

Research Article

How Many Bedrooms Do You Need? A Real-Estate Recommender System from Architectural Floor Plan Images

Y.S. Gan ¹, Shih-Yuan Wang,² Chieh-En Huang,³ Yi-Chen Hsieh,³ Hsiang-Yu Wang,³ Wen-Hung Lin,³ Shing-Nam Chong,³ and Sze-Teng Liong ³

¹School of Architecture, Feng Chia University, Taichung, Taiwan

²Graduate Institute of Architecture, National Chiao Tung University, Hsinchu, Taiwan

³Department of Electronic Engineering, Feng Chia University, Taichung, Taiwan

Correspondence should be addressed to Sze-Teng Liong; stliong@fcu.edu.tw

Received 7 March 2021; Revised 16 June 2021; Accepted 28 July 2021; Published 6 August 2021

Academic Editor: Jianping Gou

Copyright © 2021 Y.S. Gan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper introduces an automated image processing method to analyze an architectural floor plan database. The floor plan information, such as the measurement of the rooms, dimension lines, and even the location of each room, can be automatically produced. This assists the real-estate agents to maximise the chances of the closure of deals by providing explicit insights to the prospective purchasers. With a clear idea about the layout of the place, customers can quickly make an analytical decision. Besides, it reduces the specialized training cost and increases the efficiency in business actions by understanding the property types with the greatest demand. Succinctly, this paper utilizes both the traditional image processing and convolutional neural networks (CNNs) to detect the bedrooms by undergoing the segmentation and classification processes. A thorough experiment, analysis, and evaluation had been performed to verify the effectiveness of the proposed framework. As a result, a three-class bedroom classification accuracy of ~ 90% was achieved when validating on more than 500 image samples that consist of the different room numbers. In addition, qualitative findings were presented to manifest visually the feasibility of the algorithm developed.

1. Introduction

When designing a building, the most indispensable tool for the architect is the floor plan, which also served as an important element to provide building guidelines and instructions for construction. In brief, a floor plan demonstrates the relationships between rooms, spaces, and other physical characteristics in a visual form. The floor plan usually specifies the basic layout dimensions (i.e., room size, height, and length) with an annotated scale factor. The architects often utilize several symbols or icons to enhance the interpretation of the design of a floor plan, for instance, simple outlines to indicate the features of walls, windows, and door openings. Besides, the floor plan suggests the arrangement of space by including the details of fixtures and furniture like the stove, bathtub, sofa, and toilet bowl. Nonetheless, the floor plan design is not limited to housing; it is applicable to any building type such as the office and Church.

Despite the fact that there are standard architectural symbols used to represent common building components and features, architectural drafting can basically be categorized into three types: process drawings (sketches/schematics), construction documents (drafted drawings), and presentation drawings (illustrated sketches). However, some architects artistically add visual interest to the floor plan to convey the intended idea and improve the understanding via graphic language. Although there are design guidelines for the sketching and drafting of a floor plan, it is subjective to the individual as the design shows some room for creativity and flexibility to attract the attention of viewers. Since there are plenty of design alternatives for the appearance of a floor plan, it requires substantial effort in rendering, extracting, learning, and recognizing the semantic information from the human perspective. Therefore, it increases the difficulty when processing and interpreting the floor plans using image analysis techniques.

To the best of our knowledge, this is the first attempt to comprehensively address and analyze the details of the rooms from the floor plans. In this paper, a benchmark framework is provided to automatically determine the location and the number of bedrooms from a floor plan. Following the significant success with deep learning frameworks, there has been a surge of interest resolving in computer vision-related problems. For instance, inspirational applications developed recently, such as biomedical field [1], unmanned aerial vehicle navigation [2], facial expression recognition [3], and soil structure assessment [4]. Thus, motivated by these works, several deep learning approaches are applied in the bedroom detection task herein to facilitate the automatic computational image processing analysis.

In summary, the main contributions of this paper are highlighted briefly as follows:

- (1) Application of a series of preprocessing techniques to improve the image quality, such as noise removal, wall thickness adjustment, and image scaling.
- (2) Proposal of a visual understanding mechanism to distinguish the bedroom from the floor plan by implementing the segmentation and classification processes.
- (3) Comprehensive experimentation on the dataset to verify the effectiveness of the algorithms evaluated. Both the qualitative and quantitative results are presented.

The remainder of the paper is organized as follows: a review of related literature is presented in Section 2. Subsequently, Section 3 describes the complete framework in detail, including the preprocessing method, the configuration of experiment settings, and performance metrics for result validation. Section 5 reports and analyzes the experimental results. Finally, conclusions are drawn in Section 6.

2. Related Work

A residential floor plan is a key element to provide the prospective buyer the essence of the property regarding the internal amenity, the outlook, and the interaction of the spaces that are viewed from above. The floor plan serves as the most essential guide for the home buyer to consider purchasing the property [5]. An eye-catching and expressive floor plan contains a colorful design, accurate scale, basic furniture icons, and a clear flow of spaces. A residential project has a briefer form and contains lesser information compared to a commercial project as it is a simple diagram that offers a conceptual starting point. Note that there is no standard real-estate floor plan and thus the oversimplified or overprofessional design may cause confusion to the buyer. Therefore, there is a lack of an automatic system to relate the architectural design to computer vision technology. Specifically, this automatic task is useful in assisting the buyers to quickly determine the number of bedrooms in each floor plan, classify the space according to the floor plan, analyze

the amount of space in a floor plan, and determine the exact locations for each room.

Albeit the emergence of artificial intelligence, the existing literature studies in analyzing the architectural designs with this technology are manageably finite. For example, Bayer et al. [6] suggested a semiautomatic method to automatically generate the floor plans of specific buildings. Concretely, the long short-term memory (LSTM) [7] was utilized as predictive modeling to achieve this task. However, instead of passing the floor plan images to the suggested model structure, the input information requires manual human effort to extract certain information from each sample image. Besides, the trained model insufficiently describes the detailed contextual characteristics of the floor plan, such as the actual position of the basic building blocks (i.e., walls, doors, and windows). A similar work that performs the floor plan generation is conducted by [8]. Yet, a deeper understanding of the current development state of image processing approaches is integrated to resolve for the best placement solution. The system is built on the Grasshopper environment to make the user interface design legible and easy to use.

On the other hand, Liu et al. [9] proposed a novel convolutional neural network (CNN) architecture, namely FloorNet, to reconstruct the 3D floor plans by physically scanning the indoor spaces over a visual display. Concretely, a triple-branch hybrid design is implemented to simultaneously process the 3D coordinate points, 2D floor plan, and images captured, to form the final floor plan. Thus, a dataset that comprises ~ 155 residential house layouts has been created. This work provides a reverse engineering solution on the floor plan image, whereby they did not process on the existing images; instead, they introduce a series of image analyses to generate a new floor plan image. The main intuition of this work is to cope with the absence of floor plan design problem, especially in the region like North America. Recently, an improved method that can generate the floor plan via 3D scanning with higher accuracy and higher speed is introduced by [10]. Specifically, the proposed pipeline, namely, Scan2Plan demonstrated outstanding results when evaluated on the publicly available Structured3D [11] and BKE [12] datasets.

De las Heras et al. [13] designed a structural element recognition system to detect the wall and room from four datasets with different ground-truth characteristics. The proposed approach first extract the main entities using statistical segmentation, such as the walls, door, and windows. Then, the structural compositions of the building are identified and distinguished using an image recognition technique. The authors claimed that the proposed algorithm is adaptable to any graphical representation, as it can extract the structural elements without prior knowledge of the modeling conventions of the plan. Nevertheless, further analysis and classification regarding the type or function of the room are not presented in the work. On the other hand, Khade et al. [14] focused on extracting the outer shape of the floor plan. A series of algorithms is suggested, in which geometric features such as area, corners, quadrants, distance, slope, and angle are involved. To evaluate the robustness of

the proposed framework, the experiments are tested on both the original and rotated images of a synthetic dataset.

Besides, [15] introduced a method to detect the elements of the walls and identify the key objects, as well as determining the characters from the floor plan images. The methodologies to accomplish this task are to adopt a fully convolutional network (FCN) model and an optical character recognition (OCR) technique. In brief, OCR is to retrieve meaningful room labeling. Experiments were carried out on two datasets, namely, a publicly available dataset (CVC-FP) and self-collected from the real-estate website. The experiments were performed on the datasets were wall segmentation, object detection (door, sliding door, kitchen oven, bathtub, sink, and toilet), and text recognition. Although promising wall segmentation was reported in the paper, the proposed approach did not compare to the state of the art. Besides, the number of testing samples to evaluate the object detection and text recognition performance was relatively few (<50 images). Thus, this work does not provide conclusive empirical evidence to verify the effectiveness of the proposed method.

On another note, Ahmed et al. [16] recognized the room by differentiating the walls with different thicknesses (i.e., thick, medium, and thin). They also pointed out that the wall thickness (i.e., thin wall) greatly affected the wall detection and hence influenced the recognition performance. The following year, the same research group [17] presented an automatic system to analyze and label the architectural floor plans. Particularly, the SURF [18] feature descriptor is utilized to spot the building element in enhancing semantic information extraction. This paper extends the previous paper by dividing the rooms into several subpartitions for the cases of a shared room. However, there is some false detection especially when the text touched the graphics component. Later, Ahmed et al. [19] separated the wall and text in the image using simple morphological erosion and dilation techniques. As a result, a promising result of the recall of 99% is exhibited when evaluating ~ 90 images. The prevalence performance mostly contributed to the success in removing the thin lines especially the text touching the lines.

Recently, Goyal et al. [20] proposed SUGAMAN (Supervised and Unified framework using Grammar and Annotation Model for Access and Navigation) to briefly describe the indoor environment in natural language from the building floor plan images. They represented the room features by adopting a local orientation and frequency descriptor (LOFD). Then, a single-layer neural network with 10 neurons is employed to learn the room annotations for room classification. To examine the effectiveness of the proposed algorithm, experiments are conducted on a dataset with more than 1000 image samples. Results demonstrated that the proposed method outperformed the state of the art by attaining higher classification accuracy when identifying the decor items (i.e., bed, chair, table, sofa, etc.). However, there are some scenarios where the proposed method incorrectly classifies the type of the room (i.e., kitchen vs. room). Besides, the result of floor plan creation based on LOFD is discouraging.

3. Proposed Method

The proposed algorithm aims to determine the existence of the bedroom and its respective location, as well as computing the number of bedrooms from an architectural floor plan image. Figure 1 lists the proposed framework of the proposed method. In brief, it incorporates four primary stages: wall extraction, wall thickening and door gap closing, room partitioning and decor item retrieval, and bedroom classification. The details of each step are discussed in this section by providing greater details in terms of their respective mathematical derivations and pseudocodes. For better visualization, Figure 2 depicts the conceptual diagram for each stage. Note that, the dataset involved in the experiment is Robin because the decor items shown on the images are clear and do not contain any text information.

3.1. Wall Extraction. The first step is to acquire the wall lines whilst removing the decor items. Prior to that, all the images are converted from RGB colorspace to grayscale. The original RGB image is shown in Figure 3(a) and the grayscale image is illustrated in Figure 3(b). Then, Otsu's algorithm is employed to perform the thresholding technique to enhance image contrast. Otsu's algorithm is one of the simplest methods to categorize the pixel intensity into two classes and the output is generated by minimizing the intraclass variance of each image. It is noticed that the decor items are appearing in thinner lines compared to the walls. Therefore, a simple morphological image processing method, namely, closing, is adopted to remove the objects with thin lines.

The closing morphology operation of image A by the structuring element B is the composite of erosion and dilation operations, defining as follows:

$$A \cdot B = (A \oplus B) \ominus B, \quad (1)$$

where $X \oplus Y$ denotes the dilation operation of image X by the structuring element Y . On the other hand, $X \ominus Y$ denotes the erosion operation of image X by the structuring element Y . In brief, the dilation and erosion operations can be defined as follows, respectively:

$$\begin{aligned} X \oplus Y &\equiv \{(x + y) \mid \forall x \in X, y \in Y\}, \\ X \ominus Y &\equiv \{x \in \mathbb{Z}^2 \mid (x + y \in X, \forall y \in Y)\}. \end{aligned} \quad (2)$$

By doing so, the decor items can be eliminated, in the meantime, preserving the wall lines, as illustrated in Figure 2(b). Besides, Figure 3(c) shows the output image after extracting the wall lines on another sample, whereby lesser furniture and elements exist in this 3-bedroom image. Since the decor items are represented by a set of simple synthetic architectural symbols made up of straight lines and simple shapes, this wall extraction process occupies relatively lesser computational cost and algorithm complexity.

3.2. Wall Thickening and Door Gap Closing. As there are noticeable door gaps after performing the wall extraction in the previous step (i.e., Figure 2(b)), the gaps are filled by

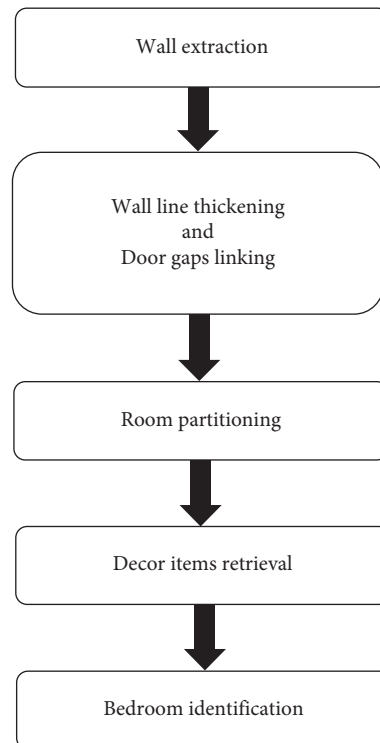


FIGURE 1: The proposed pipeline incorporates five stages.

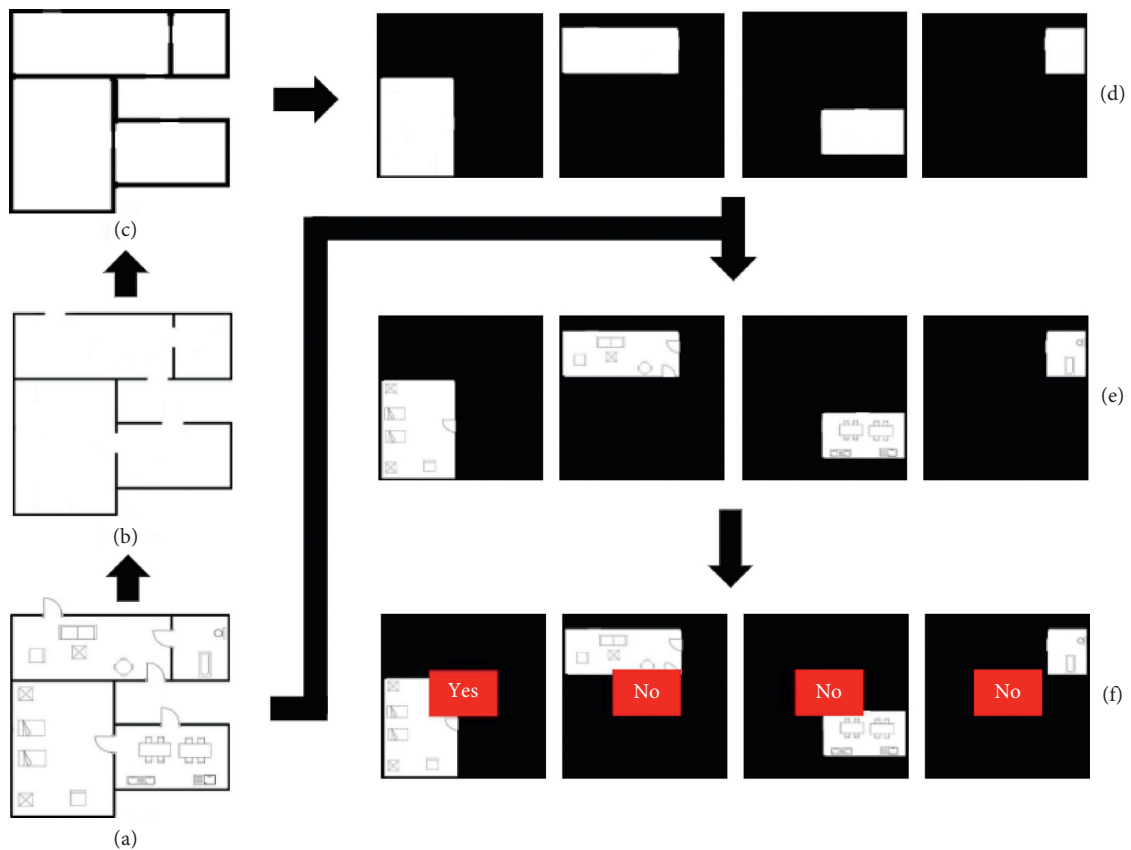


FIGURE 2: Overall process flow illustration: (a) original dataset; (b) wall extraction; (c) wall line thickening and door gaps linking; (d) room partitioning; (e) decor items retrieval, and (f) bedroom identification.

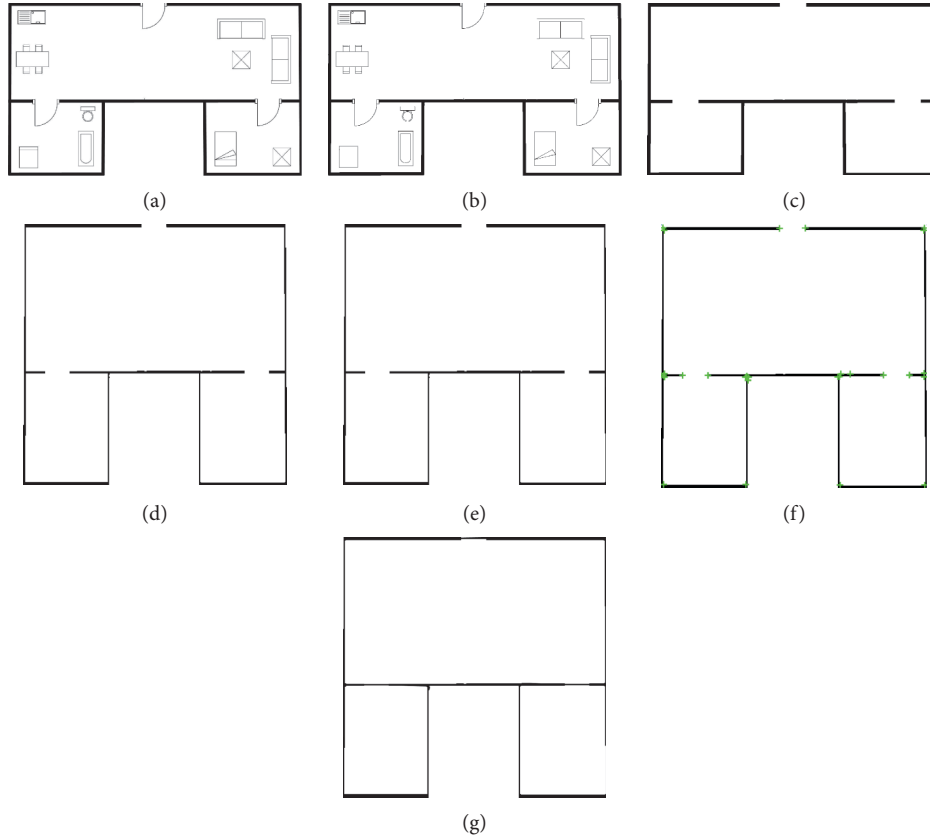


FIGURE 3: Output generated after performing each image transformation process: (a) original image; (b) pixel binarization; (c) wall extraction; (d) image dilation; (e) image resize to 400×400 ; (f) edge detection using FAST algorithm, and (g) door gaps linking.

thickening the walls to facilitate the gaps connection. A dilation morphological process is performed to improve the visual appearance of the wall lines, as shown in Figure 3(d). The dilation operation is expressed as

$$A \oplus B = \bigcup_{b \in B} A_b. \quad (3)$$

Since the resolution for the images in the Robin dataset is varied, the size is standardized such that the height and width are having the same length, namely, 400×400 , as shown in Figure 3(e). Next, the pairs of the door gaps coordinates for the linkage are identified. It can be achieved by applying Hough transform [21] and the FAST algorithm for edge detection. Subsequently, the width and the exact position of the door gaps are obtained, as shown in Figure 3(f).

The algorithm details to acquire the wall lines information are shown in Algorithm 1. In brief, there are three inputs required to link the door gaps, namely, the position, width, and the total number of the door gaps. Lines 1 and 2 in Algorithm 1 scan all the pixels in the image. Line 3 compares two coordinates to decide if they are the pair of the door gap, depending on the predefined width parameter. Lastly, line 4 links the pair of door gap coordinates, as shown in Figure 3(g).

Figure 2(c) shows the output of the door gaps linking after increasing the thickness of the wall.

3.3. Room Partition and Decor Items Retrieval. This step is to split the multiple rooms from each floor plan into individual rooms. Since the floor plan images in Robin are generated by the computer, it is found that there is some absurdly small size of rooms in a few floor plan images. Therefore, if the detected room region occupies less than 300 pixels, it will be eliminated from the next processing step. Figure 2(d) illustrates the segmented rooms from a floor plan.

After identifying and filtering the rooms, the rooms are overlapped with the original floor plan image, such that the decor items appear in every single room. Figure 2(e) illustrates the insertion of the original decor items on the respective partitioned rooms. The pseudocode to realize the room partition and decor items retrieval steps is shown in Algorithm 2. In brief, lines 1 to 27 detect the single rooms by identifying the pixel intensity values. A room should be encircled by a continuous pixel value of 0 in a rectangle shape. Concretely, the program will scan the entire image from left to right and top to bottom, to search for the first appeared pixel intensity value of 255. Then, that particular pixel is arbitrarily set to T , where T is initially defined as 1. This specific pixel is treated as a reference point such that the pixel values of all the four directions (i.e., top, bottom, left, and right) are set to 1. This pixel value updating step will be terminated until the pixel value 0 is met. Consequently, another scanning to search for the next pixel with an intensity of 255 will be performed. If found, that particular

```

Input:  $D(i, :), D(j, :)$  ← coordinate points of the door gaps
 $W$  ← width of the door gaps
 $N$  ← total number of door gaps
Output: the door gaps are closed and independent rooms can be identified
(1) for  $i = 1; i \leq N; i++$  do
(2)   for  $j = 1; j \leq N; j++$  do
(3)     if  $|D(i, :) - D(j, :)| \leq W$  then
(4)       Link  $D(i, :)$  and  $D(j, :)$ 
(5)     else
(6)       break.

```

ALGORITHM 1: Door gap closing.

```

Input:
 $A(i, j)$  ← coordinate point of an independent room image
 $m$  ←  $x$ -axis image size of an independent room
 $n$  ←  $y$ -axis image size of an independent room
Output:
 $T$  ← number of independent rooms
(1)  $v = 1;$ 
(2)  $T = 1;$ 
(3) While  $v \neq 0 \triangleright v = 1$  indicates that a new area has been found
(4) do
(5)    $v = 0;$ 
(6)    $b = 0;$ 
(7)   for  $i = 1; i \leq n; i++$  do
(8)     for  $j = 1; j \leq m; j++$  do
(9)       if  $A(i, j) == 255$   $\triangleright$  pixel intensity of 255 indicates white
(10)      then
(11)         $A(i, j) = T;$ 
(12)         $v = 1;$ 
(13)        break
(14)    $Q = 1;$ 
(15)   While  $Q \neq 0 \triangleright Q = 1$  indicates there may be missing points
(16)   do
(17)      $Q = 0;$ 
(18)     for  $i = 1; i \leq n; i++$  do
(19)       for  $j = 1; j \leq m; j++$  do
(20)         If  $A(i, j) == T$  then
(21)           Convert the pixel values to 255 in the top, bottom, left, and right directions of points  $A(i, j)$  into  $T$ 
(22)           This process is terminated when it encounters 0.
(23)         if a pixel is converted then
(24)            $Q = 1;$ 
(25)         else if  $A(i, j) == 255$  then
(26)           Determine whether there is  $T$  before 0 is encountered in the four directions of point  $A(i, j)$ .
(27)          $T = T + 1$ 
(28)   for  $i = 1; i \leq T; i++$  do
(29)     if there is  $i$  on the boundary or the total number of pixels with a value of  $i$  is less than 300  $\triangleright$  These two cases are not considered as
a room
(30)     then
(31)       continue;
(32)     else
(33)       Convert all  $i$  to 0 and convert other values to 255.
(34)       Stack the decor items on the respective detected rooms.

```

ALGORITHM 2: Room segmentation and decor items retrieval.

pixel is set to the value of $T = T + 1$, which is 2. Hence, it will be treated as the reference point again to update the surrounding pixels with value $T + 1$. The above-mentioned steps are repeated until pixel intensity 255 will no longer be found. Intuitively, the newly found pixel with the intensity of 255 is regarded as the new room definition. As such, the resultant T value denotes the total number of the new room formed in an image. Then, lines 28 to 33 determine if the room satisfies the room size requirement. Particularly, if the room has an area size of fewer than 300 pixels, the room is ignored. This is because there are some unusual rooms as the images are artificially generated by the computer. Finally, line 34 stacks the decor items to the individual detected rooms.

3.4. Bedroom Classification. This stage differentiates the rooms detected into bedroom/nonbedroom categories. The convolutional neural network (CNN) is employed as both the feature extractor and classifier for the bedroom classification. Intuitively, the shape, characteristics, patterns of a bed are the key features to decide if the image is a bedroom/nonbedroom. Thus, CNN architectures are expected to learn the features of the bed in order to make the correct predictions.

Several pretrained neural networks (i.e., AlexNet [22], ResNet [23], SqueezeNet [24], and GoogLeNet [25]) are utilized with slight modification. Note that, most of the parameters existing in the networks are transferred to adapt and learn the new characteristics of the floor plan images. Specifically, the parameters will be fine-tuned automatically throughout the learning progress. Concretely, these network architectures are comprised of five types of operation: convolution, ReLU, pooling, fully connected, and dropout. The bedroom images are first standardized to a certain size (i.e., $\aleph \times \aleph$).

- (1) Convolution and ReLU: The image performs a dot product between a kernel/weight and the local regions of the image. This step can achieve blurring, sharpening, edge detection, and noise reduction effect. ReLU is an elementwise activation function and is usually applied as the thresholding technique to eliminate the neurons that are playing a vital role in discriminating the input and is essentially meaningless.

Each e_{ij} pixel in the image is defined as

$$e_{ij}^l = \{f^l(x_{ij}^l + b^l) | i = 1, 2, \dots, \aleph, j = 1, \dots, \aleph\},$$

where $x_{ij}^l = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab}^l y_{(i+a)(j+b)}^{l-1}$,

(4)

l indicates the layer, x_{ij}^l is the pixel-value vector corresponding to e_{ij} pixel, f^l is the ReLU activation function, w^l is the coefficient vector, and b^l is the bias determined by the feature map.

Thus, for an input x , the ReLU function can be indicated as

$$f(x) = \max(0, x). \quad (5)$$

- (2) Pooling: To downsample the image along the spatial dimensions (i.e., width and height). This allows dimension reduction and enables the computation process to be less intensive.

The k -th unit of the image feature in the pooling layer is expressed as

$$\text{Pool}_k = f(\text{down}(C) * W + b), \quad (6)$$

where W and b are the coefficient and bias, respectively. $\text{down}(\cdot)$ is a subsampling function:

$$\text{down}(C) = \max\{C_{s,l} | s \in Z^+, l \in Z^+ \leq m\}. \quad (7)$$

$C_{s,l}$ is the pixel value of C in e and m indicates the sampling size.

- (3) Fully connected: All the previous layer and the next layer of neurons are linked. It acts like a classifier based on the features from the previous layer.
- (4) Dropout: The neurons are randomly dropped out during the training phase. This can avoid overfitting phenomena and enhance the generalization of the neural network trained.

Figure 2(f) shows the categorization of each partitioned room to bedroom or nonbedroom after adopting the CNN method.

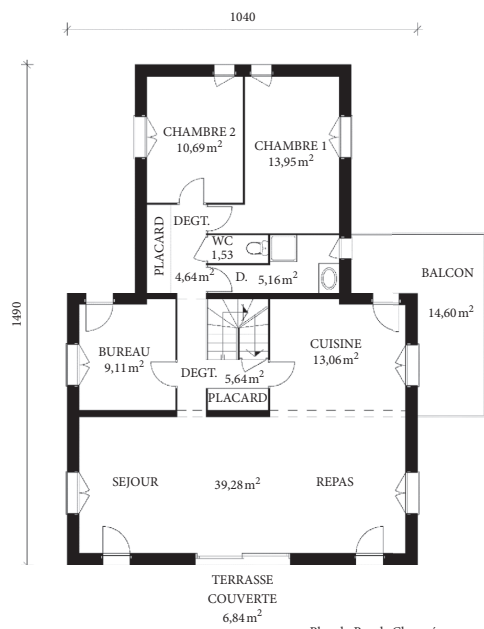
4. Dataset Description

Thus far, there are limited publicly available datasets that contain the architectural floor plan images. For instance, these four datasets: CVC-FP [26], SESYD [27], Robin [28], and Rent3D [29] usually served as the experimental data in academic studies. The detailed information of these datasets is shown in Table 1 and the sample images are illustrated in Figure 4. It is observed that the databases are largely limited on their own. Concisely, the floor plan images may vary in different aspects: (1) building types (i.e., theater, school, house, and museum); (2) multidimensional images (i.e., 2D and 3D); (3) representation types (i.e., sketches and computer-aided design); and (4) furniture layouts (i.e., walls, windows, doors, sofa, and stairs). Particularly, Rent3D and CVC-FP are the scanned images. The contents are mostly in text, rather than displaying the furniture icon. On the other hand, Robin and SESYD comprised the computer-generated floor plan with lesser image noise, compared to the other two datasets. However, the wall thickness of SESYD is relatively thin. Therefore, it may lead to some errors during the wall segmentation process. Based on the pros and cons of the datasets discussed above, only the Robin dataset is considered as the experimental data in this paper.

From the pros and cons summarized in Table 1, we opt to select the Robin dataset as the experiment data to evaluate the proposed framework. In short, the repository of the building plans (Robin) dataset can be categorized into three main classes, namely, 3 rooms, 4 rooms, and 5 rooms. The

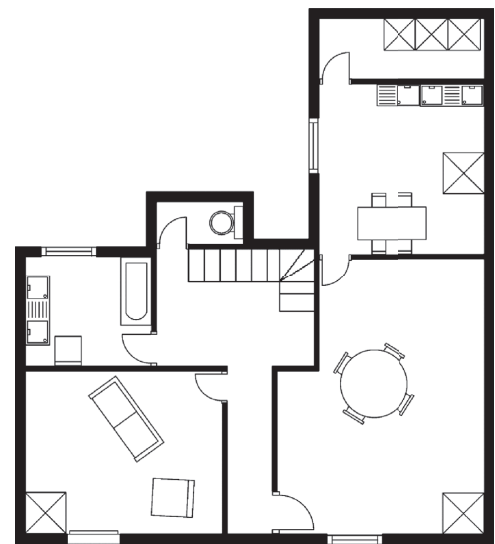
TABLE 1: Detailed information of the four floor plan databases.

	CVC-FP	SESYD	Robin	Rent3D
Presence of text	✓	✗	✗	✓
Total number of images	122	10	510	250
Item	Table	✗	✓	✗
	Chair	✗	✓	✗
	Sink	✓	✓	✓
	Toilet bowl	✓	✓	✓
	Bathtub	✓	✓	✓
	Bed	✗	✓	✗
	Cabinet	✗	✓	✗
Sofa	✗	✓	✓	
Fake/generated	✗	✓	✓	✗
Real/scan	✓	✓	✗	✓

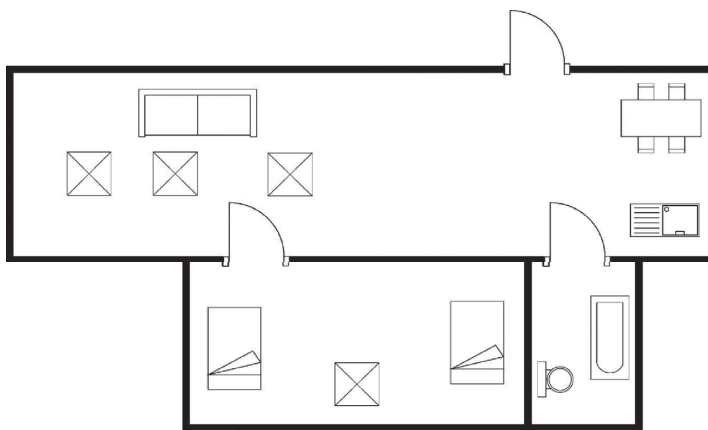


Plan du Rez de Chaussée
 échelle 1/100^e soit 1 cm pour 1 m
 alain meunier architecte d.p.l.g.

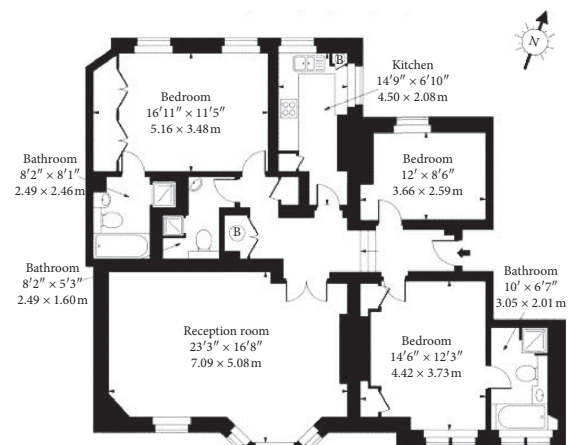
(a)



(b)



(c)



(d)

FIGURE 4: Sample images from the four datasets: (a) CVC-FP, (b) SESYD, (c) Robin, and (d) Rent3D.

images for the three classes compose an equal number of the sample which are 170 each. The item contained in the images includes table, chair, sink, toilet bowl, bathtub, bed, and sofa. The spatial resolutions for the images are varied, with the average sizes of 1085×1115 , 1188×1280 , and 1593×1859 for 3 rooms, 4 rooms, and 5 rooms, respectively. The dataset contains both the grayscale and colored images. However, all the images are portrayed in monochromatic color, in which only actual white and black pixels are presented. The illustration of the sample images and the details of each image category are portrayed in Table 2. Note that the images shown are randomly selected from the dataset. The image data that support the findings of this study are openly available in [30].

Nonetheless, there is no textual description provided for this floor plan dataset. Since this dataset is synthetic data and the elements are aggregated automatically, some image samples may not be in accordance with the real-world scenario. For example, it can be seen that the sample image #1 looks more normal compared to that of the sample image #2. For example, the sample image #2 for the 3 rooms does not have any object/graphics in one of the rooms. As for the sample image #2 for the 4 rooms, three rooms are encircled by a big squared room. Lastly, for the sample image #2 for the 5 rooms, to enter one of the bedrooms, it has to go through another bedroom. Besides, the only route to the kitchen is to pass through a bathroom.

Surprisingly, among the 510 images in the Robin dataset, 22 images do not contain any bedroom furniture. Thus, the 22 images are regarded as a 0-bedroom category. Most of the images comprise one bedroom, namely, 391 images. The maximum number of bedrooms in the dataset is two, in which up to 97 images contain two bedrooms. An overview regarding the number of the bedroom with their respective sample images is illustrated in Table 3.

5. Experiment Results and Discussion

5.1. Performance Metric. There are two evaluation metrics to validate the performance of the proposed framework, namely, accuracy and $F1$ -score [31]. Specifically, accuracy is the most intuitive performance measure and shows how accurate and precise the result is generated. Since the number of bedrooms is inconsistent in the Robin dataset, $F1$ -score is used to tackle to imbalance class problem and to avoid the bias phenomena. Mathematically, these two metrics can be expressed as

$$Accuracy := \frac{TP + TN}{TP + FP + TN + FN}, \quad (8)$$

and

$$F1 - score := 2 \times \frac{Precision \times Recall}{Precision + Recall}, \quad (9)$$

where

$$Recall := \frac{TP}{TP + FN}, \quad (10)$$

and

$$Precision := \frac{TP}{TP + FP}, \quad (11)$$

where

- (1) TP (true positive): the model correctly classified the bedroom
- (2) TN (true negative): the outcome where the model correctly predicts that it is not a bedroom
- (3) FN (false negative): the event where the model does not predict the bedroom correctly, while in fact, it is a bedroom
- (4) FP (false positive): the test result indicates that it is a bedroom, but it is not

5.2. Classification Performance and Analysis. All the 510 floor plan images from the Robin dataset are used as the experimental data. The bedroom classification results using five different CNN architectures (i.e., AlexNet, GoogleNet, SqueezeNet, ResNet-101, and ResNet-50) are shown in Table 4. Note that the number of epochs for each CNN is set to the range of 10 to 100. The learning rates and the minibatch size are set to 0.0001 and 128. The details of the configuration of the CNN architecture and parameter values are listed in Table 5.

The three types of images are merged into a composite dataset to provide a fair training and testing experiment environment. Concisely, in the classification stage, 5-fold cross-validation is utilized to validate the performance of the transfer learning model on an unseen image sample. Thus, for each fold, $510/5 = 102$ images are treated as the testing sample, while the remaining 408 images served as the input to the architecture for parameter training (i.e., weights and biases). This process will be repeated 5 times until all the images have been tested at least one time. Note that, there is no overlapping data between the training and testing sets.

Table 4 reports one of the highest classification performances obtained among the epoch range. It is observed that the highest accuracy exhibited is produced by GoogleLeNet, in which the accuracy and $F1$ -score obtained are 98.40% and 98.80%, respectively. From Table 4, it can be seen that although ResNets are the two largest architectures among all the CNN methods, the classification results are the lowest (i.e., accuracy = 81.40% and $F1$ -score = 88%). This implies that learning the bedroom features leads to an overfitting phenomenon when a huge CNN is employed. Nevertheless, this binary classification task with limited data samples has demonstrated that the transfer learning technique is capable of training the discriminant features on small-size data.

To provide further insights into the classification performance, a detailed investigation is done regarding some correct and incorrect predicted bedroom images. Since there are a total of 510 images in the dataset, Table 6 displays partial numerical results that are randomly selected from the dataset. Among the samples, 8 cases produce the $F1$ -score of 100%, whereas 2 cases generate 66.67%. The reason for false detection may be because of

TABLE 2: Detailed information of Robin dataset [28].

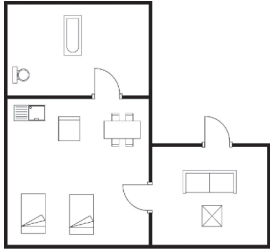
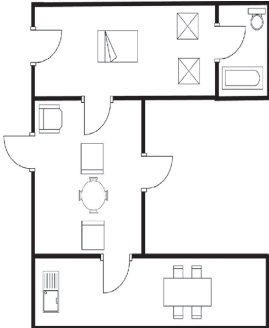
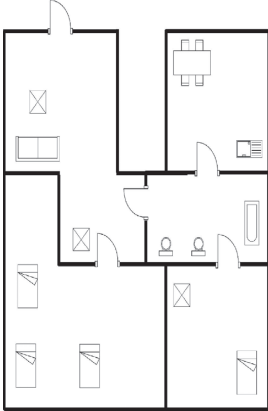

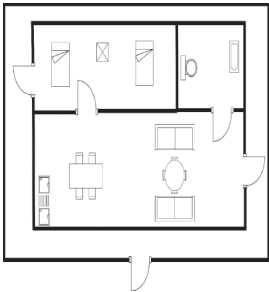
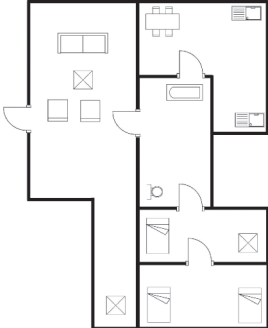
	3 rooms	4 rooms	5 rooms	
Image number	170	170	170	
Resolution	(Average) (Minimum) (Maximum)	1085 × 1115 603 × 1194 1204 × 1244	1188 × 1280 793 × 1228 1444 × 2050	1593 × 1859 1116 × 2365 2404 × 2380
Sample picture (#1)				
Sample picture (#2)				

TABLE 3: The number of bedroom in the Robin dataset [28].

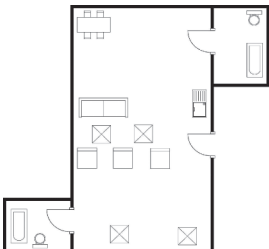
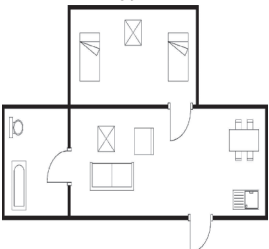
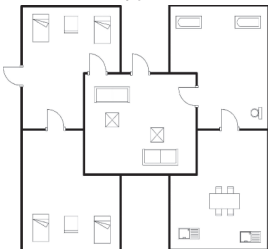
	0 bedrooms	1 bedroom	2 bedrooms
Image number	22	391	97
Sample picture			

TABLE 4: Performance % of the bedroom classification when utilizing five convolutional neural networks, in terms of accuracy (Acc) and F1-score (F1)

Method	Epoch	Acc	F1
AlexNet	60	90.5	97.5
GoogLeNet	30	98.4	98.8
ResNet-50	60	81.8	88
ResNet-101	30	81.8	87.2
SqueezeNet	70	96.6	97.9

TABLE 5: Configuration type and parameters of CNN models.

Configuration type	Parameter
Optimizer	SGDM
Momentum	0.9000
InitialLearnRate	0.0001
LearnRateSchedule	Piecewise
LearnRateDropFactor	0.2000
LearnRateDropPeriod	10
L2Regularization	0.0001
GradientThresholdMethod	L2 norm
GradientThreshold	Inf
MaxEpochs	[10, 100]
MiniBatchSize	128
Verbose	1
VerboseFrequency	50
ValidationPatience	Inf
Shuffle	Once
ExecutionEnvironment	Auto

TABLE 6: Detailed analysis of ten floor plan samples with GoogLeNet.

Image	TP	TN	FP	FN	F1-score (%)
Rob_001	2	1	0	0	100
Rob_002	1	1	0	1	66.67
Rob_003	1	1	0	1	66.67
Rob_004	2	1	0	0	100
Rob_005	2	1	0	0	100
Rob_006	2	1	0	0	100
Rob_007	2	1	0	0	100
Rob_008	2	1	0	0	100
Rob_009	3	1	0	0	100
Rob_010	2	1	0	0	100

several factors. For instance, the insufficient training sample makes the architecture not trained well. Besides, an unusual room image is shown in sample picture #2 for 4 rooms in Table 2. Nevertheless, the accuracy and *F1*-score attained are satisfactory when applying some popular CNN architecture, such as GoogLeNet.

Aside from the numerical results, the qualitative visualization is shown in Figures 5 and 6 to provide further classification context clues. Figure 5 depicts the activations of GoogLeNet. This particular activation layer selected is allocated in the third quarter of the network. There are noticeable brighter intensity pixels when there is a bed at the respective position. Besides, the activations of ResNet for the layer at a similar location (i.e., the third quarter of the network) are shown in Figure 6 for a fair comparison. Apparently, Figure 6(a) denotes that the bottom left corner should have a bedroom, as this particular in the activation image has bright pixels. However, that location does not have any bedroom, which means that the bedroom features learned in this network are insufficiently precise.

The detailed properties of all the network architectures are tabulated in Table 7, which include the depth, size, and the number of the learnable parameter in the network. Amongst, SqueezeNet occupies the least size (5 MB) and the

fewest parameters (1 million), but the network depth is ~ 2.5 times larger than the AlexNet. The deepest network is ResNet-101, which has a depth of 347. As the largest network that contains the largest number of the learnable parameter is AlexNet size of 61 million, AlexNet is the shallowest network among the networks presented (depth size of 25).

The proposed method is capable of tackling the images with different orientations as the transfer learning architecture is designed such that the model is invariant to orientation. Besides, the architecture learns the discriminative features automatically from the floor plan images while achieving high performance. Moreover, with a limited training sample, the classification performance is promising, namely, *F1*-score = 98.8% when employing GoogLeNet. On the other hand, the epoch number required for model training is relatively few (< 100), implying that the speed of computing and execution is quick. This primarily contributed to the advantages of the transfer learning technique, as the architecture was pretrained in an ImageNet database that consists of a million images with 1000 categories.

Since GoogLeNet exhibits the best results when performing the binary classification task. We attempt to extend the work by summarizing the number of correctly guessing images. Note that, each floor plan image may contain up to 5 rooms, yet, the bedroom numbers ranged from 0 to 2. Among the 510 images, 457 images are correctly classified to their respective bedroom number. A confusion matrix is tabulated in Table 8, providing a detailed analysis regarding the classification task. It is discerned that all the 22 images that do not have bedrooms are correctly classified. However, it is also noticed that 4 images that are supposed to contain only 1 or 2 bedrooms are classified as having 3 bedrooms. Nevertheless, the proposed algorithm exhibits a promising bedroom classification of 89.61%. Moreover, the gradient-weighted class activation mapping (Grad-CAM) [32] method is utilized to highlight important regions in images that are trained by the CNN model. The interpretation of GoogLeNet model predictions using Grad-CAM is portrayed in Figure 7. These pleasing numerical and qualitative performances further verify the capability and feasibility of the framework developed.

To further verify the effectiveness of the proposed framework, the performance compared to the state of the art is tabulated in Table 9 when conducting different detection tasks on the floor plan images. Succinctly, Table 9 lists the accuracy of (a) bedroom detection, the detected rooms are correctly recognized as the bedroom; (b) room detection, the rooms are successfully being detected; and (c) room-type detection, the detected rooms are correctly classified as the room-type, such as bedroom, drawing room, and kitchen. On the other hand, nonroom-type include the parking porch, bathroom, study room, and prayer room. The overall accuracy achieved in the previously published work is quite promising, namely, $> 80\%$. Note that, methods #1-#2 evaluated the algorithms on the R2V dataset [33], method #3 evaluated on Rent3D dataset [29], and methods #4-#7 evaluated on the CVC-FP dataset [26]. The details of the datasets are summarized in Table 1. Notably, the proposed method herein is capable of



FIGURE 5: Activations of GoogLeNet when the image (a) does not contain a bed and (b) contains a bed.

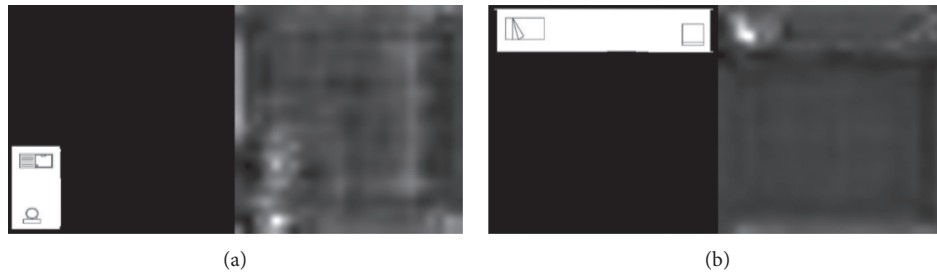


FIGURE 6: Activations of ResNet when the image (a) does not contain a bed and (b) contains a bed.

TABLE 7: The properties of the network architecture.

Method	Depth	Size (MB)	Parameter (millions)
AlexNet	25	244.66	60.95
GoogLeNet	144	29.06	6.99
ResNet-50	177	103.68	25.50
ResNet-101	347	180.34	44.44
SqueezeNet	68	5.23	1.23

TABLE 8: Confusion matrix of bedroom number calculation when utilizing GoogLeNet as the feature descriptor.

		Predicted			
		0	1	2	3
Desired	0	22	0	0	0
	1	20	355	14	2
	2	0	15	80	2
	3	0	0	0	0

detecting all the rooms and exhibiting an accuracy of 98% when detecting the bedroom.

Moreover, the recommender system can provide additional details regarding the floor plan information. In summary, a list of statistics can be generated, such as (1) the total built-up area of the floor plan; (2) the total number of bedroom and their area; (3) the total number of nonbedroom and their area; (4) the position of the bedrooms; (5) the

maximum and minimum area of each room; (6) the total number of doors; and (7) the total number of doors in each room. Taking Figure 7(a) as an example, the list of detailed statistics is tabulated in Table 10. Note that, the original floor plan in the dataset did not provide the real-world scale. Thus, the unit of the built-up area presented here is pixels². The unit (i.e., pixels²) can be easily converted to the real-world measurement (i.e., square feet, m²) by a ratio proportion.

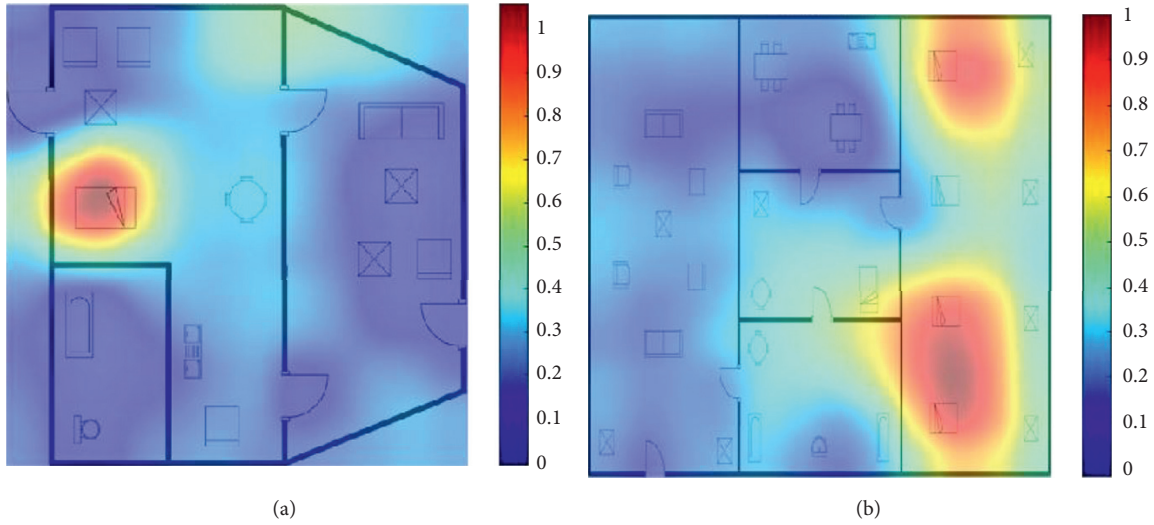


FIGURE 7: The Grad-CAM visualization to localize class-discriminative regions.

TABLE 9: Performance comparison of the bedroom detection and room detection.

No.	Method	Bedroom detection	Room detection	Room-type detection
1	[33]	0.89	—	—
2	[34]	0.83	—	—
3	[34]	0.75	—	—
4	[21]	—	0.85	—
5	[17]	—	0.89	—
6	[13]	—	0.91	—
7	[35]	—	0.86	0.94
8	Ours	0.98	1	—

TABLE 10: List of statistics generated by the recommender system.

Statistics	Values
Total built-up area	118,278 pixels ²
Total number of bedrooms	1
Bedroom's area	56,765 pixels ²
Total number of nonbedrooms	2
Nonbedroom's area	[46629, 14884] pixels ²
Position of the bed	West
Maximum area of room	56,765 pixels ²
Minimum area of room	14884 pixels ²
Total number of doors	4
Number of doors in each room	[0, 3, 3]

6. Conclusion

This paper presents a novel framework to automatically identify the location and the number of bedrooms from the floor plan images. Some traditional and data-driven image processing techniques are applied. In brief, Otsu's thresholding and morphological operations are employed to preprocess the image. Then, the rooms are extracted using the Hough transform and FAST algorithm. Finally, some popular convolutional neural network architectures are utilized to determine if the detected room is the bedroom. The quantitative performance results suggest that the proposed pipeline is feasible in recommending the house property from

architectural floor plan images. Particularly, an excellent bedroom classification accuracy of 98.4% and *F1*-score of 98.8% are achieved when employing the state-of-the-art deep learning techniques. Moreover, the visual presentation regarding the cues of the correctly detected bedroom category further verifies the effectiveness of the approach.

As future work, this framework can be extended to recognize other function rooms, such as the bathroom, dining room, or living room. Besides, the binary classification task and the limited data samples point towards the studies on the new design of the shallow neural network. Note that, some previously published works investigated the floor plan with both the text and graphics. However, the experiments presented in this work did not consider the images with text, and this points towards an important direction for future research. Therefore, it is also worth investigating the optical character recognition (OCR) technique on other types of floor plan images.

Data Availability

The image data that support the findings of this study are openly available in <https://github.com/gesstalt/ROBIN>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by Ministry of Science and Technology (MOST) (Grant nos. MOST 109-2221-E-035-065-MY2, MOST 108-2218-E-035-018-, and MOST 108-2218-E-009-054-MY2).

References

- [1] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "Nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.
- [2] H. Lee, H. Ho, and Y. Zhou, "Deep learning-based monocular obstacle avoidance for unmanned aerial vehicle navigation in tree plantations," *Journal of Intelligent and Robotic Systems*, vol. 101, no. 1, pp. 1–18, 2021.
- [3] K.-H. Liu, Q.-S. Jin, H.-C. Xu, Y.-S. Gan, and S.-T. Liong, "Micro-expression recognition using advanced genetic algorithm," *Signal Processing: Image Communication*, vol. 93, Article ID 116153, 2021.
- [4] E. V. Lavrukhin, K. M. Gerke, K. A. Romanenko, K. N. Abrosimov, and M. V. Karsanina, "Assessing the fidelity of neural network-based segmentation of soil xct images based on pore-scale modelling of saturated flow properties," *Soil and Tillage Research*, vol. 209, Article ID 104942, 2021.
- [5] Williamsburg, "Why are floor plans not used in real estate marketing?" 2020, <https://mrwilliamsburg.com/why-are-floor-plans-not-used-in-real-estate-marketing/>.
- [6] J. Bayer, S. S. Bukhari, and A. Dengel, "Interactive design support for architecture projects during early phases based on recurrent neural networks," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, pp. 27–43, Springer, Funchal, Madeira, Portuga, January 2018.
- [7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [8] G. Egor, S. Sven, D. Martin, and K. Reinhard, "Computer-aided approach to public buildings floor plan generation. magnetizing floor plan generator," *Procedia Manufacturing*, vol. 44, pp. 132–139, 2020.
- [9] C. Liu, J. Wu, and Y. Furukawa, "Floornet: a unified framework for floorplan reconstruction from 3d scans," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 201–217, Munich, Germany, September 2018.
- [10] A. Phalak, V. Badrinarayanan, and A. Rabinovich, "Scan2-plan: efficient floorplan generation from 3d scans of indoor scenes," arXiv preprint arXiv:2003.07356, 2020.
- [11] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3d: a large photo-realistic dataset for structured 3d modeling," vol. 2, no. 7, arXiv preprint arXiv:1908.00222, 2019.
- [12] J. Chen, C. Liu, J. Wu, and Y. Furukawa, "Floor-sp: inverse cad for floorplans by sequential room-wise shortest path," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2670, Seoul, Korea, October 2019.
- [13] L.-P. de las Heras, S. Ahmed, M. Liwicki, E. Valveny, and G. Sánchez, "Statistical segmentation and structural recognition for floor plan interpretation," *International Journal on Document Analysis and Recognition*, vol. 17, no. 3, pp. 221–237, 2014.
- [14] R. Khade, K. Jariwala, C. Chattopadhyay, and U. Pal, "A rotation and scale invariant approach for multi-oriented floor plan image retrieval," *Pattern Recognition Letters*, vol. 145, pp. 1–7, 2021.
- [15] S. Dodge, J. Xu, and B. Stenger, "Parsing floor plan images," in *Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pp. 358–361, IEEE, Nagoya, Japan, May 2017.
- [16] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Improved automatic analysis of architectural floor plans," in *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, pp. 864–869, IEEE, Washington, DC, USA, September 2011.
- [17] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Automatic room detection and room labeling from architectural floor plans," in *Proceedings of the 2012 10th IAPR International Workshop on Document Analysis Systems*, pp. 339–343, IEEE, Washington, DC, USA, March 2012.
- [18] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," *Computer Vision - ECCV 2006*, Springer, in *Proceedings of the European conference on computer vision*, pp. 404–417, Springer, Graz, Austria, May 2006.
- [19] S. Ahmed, M. Weber, M. Liwicki, and A. Dengel, "Text/graphics segmentation in architectural floor plans," in *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, pp. 734–738, IEEE, Washington, DC, USA, September 2011.
- [20] S. Goyal, S. Bhavsar, S. Patel, C. Chattopadhyay, and G. Bhatnagar, "Sugaman: describing floor plans for visually impaired by annotation learning and proximity-based grammar," *IET Image Processing*, vol. 13, no. 13, pp. 2623–2635, 2019.
- [21] S. Macé, H. Locteau, E. Valveny, and S. Tabbone, "A system to detect rooms in architectural floor plan images," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 167–174, ACM, New York, NY, USA, June 2010.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in neural information processing systems*, pp. 1097–1105, Lake Tahoe, Nevada, December 2012.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [24] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size," arXiv preprint arXiv:1602.07360, 2016.
- [25] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, Boston, MA, USA, June 2015.
- [26] L.-P. de las Heras, O. R. Terrades, S. Robles, and G. Sánchez, "Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool," *International Journal on Document Analysis and Recognition*, vol. 18, no. 1, pp. 15–30, 2015.
- [27] M. Delalandre, E. Valveny, and J. Y. Ramel, "Recent contributions on the sesyd dataset for performance evaluation of symbol spotting systems," Retrieved from semanticscholar.org, Dec. 2017. 2011, <https://www.semanticscholar.org/paper/Recent-contributions-on-the-SESYD-dataset-for-of-Delalandre-Valveny/42a3d89544393fe80acb6d6c4eae0239c9c96b99>.
- [28] D. Sharma, N. Gupta, C. Chattopadhyay, and S. Mehta, "Daniel: a deep architecture for automatic analysis and retrieval of building floor plans," in *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and*

- Recognition (ICDAR)*, vol. 1, pp. 420–425, IEEE, Kyoto, Japan, November 2017.
- [29] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, and S. Fidler, “Rent3d: floor-plan priors for monocular layout estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3413–3421, Boston, MA, USA, June 2015.
- [30] Robin (repository of building plans): <https://github.com/gesstalt/ROBIN>.
- [31] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation,” in *Proceedings of the Australasian joint conference on artificial intelligence*, pp. 1015–1021, Lecture Notes in Computer Science, Springer, Hobart, Australia, December 2006.
- [32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, Venice, Italy, October 2017.
- [33] C. Liu, J. Wu, P. Kohli, and Y. Furukawa, “Raster-to-vector: revisiting floorplan transformation,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2195–2203, Venice, Italy, October 2017.
- [34] Z. Zeng, X. Li, Y. K. Yu, and C. W. Fu, “Deep floor plan recognition using a multi-task network with room-boundary-guided attention,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9096–9104, Seoul, Korea, October 2019.
- [35] H. K. Mewada, A. V. Patel, J. Chaudhari, K. Mahant, and A. Vala, “Automatic room information retrieval and classification from floor plan using linear regression model,” *International Journal on Document Analysis and Recognition*, vol. 23, no. 4, pp. 253–266, 2020.