International Journal on Document Analysis and Recognition (IJDAR)
https://doi.org/10.1007/s10032-024-00487-6

ORIGINAL PAPER

Automatic floor plan analysis using a boundary attention-based deep network

Zhongguo Xu[1] · Cheng Yang[1] · Salah Alheejawi[1] · Naresh Jha[2] · Syed Mehadi[3] · Mrinal Mandal[1]

Received: 27 May 2023 / Revised: 8 April 2024 / Accepted: 11 June 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract
Floor plan is an important communication tool between architects, construction engineers, and clients for a building project. Estimation of building features from a floor plan image is often a time-consuming task. Automatic analysis of floor plan images can significantly improve work efficiency and accuracy. A few research works have been reported in the literature on automated floor image analysis. However, the scope and performance of the existing techniques are limited. In this paper, a CNN-based technique, referred to as FloorNet, is proposed for the multiclass semantic segmentation of a floor plan. The proposed FloorNet has five modules: Encoder, Room type decoder, Room boundary decoder, Multiscale room boundary attention model and Floor classification. The proposed technique is evaluated using simple brochure type and complex architectural type floor plan images. Experimental results show that the proposed technique provides an improvement of 5–11% mIoU for semantic segmentation (for 9–11 classes) compared to the state-of-the-art techniques.

Keywords Floor plan analysis · Semantic segmentation · Deep learning · Attention mechanism · CNN

1 Introduction

Floor plans are widely used in architectural design and construction applications [1]. These drawings typically show a top-down view of architectural layouts of a building. Floor plans can broadly be divided into two categories: simple brochure type (SBT) and complex architectural type (CAT) [2]. Figure 1 shows examples of these two types.

It is laborious and time consuming for humans to manually extract various information such as room area, number of windows and baseboard length from floor plans. Therefore, automated floor plan analysis has been actively studied in the last few decades [3]. An essential task in the automatic floor plan analysis is to segment a floor plan into various regions (e.g., bedroom, living-room) with correct labels. The semantic segmentation results can be used in various applications such as three-dimensional (3D) modeling and construction cost estimation. However, the large quantity of heterogeneous information in a floor plan makes the semantic segmentation a challenging task.

The early works on computer aided floor plan analysis were based on traditional image analysis such as line detection, and region growing segmentation [4]. Most of these techniques assumed that the images are represented in vector graphics image format. Recently, with the advent of deep learning neural networks (DNN), DNN based floor plan analysis has become very popular [5]. Popular DNN architectures such as FCN [6], U-Net [7], and DeepLab [8] have been used for floor plan analysis [5]. However, the DNNs were

✉ Mrinal Mandal
mmandal@ualberta.ca

Zhongguo Xu
zhongguo@ualberta.ca

Cheng Yang
cyang11@ualberta.ca

Salah Alheejawi
alheejaw@ualberta.ca

Naresh Jha
naresh.jha@albertahealthservices.ca

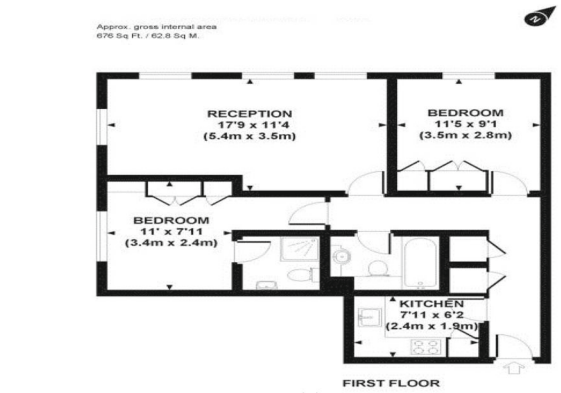
Syed Mehadi
mehadi@clinisys.ca

[1] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada

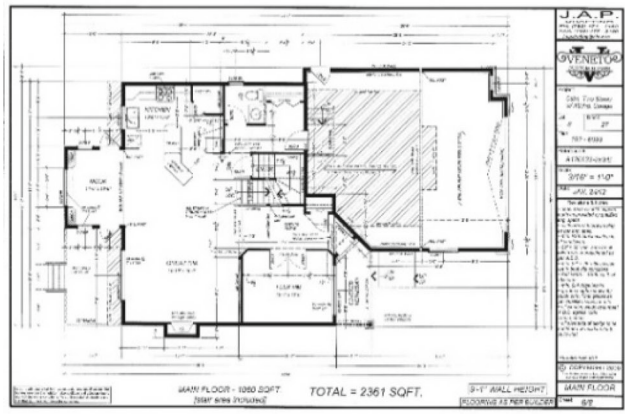
[2] Department of Medicine, University of Alberta, Edmonton, AB, Canada

[3] Clinisys EMR Inc., Edmonton, AB, Canada

Published online: 26 June 2024



(a)



(b)

Fig. 1 Examples of floor plan layouts **a** simple brochure type (SBT) drawing and **b** complex architectural type (CAT)

developed mostly for natural images, and therefore these networks may not be very efficient for analysis of floor plan images.

The objective of this paper is to propose a deep learning network, henceforth referred to as the FloorNet, which is tailored for robust semantic segmentation of both SBT and CAT floor plans. The FloorNet is an extension of our previous work [9]. It is designed as a multi-task network that has an Encoder to extract the hierarchical features from the floor plan image. These hierarchical features are processed by a room boundary decoder (RBD) and a room type decoder (RTD) to recognize the room-boundary and room-type pixels, respectively. The major contributions of this paper are as follows:

(1) The FloorNet proposes an enhanced multiscale room boundary attention model (MRBAM), which helps refine the room type pixels by suppressing the noises near the room boundaries, resulting in an improved performance.
(2) Improved RBD and RTD are developed in the FloorNet by replacing the linear interpolation upsampling method by the upconvolution. The learning process of the upconvolution is more helpful to recover the spatial details compared to the linear interpolation methods. Since the CAT floor plan is more complex, the CNN encoder is improved in the FloorNet by utilizing a deeper backbone (i.e., DenseNet) to efficiently extract the features.
(3) To the best of authors' knowledge, this is the first comprehensive study for the automatic analysis of both the SBT and the CAT floor plan images. Experimental results show that the proposed FloorNet provides a superior segmentation performance for both types of floor plans.

The organization of the paper is as follows. Section 2 presents a review of the floorplan segmentation literature. The proposed technique is presented in Sect. 3. The performance of the proposed technique is evaluated and compared with existing techniques in Sect. 4. Section 5 presents a discussion on the architecture of the proposed network, limitations of this study and future research works, followed by the conclusions in Sect. 6.

2 Related works

In this section, we present a review of literature on automated segmentation methods for floor plan images.

2.1 Traditional floor plan analysis

Macé et al. [10] studied floor plan analysis where the walls, represented by thick lines, are first extracted from the components in the vector graphic by coupling the Hough transform [11] and image vectorization. The rooms are then segmented by recursive decomposition until convex-shaped regions are found from the wall borders. However, the reported accuracy of room segmentation is low, and the detected rooms are not labeled.

Ahmed et al. [12] used an idea similar to [10] to detect rooms and introduced new ideas on wall detection. The thick and medium lines are detected as the walls, and the thin lines are considered to be parts of symbols (e.g., windows). The doors and windows are detected from the symbols using the speeded up robust features (SURF) [13]. The rooms are detected using a similar idea as [10] and are labeled using the text. Experimental results with SBT floor plans show performance improvement over [10], but the technique may not give good performance for complex construction type floor plan where all lines may not be walls or symbols.

The contextual relationship between the floorplan elements is usually not present in the vector floor plans [5]. For example, the doors or windows are usually embedded

between the wall segments and the kitchen is generally near the dining room. Analyzing a floor plan without considering their contextual relationship is error-prone, as each element can be a constraint for other objects. Deep learning-based floor plan analysis has become popular in the last decade and achieved state-of-the-art performance. The literature review of floor plan analysis using the DNN based methods is presented in the next section.

2.2 Deep learning based floor plan analysis

Most DNNs are based on convolutional neural networks (CNNs), which have become very popular in image analysis, segmentation, and classification.

Dodge et al. [14] was one of the earliest researchers to use deep learning-based methods to analyze the SBT floor plans. In their proposed pipeline, FCN [6] is used to segment the wall pixels. Yamasaki et al. [15] also presented an end-to-end FCN to analyze the apartment floor plans. Compared to the work of Dodge et al. [14], a total of 12 different classes of room types can be detected as the output of their semantic segmentation model.

Yang et al. [16] proposed a U-Net based technique for semantic segmentation of the wall and door in CAT floor plans. Experimental results demonstrated the superiority of a CNN-based approach that can handle the complex drawings. In Jang et al. [17], a DarkNet53-based encoder-decoder (DED) network was used to segment walls and doors based on CAT floor plans. In this network, the final average pooling layer, fully connected layer, and softmax layer of DarkNet53 were removed because only 2 (i.e., wall and door) classes of objects were considered in their model.

Attention mechanisms have been successfully used in various computer vision tasks, such as facial expression recognition [18], saliency detection [19], and crowd counting [20]. Typically, the operation selects the most useful features for classification and then outputs the final features by weighing the importance of attention maps and the target maps.

Zeng et al. [21] presented a method to recognize diverse floor plan elements using a deep multitask neural network where VGG is used to extract the features from the input image. The room boundary and the room type predictions are treated as different tasks in their network. A room-boundary-guided attention (RBGA) mechanism for floor plan analysis is implemented using a spatial contextual module to explore the spatial relations between the boundary and the room elements (henceforth referred to as the RBGA-CNN technique). It has been shown that the RBGA mechanism improves the overall accuracy by approximately 4%.

3 Proposed technique

Recently, we reported a CNN-based network with an efficient boundary attention aggregated model (BAAM-CNN) [9]. This model showed promising results on SBT images. The FloorNet proposed in this paper is an improvement of this work, where various modules of [9] have been enhanced and a thorough performance evaluation with both SBT and CAT datasets conducted.

In this section, the proposed FloorNet is presented. The schematic of the overall architecture is shown in Fig. 2. The architecture consists of five modules: (i) CNN encoder, (ii) Room boundary decoder (RBD), (iii) Room type decoder (RTD), (iv) Multiscale room boundary attention model (MRBAM), and (v) Floorplan classification (FC). The details of these modules are presented in the following.

3.1 CNN encoder

The floor plan analysis starts with the CNN encoder, which includes 5 convolution blocks (as shown in Fig. 2). The purpose of the Encoder module is to generate feature maps from an input floor plan image, which would then be used by next modules for the semantic segmentation. In this work, we have used VGG16 [22], ResNet34 [23] and DenseNet121[24] as candidates for the encoder backbone for extracting floor plan features. The relative performance of these architectures will be presented in Sect. 4. Table 1 shows the size of the encoder feature maps E1-E5 corresponding to the VGG16, ResNet34 and DenseNet121 architectures. Table 1 also identifies the layers (of VGG16, ResNet34 and DenseNet121) from which these feature maps are obtained.

The input images are resized to 512×512 pixels and fed into the encoder. The consecutive five convolution blocks will generate the feature maps E1, E2, E3, E4, and E5 (the size and depth of these features maps are shown in Table 1).

The feature maps E1-E5 are shared by two parallel branches, i.e., the RBD and the RTD modules which are discussed next.

3.2 Room-boundary decoder (RBD)

After the feature maps E1-E5 are extracted by the CNN encoder, the room boundary decoder uses these features to predict the room boundaries. The function of the RBD module is to generate feature maps for the room boundary. Based on the output feature maps, the classification module will classify the boundary pixels into three classes: background, wall, and door/window.

Figure 3 shows the schematic diagram of the RBD unit. The Feature-1 input refers to the features coming from the CNN encoder (E1, E2, E3, E4) and Feature-2 input refers to the intermediate learned features (B4, B3, B2) coming

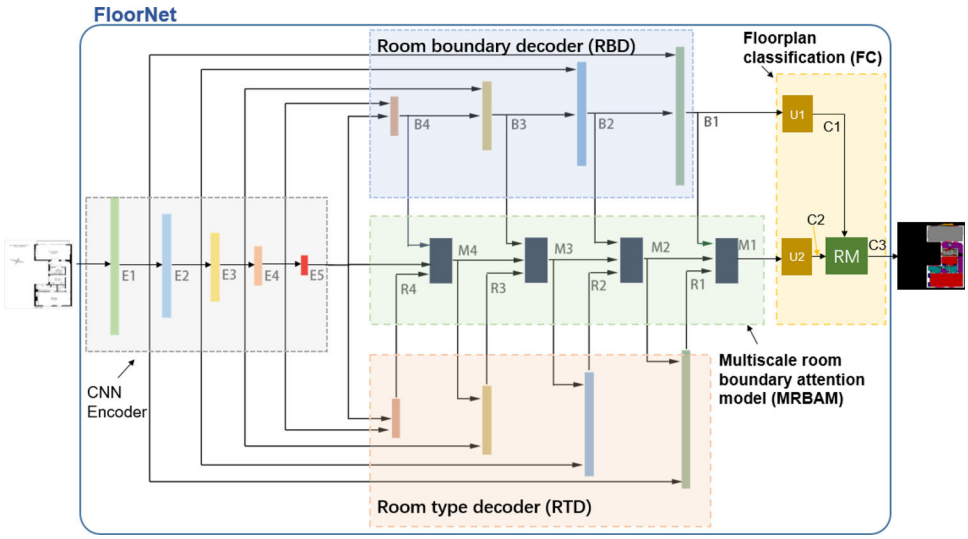


Fig. 2 Schematic of the proposed FloorNet architecture

Table 1 The size of feature maps E1-E5 (assuming an input image with size 512 × 512) in Fig. 2. Rows 3, 5 and 7 show the layers from which these feature maps are obtained from the VGG16, ResNet34 and DenseNet121 architectures

	E1	E2	E3	E4	E5
VGG16 [13]	256 × 256 × 64	128 × 128 × 128	64 × 64 × 256	32 × 32 × 256	16 × 16 × 512
O/p layer	conv3-64	conv3-128	conv3-256	conv3-512	conv3-512
ResNet34 [14]	256 × 256 × 64	128 × 128 × 64	64 × 64 × 128	32 × 32 × 256	16 × 16 × 512
O/p layer	conv1	conv2_x	conv3_x	conv4_x	conv5_x
DenseNet121 [15]	256 × 256 × 64	128 × 128 × 64	64 × 64 × 128	32 × 32 × 256	16 × 16 × 512
O/p layer	1st conv layer	Trans. layer 1	Trans. layer 2	Trans. layer 3	Dense block 4

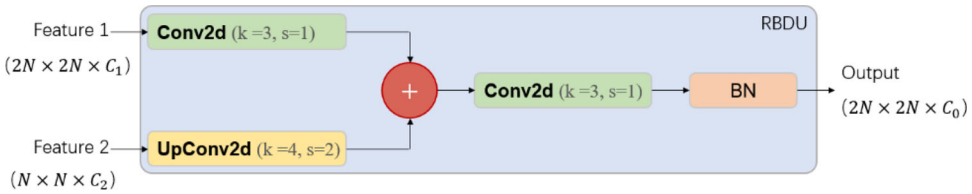


Fig. 3 Schematic of the room boundary decoder unit. The " + " means elementwise addition. C_2 equals $2C_1$ for the B4 layer and C_1 for the other layers in the decoder. C_o equals C_1 for the B4 layer and $C_1/2$ for the other layers in the decoder. BN means batch normalization. (k, s) refers to (kernel size, stride value)

from the preceding RBD unit (except for the first RBD unit for which Feature-2 is the feature map E5 from the CNN encoder). The size of the RBD unit feature maps B1-B4 is shown in Table 2.

It is observed (in Fig. 3) that the size of Feature 2 of an RBD unit is always half (in both directions) the size of Feature-1. Therefore, Feature-2 is upsampled before the addition with the convolution output of Feature-1. The summation of filtered Feature-1 and upsampled Feature-2 goes through another convolution layer to learn the features, and

Table 2 The size of various feature maps B1-B4, R1-R4, M1-M4 and C1-C3 in Fig. 2. Note that N_c denotes the total number of segmentation classes

B1/R1/M1	B2/R2/M2	B3/R3/M3	B4/R4/M4
256 × 256 × 32	128 × 128 × 64	64 × 64 × 128	32 × 32 × 256
C1	C2	C3	
512 × 512 × 3	512 × 512 × (N_c-2)	512 × 512 × N_c	

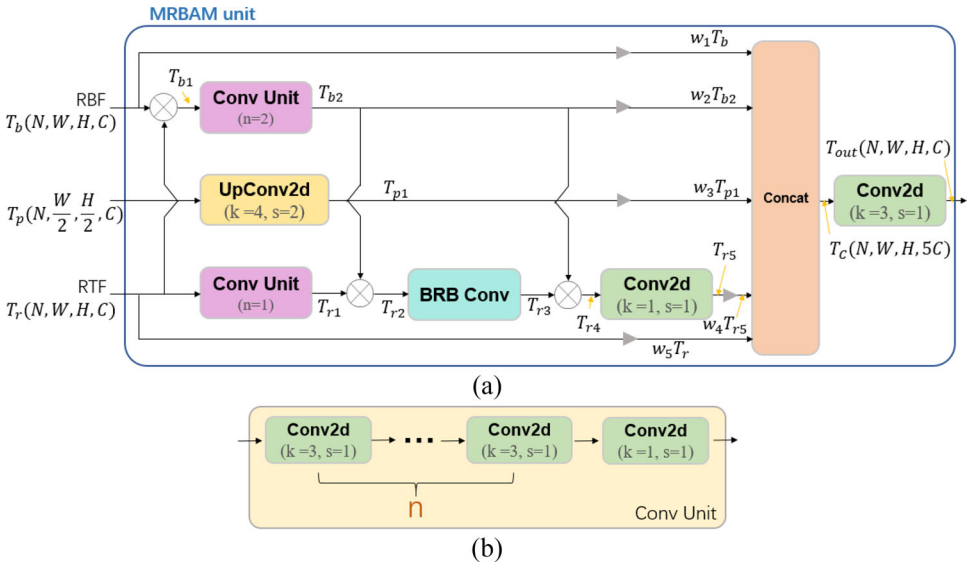


Fig. 4 Schematic diagrams of **a** MRBAM unit and **b** Conv Unit. T_b is the room-boundary feature, T_p is the output feature of the preceding MRBAM unit, and T_r is the room-type feature. N is the number of features (which is 1 in this work). The weights w_1, w_2, w_3, w_4 and w_5 are applied on the five feature maps before the concatenation operation

a batch normalization layer (denoted as "BN") is used to stabilize the learning process. Note that the UpConv2D means the upconvolution with filters of size 4×4.

3.3 Room-type decoder (RTD)

The function of the room type decoder is to predict the room type. The architecture of the RTD is similar to the RBD, and includes four RTD units. The schematic of a RTD unit is similar to that of the RBD unit shown in Fig. 3. However, unlike the RBD unit, the bottom input of a RTD unit (as shown in Fig. 2) comes from the CNN encoder and the top input comes from the preceding MRBAM unit (except for the first RTD unit for which the top input comes from the CNN encoder).

3.4 Multiscale room boundary attention model (MRBAM)

The function of MRBAM module is to combine the RBD and RTD features and perform the semantic segmentation (i.e., to predict the room type of each pixel). As shown in Fig. 2, the MRBAM has four identical units. An MRBAM unit takes the feature maps from the room boundary and room type decoders as inputs. There are four different levels in the MRBAM that processes the room boundary features and room type features at different scales. Figure 4a shows the schematic of an MRBAM unit, which has three inputs. The top input (T_b) is the room boundary feature maps coming from a room-boundary decoder and the bottom input (T_r) is the room type feature maps from a room-type decoder. The middle input (T_p) is the intermediate feature maps coming from the preceding MRBAM unit (for the first MRBAM unit this input comes from the CNN encoder).

The three inputs (T_b, T_p, T_r) pass through several modules (e.g., Conv Unit) and are finally concatenated at the Concat module. There are five inputs to the Concat module, which produces the output $T_c(N, W, H, 5C)$ where N is the number, W is the width, H is the height, and C is the number of channels. The details of each of these five inputs are presented below.

(a) The first input to the Concat module is $w_1 T_b(N, W, H, C)$

(b) The second input to the Concat module is $w_2 T_{b2}$ where

$$T_{b2} = Conv\ Unit\{T_b(N, W, H, C) \otimes T_r(N, W, H, C)\} \quad (1)$$

Note that $\otimes$ is the elementwise multiplication operator. Figure 4b shows the schematic of the Conv Unit block that stacks n 3×3 convolutions and one 1×1 convolution. Here the value of n is 2. An example of the T_{b2} visualization is shown in Fig. 5a. The grayscale feature maps are obtained by averaging the feature maps across the depth and then normalized between 0 and 255.

(c) The third input to the Concat module is $w_3 T_{p1}$ where

$$T_{p1} = Up\ Conv\ 2d\{T_p(N, W/2, H/2, C)\} \quad (2)$$

(d) The fourth input to the Concat module is $w_4 T_{r5}$ where

$$T_{r5} = Conv\ 2d\{T_{r3} \otimes T_{b2}\} \quad (3)$$

$$T_{r3} = BRB\ Conv\{T_{r1} \otimes T_{b2}\} \quad (4)$$

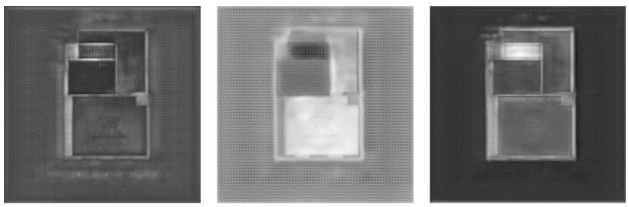


Fig. 5 An example of gray-scale visualization for feature transformation in the MRBAM unit. **a** T_{b2}, **b** T_{r1}, and **c** T_{out} refer to the feature maps denoted in Fig. 4a

$$T_{r1} = Conv\ Unit\{T_r(N,\ W,\ H,\ C)\} \tag{5}$$

An example of the T_{r1} visualization is shown in Fig. 5b. Note that the BRBConv(.) is a boundary-refinement-block convolutional layer to refine the room boundary features. The kernels are square matrices of size M ∗ M where M is an odd integer. The size of the kernel in the BRBConv is one quarter of the input feature size. A kernel example with M = 17 is shown in Fig. 6. Note that the matrix elements are ones in the horizontal, vertical and diagonal directions, and the center element is four.

(e) The fifth input to the Concat module is $w_5 T_r(N,\ W,\ H,\ C)$.

Five different inputs are concatenated by a MRBAM unit. In this work, we have used the input weights $[w_1,\ w_2,\ w_3,\ w_4,\ w_5] = [1,\ 1,\ 1,\ 7,\ 1]$ to obtain the best performance. The output T_c(N, W, H, 5C) of the concatenation layer is passed through a convolutional layer to reduce the depth from 5C toC. An example of the T_{out} visualization is shown in Fig. 5c.

3.5 Floorplan classification (FC)

The function of FC module is to predict the final floor plan semantic segmentation result based on the feature maps from the RBD and MRBAM modules.

In the FC module, the inputs B1 and M1 pass through the U modules, resulting in outputs C1 and C2, respectively. As shown in Table 2, the size of C1 is the same as that of the original input image and the depth of C1 is 3 indicating the prediction result of background, wall, or door/window. Similarly, C2 has the same size as that of the original input image and the depth of C2 is (N_c-2) indicating the prediction result of all segmentation class excluding the wall, and door/window.

The C1 and C2 represent the probability values of different pixel classes. For each pixel location, C1 provides the probability of {background, wall, door/window} classes. On the other hand, for each pixel location, C2 provides the probability of 7 (for R3D dataset) or 9 (for CAFP dataset) classes. The details about these datasets are presented in Sect. 4. Note that C2 does not provide the probabilities of the window/door and wall classes as these probabilities are provided by C1.

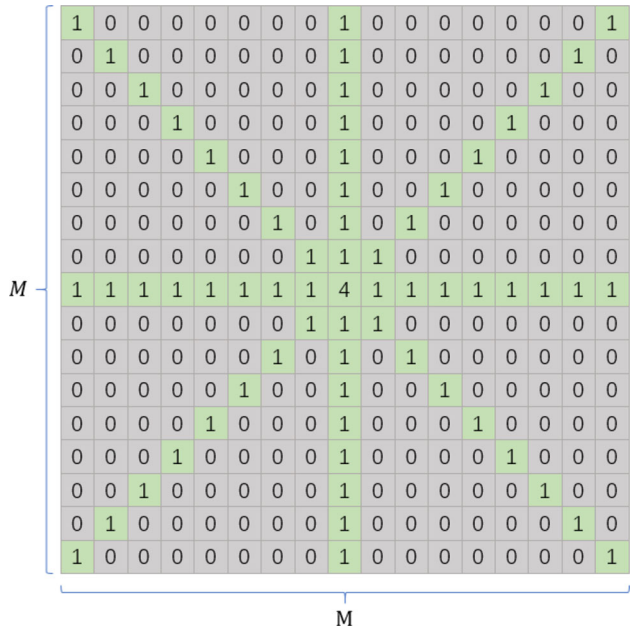


Fig. 6 Schematic diagram of BRB kernel for M = 17

The Room Merging (RM) module combines the C1 and C2 and generates the final semantic segmentation result of a pixel at (x,y) location using the following approach.

1. Consider the C1 and C2 values at (x,y) location.
2. Consider C1 first. If the pixel is of type door/window or wall, the pixel is classified as door/window or wall. If the pixel is of type background, the class of the pixel is determined by the C2 values.

4 Experiments

4.1 Datasets

In this paper, two datasets are used to evaluate the performance of the proposed technique: (1) the R3D dataset [21] and (2) the complex architecture type floor plan (CAFP) dataset. Figure 1a shows an image example from the R3D dataset and Fig. 1b shows an image from the CAFP dataset. Both datasets have pixel-wise ground truth labels for floor plan training, validation and testing. The R3D dataset has 232 images, each of size 512 × 512 pixels. In the CAFP dataset, a total of 80 floor plan images, each of size 3400 × 2200 pixels, have been obtained from the local house builder

collaborators and are manually annotated to generate pixel-wise ground truth images. The CAFP dataset is expanded eight times by use of augmentation: (i) original, (ii) rotation of original image by 90°, 180° and 270°, and (iii) up-down flipping of 4 images from (i) and (ii). The augmented CAFP dataset includes 640 images.

4.2 Network training

As shown in Fig. 2, there are 5 modules in the proposed schematic. Each of these five modules includes CNNs that require training. The weights of the CNNs in the five modules are updated (during training) to minimize the overall loss function of the whole network.

The proposed technique has two tasks, i.e., room boundary prediction and room type prediction. The contributions of the two tasks for this network are balanced by the following weighted loss function:

$$Loss = w_b L_b + w_r L_r \tag{6}$$

where L_b is the loss function for boundary prediction and L_r is the loss function for room type prediction. The weights are calculated as follows:

$$w_b = \frac{N_r}{N_b + N_r} \text{ and } w_r = \frac{N_b}{N_b + N_r} \tag{7}$$

where N_b and N_r are the numbers of boundary pixels and room pixels, respectively. In Eq. (6), the loss function for a specific task (i.e., $L_b \, or \, L_r$) is defined by:

$$L_{task} = -\frac{N - N_i}{\sum_{j=1}^{N_c}(N - N_j)} \sum_{i=1}^{N_c} (y_i \log(p_i)) \tag{8}$$

where N is the total number of ground-truth pixels, N_c is the number of classes for the task, y_i is the label for class i, and p_i is the predicted probability of class i.

The proposed network is trained on Google Colab GPU High-RAM. The Adam optimizer is used in the training process for 210 epochs. Table 3 shows the dataset setup for the R3D and the augmented CAFP in the stages of training, validation and testing. As shown in Table 3, each floor plan in R3D is segmented into 9 categories. Since the CAFP floor plans have more information than the R3D floor plans, the CAFP floor plans are segmented into 11 categories. The batch size is 1.

4.3 Performance metrics

In this paper, we use the Intersection over Union (IoU) as the metric to evaluate the semantic segmentation performance.

Table 3 The dataset for performance evaluation and the classification categories

	R3D	CAFP (Augmented)
Number of training images	160	416
Number of validation images	19	96
Number of testing images	53	128
Total number of images	232	640
Classification categories	9 categories	11 categories
	- background	- background
	- closet	- closet
	- washroom	- washroom
	- LKD* room	- LKD* room
	- hall	- hall
	- bedroom	- bedroom
	- window/door	- window/door
	- wall	- wall
	- balcony	- laundry
		- garage
		- stairs

*LKD refers to living-room/kitchen/dining-room

The IoU of class i is defined as follows:

$$IoU_i = \frac{S_I}{S_U} \tag{9}$$

where S_I is the intersection area of the predicted segmentation and the groundtruth for class i, S_U is the union area of the predicted segmentation and the groundtruth for class i. As the semantic segmentation involves more than two classes, the mIoU, as defined below, is used as the overall performance metric.

$$mIoU = \frac{\sum_i IoU_i}{N_c} \tag{10}$$

4.4 Performance evaluation

The VGGNet, ResNet and DenseNet are widely used in the literature as CNN backbones for extracting features. Table 4 shows the performance of the proposed FloorNet using VGG16, ResNet34 and DenseNet121 as the encoder module. It is observed that for the R3D dataset, the mIoU of the DenseNet121-based network is 69%, which is 10% higher than that of the VGG16 network. For the CAFP dataset, the mIoU of the DenseNet121 network is 60%, which is 9% higher than that of the VGG16 network.

Table 4 Performance of the proposed FloorNet (in Fig. 2) with VGG16, ResNet34 and DenseNet121 models in the CNN encoder module

Encoder Model	mIoU (%)	
	R3D	CAFP
VGG16	59.08	51.31
ResNet34	66.87	55.34
DenseNet121	68.65	59.88

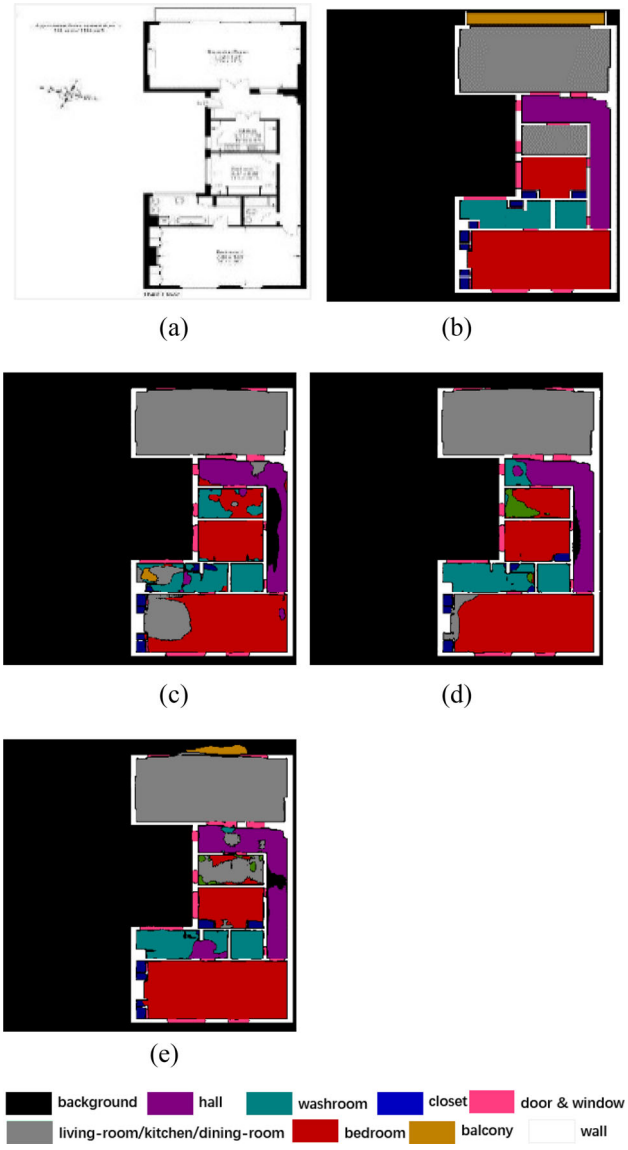


Fig. 7 Visual comparison of floor plan segmentation results produced by the proposed method for an image from the R3D dataset: **a** original image, **b** ground truth, **c** prediction of the VGG16-based network, **d** prediction of the ResNet34-based network, **e** prediction of the DenseNet121-based network

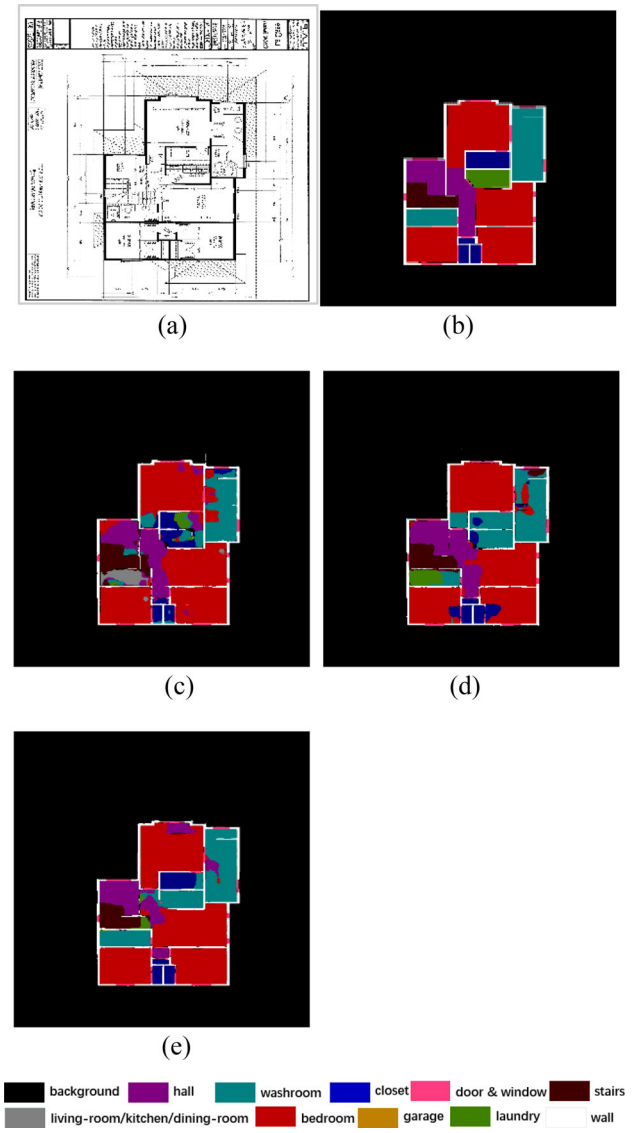


Fig. 8 Visual comparison of floor plan segmentation results produced by the proposed method based on one example from the CAFP dataset: **a** original image, **b** ground truth, **c** prediction of the VGG16-based network, **d** prediction of the ResNet34-based network, **e** prediction of the DenseNet121-based network

Figures 7 and 8 show the visual comparison of floor plan recognition results produced by our method based on the R3D and CAFP datasets, respectively. From the figures, the prediction of the DenseNet121-based network has a better performance than the VGG16-based and the ResNet34-based networks because the prediction of the DenseNet121 has less noise in large spaces.

Table 5 shows the performance comparison between the RBGA-CNN, DED and the proposed DenseNet121-based work for the R3D and CAFP datasets. For the R3D dataset, the mIoU of the proposed network is 24%, and 15% higher than that of the DED, and the RBGA-CNN, respectively.

Table 5 Performance comparison of the proposed technique with the state-of-the-art techniques DED [15], and RBGA-CNN [19]. The last row shows the performance of the proposed technique using the DenseNet121 encoder

Methods	mIoU (%)	
	R3D	CAFP
DED [17]	44.73	40.07
RBGA-CNN [21]	54.22	49.02
Proposed	68.65	59.88

When the CAFP dataset is used, the proposed technique also shows better performance than the DED and RBGA-CNN methods.

Note that because of the attention mechanism, the RBGA-CNN, and the proposed work provides a significant performance improvement over the DED model. Figure 9 shows the training loss for the RBGA-CNN, and the proposed work. Although both techniques use the attention mechanism, the proposed technique can achieve a lower loss in the training process.

Note that all experimental evaluations were performed on Google Colab GPU High-RAM environment. The inference time required for the proposed FloorNet is approximately 65–75 ms for one image, which shows relatively low computational requirement for testing environment.

5 Discussion

In this section, an ablation study of various modules proposed in the FloorNet is presented. We then discuss the reasons why the MRBAM in the FloorNet is beneficial for the room type prediction. Finally, a few limitations of the FloorNet and future works are discussed.

5.1 Analysis on the decoder and attention modules

As mentioned earlier, the objective of this work is to propose a CNN-based network that is robust for semantic segmentation of both SBT and CAT floor plan types. We first reported the BAAM-CNN in [9] for the semantic segmentation of SBT floor plans by improving the RBGA-CNN network [21]. In this paper, we propose the FloorNet by further enhancing different modules of the BAAM-CNN. In this section, we present a detailed ablation study to show the improvements caused by these modifications.

Table 6 shows the experimental results of the ablation study. FloorNet-a is a variant of FloorNet where VGG in the Encoder module of RBGA-CNN is replaced by the ResNet. FloorNet-b (i.e., BAAM-CNN) is an improved version of

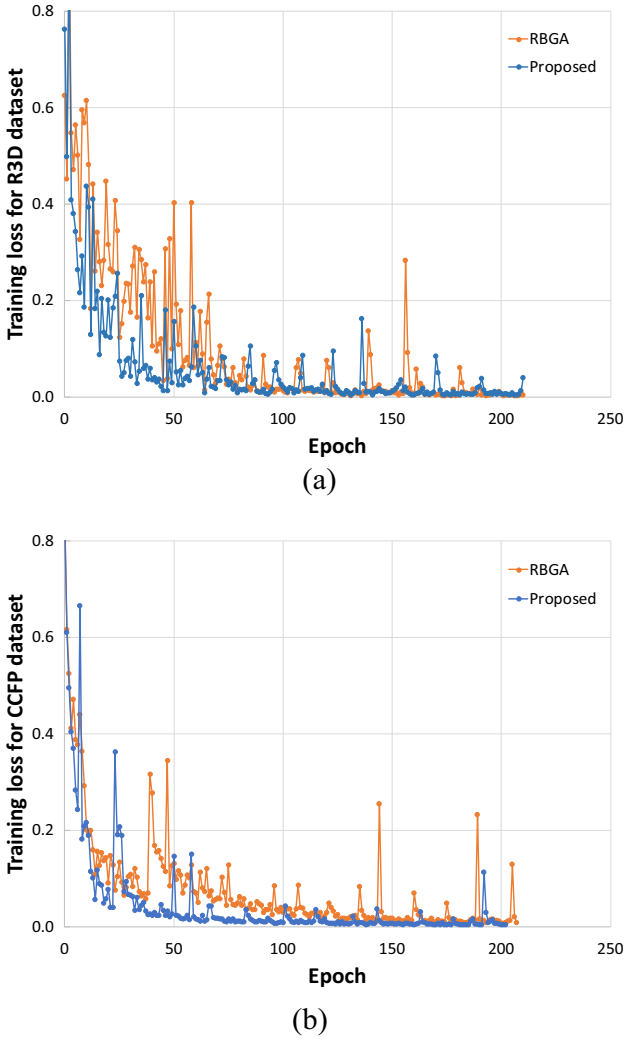


Fig. 9 Training loss of the RBGA-CNN, and the proposed DenseNet121-based technique for the **a** R3D dataset and **b** CAFP dataset

FloorNet-a where the RBGA attention module is upgraded to BAAM. FloorNet-c is an improved version of FloorNet-b where ResNet in the Encoder module is replaced by the DenseNet. FloorNet-d is an improved version of FloorNet-c where the upsampling is done by using the UpConv2d operation instead of using the linear interpolation (see Fig. 3). The proposed FloorNet includes all modifications on the CNN Encoder, RBD, RTD and the attention module. The results show that the mIoU of the network is enhanced when the new encoder, the improved decoders and attention module are introduced.

5.2 Analysis on the decoder and attention modules

The ablation study results (in Table 6) show that the attention module is beneficial for the floor plan segmentation. This section presents a qualitative analysis of the reasons.

Table 6 Ablation study on the improvements offered by the enhancements proposed in various modules based on the CAFP dataset. Note that the superscript v1 refers to the modules in [9, 21] and v2 refers to the version presented in Sect. 3.2 and 3.3 of this paper

Architecture		mIoU (%)
RBGA-CNN [21]	CNN Encoder (VGG) + RBD^{v1} + RTD^{v1} + RBGA	49.02
FloorNet-a	CNN Encoder (ResNet) + RBD^{v1} + RTD^{v1} + RBGA	50.83
FloorNet-b: (i.e., BAAM-CNN [9])	CNN Encoder (ResNet) + RBD^{v1} + RTD^{v1} + BAAM	54.21
FloorNet-c	CNN Encoder (DenseNet) + RBD^{v1} + RTD^{v1} + BAAM	54.95
FloorNet-d	CNN Encoder (DenseNet) + RBD^{v2} + RTD^{v2} + BAAM	56.45
FloorNet	CNN Encoder (DenseNet) + RBD^{v2} + RTD^{v2} + MRBAM	59.88

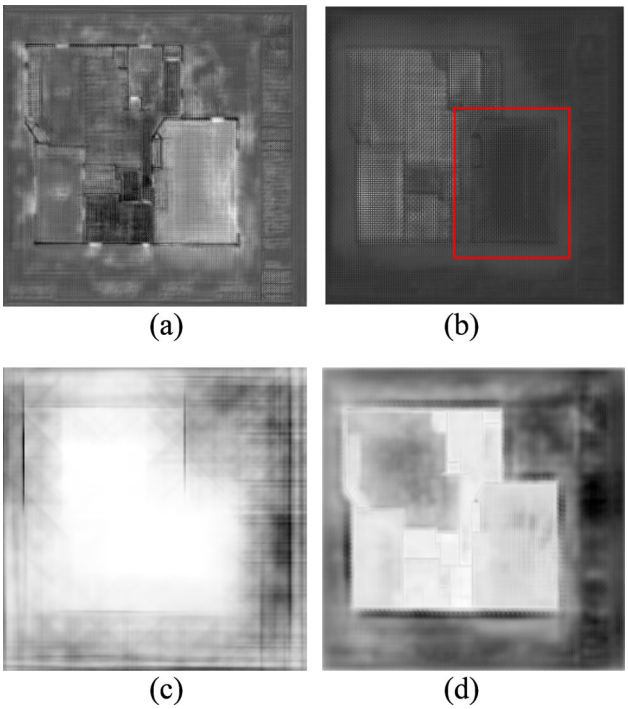


Fig. 10 An example of gray-scale visualization for feature transformation in an MRBAM unit. **a** T_b, **b** T_r, **c** T_{r3}, and **d** T_{out} refer to the feature maps denoted in Fig. 4a

Figure 10 shows an example (from CAFP dataset) of gray-scale visualization for the feature transformation in the fourth MRBAM unit (MRBAM consists of four MRBAM units (shown in Fig. 2)). The grayscale feature maps are obtained by averaging the feature maps across the depth and then normalized between 0 and 255. Figure 10a shows the well-learned room boundary (RB) feature map that is an input for this MRBAM unit. Figure 10b shows the room type (RT) feature map that is another input for this MRBAM unit. In each room (e.g., the red box), the pixels in the center area differ from the pixels near the boundaries, indicating an inconsistent prediction of the room type. The MRBAM fuses the feature maps (e.g., Fig. 10a and b) from two tasks (i.e., RB prediction and RT prediction) through element-wise multiplication. The directional kernels are used to process the fused features to address the problems that the room boundaries in a floor plan are not only horizontal or vertical (see Fig. 10c). As shown in Fig. 10d, the well-predicted room boundary is helpful to suppress the noises for the room type pixels near the room boundaries, resulting in a uniform and improved room type feature map.

6 Limitations and future works

The limitations and future works of this study are summarized below.

First, floor plans, especially the CAT floor plans, typically consist of a large quantity of heterogeneous information. Resizing the input floor plan image to 512 × 512 for the CNN network would reduce the resolution of the floor plan elements. A thorough investigation into the input image size might result in better performance. However, a high-resolution image input may lead to memory issues and require a high computational requirement. Recently, ViTs have emerged as a promising method for computer vision tasks with high computational efficiency and scalability [25]. Presumably, the patch-based scheme of ViTs may help mitigate the memory issue and high computational requirement. The segmentation of a floor plan may be improved using ViTs with a high-resolution input.

Second, the MRBAM module is developed without considering the contextual information among channels. Prior researches [26–28] found that the channel contextual information (CCI) could significantly improve the semantic segmentation performance. A CCI module can be developed to fine tune the feature maps from the CNN Encoder.

7 Conclusions

In this paper, an efficient technique, namely FloorNet, is proposed by developing a multiscale room boundary attention model (MRBAM). The FloorNet starts with an enhanced encoder by implementing the DenseNet121. The output feature maps of the encoder are shared by two simultaneous branches, i.e., the room boundary prediction and the room type prediction. The MRBAM combines the room boundary features and the room type features at different scales. Each MRBAM unit uses the well-predicted room boundary features to fine tune the room type features. The learned feature of a MRBAM unit is passed to the next-level convolution layer, through which the room type prediction is improved by the attention mechanism. The proposed technique is evaluated using two types (SBT and CAT) of floor plan images. Experimental results have shown that the proposed technique can achieve a superior performance compared to the state-of-the-art methods for both floor plan types.

Acknowledgements We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) (Grant number EGP 543740-19).

Author contributions All authors contributed to the manuscript. The project conceptualization was done by Mrinal Mandal, Naresh Jha and Syed Mehadi. Material preparation, and data collection were done by Zhongguo Xu, Cheng Yang, and Naresh Jha. Methodology development, experimental evaluations and analysis were done by Zhongguo Xu, Cheng Yang, Salah Alheejawi and Mrinal Mandal. The manuscript was written by Zhongguo Xu and reviewed by Mrinal Mandal. The funding was acquired by Mrinal Mandal and Syed Mehadi. The research was supervised by Mrinal Mandal. All authors read and approved the final manuscript.

Declarations

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

References

1. McGuire, R.H., Schiffer, M.B.: A theory of architectural design. J. Anthr. Archaeol. **2**(3), 277–303 (1983)
2. Kim, S., Park, S., Kim, H., Yu, K.: Deep floor plan analysis for complicated drawings based on style transfer. J. Comput. Civ. Eng. (2021). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000942
3. Kemble Stokes, H.: An examination of the productivity decline in the construction industry. Rev. Econ. Stat. **63**(4), 495 (1981)
4. Zhengda, L., 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. **567**, 58–74 (2021). https://doi.org/10.1016/j.ins.2021.03.032
5. Pizarro, P.N., Hitschfeld, N., Sipiran, I., Saavedra, J.M.: Automatic floor plan analysis and recognition. Autom. Constr. **140**, 104348 (2022)
6. Shelhamer E., Long J., Darrell T. Fully convolutional networks for semantic segmentation, In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 640–651 (2017)
7. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III, pp. 234–241. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
8. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII, pp. 833–851. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
9. Xu Z., Yang C., Alheejawi S., Jha N., Mehadi S., Mandal M. Floor plan semantic segmentation using deep learning with boundary attention aggregated mechanism. In *4th International Conference on Artificial Intelligence and Pattern Recognition* (2021)
10. Mac´e S., Locteau H., Valveny E., Tabbone S. A system to detect rooms in architectural floor plan images. In *9th International Workshop on Document Analysis Systems* (2010)
11. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (1972). https://doi.org/10.1145/361237.361242
12. Ahmed, S., Weber, M., Liwicki, M., Langenhan, C., Dengel, A., Petzold, F.: Automatic analysis and sketch-based retrieval of architectural floor plans. Pattern Recognit. Lett. **35**, 91–100 (2014). https://doi.org/10.1016/j.patrec.2013.04.005
13. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008). https://doi.org/10.1016/j.cviu.2007.09.014
14. Dodge S., Xu J., Stenger B. Parsing floor plan images. In *5th IAPR International Conference on Machine Vision Applications* (2017)
15. Yamasaki T., Zhang J., Takada Y. Apartment structure estimation using fully convolutional networks and graph model. In *the ACM Workshop on Multimedia for Real Estate Tech* (2018)
16. Yang J. Jang H., Kim J. Kim J. Semantic segmentation in architectural floor plans for detecting walls and doors. In *11th International Congress on Image and Signal Processing* (2018)
17. Jang, H., Kiyun, Y., Yang, J.: Indoor reconstruction from floorplan images with a deep learning approach. ISPRS Int. J. Geo-Inf. **9**(2), 65 (2020). https://doi.org/10.3390/ijgi9020065
18. Fernandez P.D.M., Pena F.A.G., Ren T.I., Cunha A. FERAtt: Facial expression recognition with attention net. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2019).
19. Zhao T., Wu X. Pyramid feature attention network for saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (2019)
20. Varior R.R., Shuai B., Tighe J., Modolo D. Scale-aware attention network for crowd counting. arXiv: 1903.02025
21. Zeng Z., Li X., Yu Y., Fu C. Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In *International Conference on Computer Vision* (2019)
22. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations* (2015)
23. He K., Zhang X., Ren S., Sun J.: Deep residual learning for image recognition. In *IEEE Computer Vision and Pattern Recognition* (2016)
24. Huang G., Liu Z., van der Maaten L., Weinberger K.Q. Densely connected convolutional networks. In *IEEE Computer Vision and Pattern Recognition* (2016)

25. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017)
26. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII, pp. 3–19. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
27. Fu J., Liu J., et al. Dual attention network for scene segmentation. In *IEEE Computer Vision and Pattern Recognition* (2019)
28. Li, Z., Sun, Y., Zhang, L., Tang, J.: CTNet: context-based tandem network for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 9904–9917 (2022). https://doi.org/10.1109/TPAMI.2021.3132068

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.