from 3D Scans of Indoor Scenes. *arXiv* **2020**, arXiv:2003.07356.
27. Wang, L.; Sohn, G. An integrated framework for reconstructing full 3d building models. In *Advances in 3D Geo-Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 261–274.
28. Gankhuyag, U.; Han, J.H. Automatic 2D Floorplan CAD Generation from 3D Point Clouds. *Appl. Sci.* **2020**, *10*, 2817. [CrossRef]
29. Rottensteiner, F.; Briese, C. *Automatic Generation of Building Models from LIDAR Data and the Integration of Aerial Images*; ISPRS: Dresden, Germany, 2003.
30. Jang, H.; Yu, K.; Yang, J. Indoor reconstruction from floorplan images with a deep learning approach. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 65. [CrossRef]
31. Dodge, S.; Xu, J.; Stenger, B. Parsing floor plan images. In Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), Nagoya, Japan, 8–12 May 2017; pp. 358–361.
32. Available online: https://github.com/tensorflow/models/tree/master/research/object_detection (accessed on 27 September 2021).
33. Available online: http://corp.kovi.com/ (accessed on 27 September 2021).
34. Available online: https://unigine.com/ (accessed on 27 September 2021).