Electronics, Communications and Networks
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA241322

Precise Floor Plan Recognition Algorithm Based on Keypoints

Deshuai YIN [a,1], Guanli SHI [b], Yifan CHEN [b] and Lin LUAN [b]
[a] *School of Computer and Technology, Beijing Institute of Technology, China*
[b] *Software Engineering, R&D Center, Qingdao, China*
ORCiD ID: Deshuai YIN https://orcid.org/0000-0002-6531-5340, Guanli SHI https://orcid.org/0009-0007-6871-5933,Yifan CHEN https://orcid.org/0009-0005-1437-0840, Lin LUAN https://orcid.org/0009-0003-2268-5207

Abstract. Vectorization of floor plans (VFP) is an object detection task that involves the localization and recognition of different structural primitives in 2D floor plans(FP). The output of VFP can be further processed for the purpose of plan reconstruction, 3D reconstruction and automatic furniture layout. So far how to make the existing 2D floor plans vectorized faces the problem of recognition inaccuracy and inefficiency. This paper proposed a floor plan recognition algorithm based on key points, which is meaningful and useful. First, the algorithm identifies the effective subject of the FP with the help of the object detection algorithm; then, it builds a deep backbone network to identify the key points and semantic information of the marked plane elements; finally, the algorithm utilizes the post-processing algorithm to optimize and retrieve vectorized data information. Compared with existing methods, the algorithm adopted in this paper enhances the support for the recognition of elements such as sloping walls and bay windows, and effectively improves the recognition accuracy.

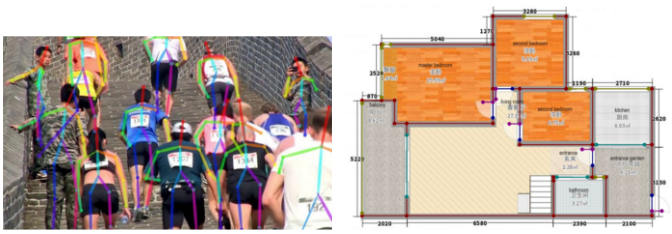
Keywords. Key Points, Floor Plan Recognition, Semantic Information, Object Detection, Home Decoration Design

1. Introduction

2D floor plans serve as the foundation for interior remodelling and design. However, rasterised or pictorial floor plans lack crucial information required for professional design, such as wall locations, door and window sizes, and floor plan coordinates. Retrieving vectorised information from floor plan images is a prerequisite for 3D reconstruction, house decoration design, and automatic furniture layout. Therefore, accurately retrieving the vectorized information has become an urgent problem.

Conventional approaches typically use heuristics to identify semantic graphical symbols in floor plans or employ OpenCV image processing methods to extract elements from floor plans. However, the accuracy of these approaches depends on sim-

[1]Corresponding Author: Deshuai Yin, Beijing Institute of Technology, No 16 Lengquan East Road, Haidian District, Beijing, China; E-mail: 3220185088@bit.edu.cn.



(a) Human pose estimation (b) Recognition result image

Figure 1. Human pose estimation and floor plan recognition

ple floor plans and images without or with limited noise[1,2,3]. The adaption of CNN(Convolutional Neural Network)-based image processing frameworks for recognising complex floor plans has become increasingly prevalent with the emergence and maturity of deep learning. One approach involves extracting elements of a floor plan through image semantic segmentation and instance segmentation [4,5,6]. However, this method often produces blurred boundaries, so it is difficult to distinguish accurately different elements. Another alternative method is to extract connection points and connecting lines from the floor plan through heatmap or skeleton recognition network. In fact, using skeleton network to extract human joint points to estimate 2D pose(Figure 1(a)) is similar to retrieve information of wall inflection points or connection points of different types of components in building floor plans(Figure 1(b)). This method can differentiate between different elements based on the categories of the connection points [7,8], while it can not provide thickness information. But in actual engineering applications, the location information of building elements is often more crucial than the thickness information.This is because it is easier to adjust the thickness of an element than its location when using design tools. Thus, this paper selects the optimized skeleton recognition network as the foundation of the floor plan recognition framework. The algorithm first locates the main body of the floor plan using an object detection algorithm. Then, it employs the deep skeleton neural network to identify the key points of each element in the floor plan, along with the semantic information. Finally, the vectorized resultant data is obtained through the optimization of the post-processing method. The main contributions of this paper compared to previous works are:

1. Proposes the Keypoints Association Fields (KAF) as the key point connection expression, which is more adaptive to the recognition of architectural floor plans.
2. Proposes an optimized skeleton network and respective loss function to improve the recognition accuracy.
3. Proposes optimization method with non-shortest suppression to obtain more reasonable and artistic wall lines.

2. Related Work

Recently structured geometric reconstruction of building floor plans through recognition has been an active research field in computer image processing. Early work on floor plan understanding relied on a bunch of low-level image processing and powerful heuristics. [9]introduces the idea of separating text from graphics to analyze floor plans. [2] ex-

tracts the main structures such as wall lines through morphological operations, Hough transform or image vectorization techniques. [1,10]introduce heuristic method including convex packet optimization, polygon approximation to locate accurate wall lines. These conventional image processing-based and heuristic methods often require the help of manual parameter tuning or threshold setting, so the generalization of these methods is greatly limited.

Deep learning techniques has led to significant progress in retrieving room structure, particularly in terms of generalization. [11] utilises a fully convolutional network (FCN) to detect wall pixels and subsequently employs the Faster R-CNN framework to detect elements such as doors and windows. [6] improves system performance by optimizing the loss function and adjusting the structure of the deep neural network. [12,13] implement the recognition and extraction of floor plan elements using semantic and instance segmentation frameworks, as well as multi-model federation. Building on this idea, [14] introduces the offset-guided attention (Transformer) module and channel fusion module, which further improved room semantic prediction.[15,16] utilise orientation-aware kernels and Generative Adversarial Networks (GAN) to enhance the efficiency and accuracy of image segmentation tasks. However, the extracted or segmented elements' boundaries are relatively noisy, making it challenging to distinguish the boundaries between different elements.

At the same time another research direction is underway, which employs skeleton recognition network frameworks like HourGlass [17], OpenPose [18], and Unet[8] to extract the connection points and lines in the floor plan. The category of the connection points are used to distinguish different elements. [7] utilizes CNN networks to create and extract high-level features to recognize room elements and applied integer planning to encode high-level constraints. This method achieves structured vector information output, but can not handle the condition of inclined wall lines. [19]proposes one method to learn to infer relationships between nodes by exchanging information through a convolutional message-passing neural network (Conv-MPN). BlumNet on GCD (Graph Component Detector) proposed by [20] is able to extract coarse and fine-grained skeleton features and can utilize local and global constraints in GR to form the final skeleton.HEAT (Holistic Edge Attention Transformer) proposed in [21] carries out detection of corner points and end-to-end classification of candidate edges between corner points by introducing an attention mechanism. [22] proposes to utilize the sequence prediction function of transformer to directly output a variable-length, ordered sequence of vertices for each room.[23] discusses two-stream graph neural networks to process the line segments and partitioned regions respectively. [24] reconstructs the household floor plan by outputting the corner points and the directions corresponding to each corner point.In summary, skeleton recognition networks can extract very accurate information about the location of planar graph elements and generalize better than semantic segmentation networks.

3. Dataset

In this paper, we collect and manually label a total of about 5000 floor plan images from the public web as the dataset for the experiment. we label the start point as well as the end point of the wall through the labelme[25] tool, and each section of the wall is represented by a line segment, and the elements such as windows and doors are also

Figure 2. Floor plan image annotation

labeled by the same way.Figure 2 shows one example of labeled image of the datasets. Uniquely, windows and doors are located on the walls, so the line segments labeling windows and doors need to be marked on top of the labeled walls by programming. The data in this paper is expanded to 7000 from 5000 by data augment techniques, especially those data about slanting walls and elements with less data in some categories. The data augment techniques adopted in this paper includes flipping noise injection, rotation etc. The dataset is randomly selected according to the ratio of 0.85 and 0.15 for training and testing data, so 5950 data are randomly selected as the training set and the remaining 1050 images are used as the testing set.

4. Method

Figure 3 shows the flow schematic of the key point based floor plan recognition algorithm, the whole system is mainly divided into five parts, which are the main body detection part, the backbone network part, the initial network part, the refinement network part and the post-processing part. The input of the main body detection part is the original picture, which is recognized by the object detection network to get a more accurate picture of the main body of the floor plan. The backbone, initial net and refinement net constitute the framework of floor plan skeleton recognition, the whole work flows as shown in Figure 4, where Initial is the architecture of the initial network and RS is the refinement network. The backbone network part uses a structure similar to the lightweight network Mobilenet V1, as shown in Table 1, and its input is the feature map after scaling the image gotten from main body detection part to 512*512. The output of the initial network part will have two branches, and the model structure is shown in Figure 5(a), which receives the output of the backbone network as input, and outputs the heatmap of the labeled key points and the KAF key point correlation field map respectively. The network in the refinement stage, whose input is the output of initial stage, is shown in Table 2, which is stacked from the RefinementStageBlock shown in Figure 5(b),and its output is the same as that of the network in the initial stage, which is the correction and optimization of the results of the previous step. The post-processing part is to optimize the generated key points and connecting lines of various elements.

4.1. Main Body Detection

Through the observation of a large number of floor plans, it is found that the effective floor plan region actually does not account for a particularly large proportion of the whole

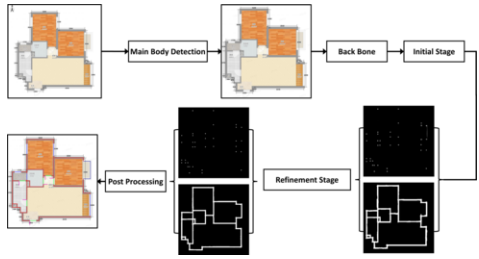


Figure 3. Overall Architecture

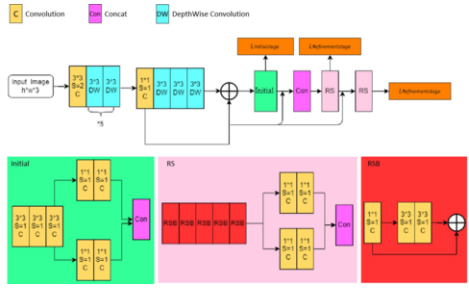
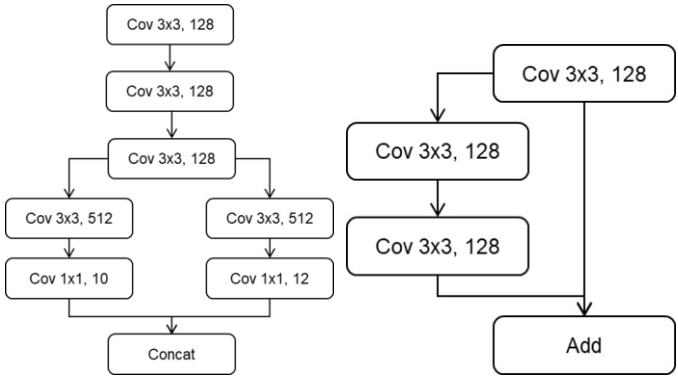


Figure 4. The architecture of recognition network

Table 1. Network architecture of Backbone

Type/Stride	Filter Shape	Input Size
Conv/s2	3x3x3x32	512x512x3
SeparableConv2D/s1	3x3x32 1x1x32x64	256x256x32
SeparableConv2D/s1	3x3x64 1x1X64x128	256x256x64
SeparableConv2D/s1	3x3x128 1x1x128x128	256x256x128
SeparableConv2D/s1	3x3x128 1x1x128x256	256x256x128
SeparableConv2D/s1	3x3x256 1x1x256x512	256x256x256
5×SeparableConv2D/s1	3x3x512 1x1x512x512	256x256x512

picture, and the effective region of some pictures accounts for even less than half of the picture, and in some cases, a large number of ineffective regions will have a great noise effect on the model, so it is necessary to detect the effective region of the input picture. Currently, most of the better object detectors used are based on deep learning networks. Considering the speed and effectiveness of detection, this paper chooses the lighter and faster YOLO model[26] as the basic network of the object detection module, and when a valid object floor plan is detected, the input image will be cropped as the input of the subsequent model for further processing. The architecture of the main body detection is



(a) Initial stage network architecture (b) Element of refinement stage block

Figure 5. Network of initial and element of refinement stage

Table 2. Refinement Stage Network

Network Architecture	Input Size	Output Size
ConCat	256x256x22 256X256X128	256x256x150
RefinementStageBlock1	256x256x150	256x256x128
RefinementStageBlock2	256x256x128	256x256x128
RefinementStageBlock3	256x256x128	256x256x128
RefinementStageBlock4	256x256x128	256x256x128
RefinementStageBlock5	256x256x128	256x256x128

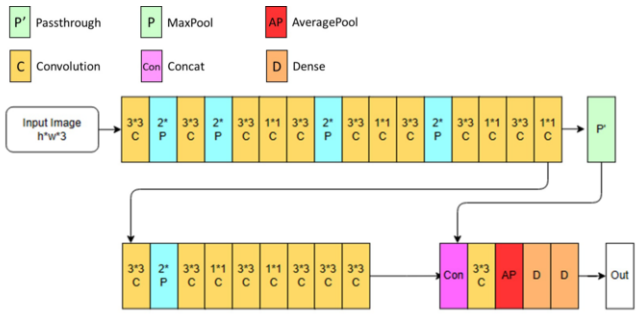


Figure 6. Main body detection

shown in Figure 6.

4.2. Key Point Identification and Extraction

key point-based floor plan recognition is actually similar to the human skeleton posture detection problem, and the detection methods for the latter can be divided into two main categories, that is, one is the top-down method, which detects the human body first and

then estimates the posture of a single person; the other is the bottom-up method, which detects the key points of the human body first and then connects them to form the human skeleton based on the detected joints.MultiPoseNet [27] points out that the bottom-up method is faster than the top-down method, and the floor plan recognition is better to make use of the bottom-up method. Therefore, this paper adopts the method of detecting the key points first and then connecting the key points. The network structure used in this paper is similar to the lightweight Lightweight-Openpose [28] of Openpose[18], and the size of the feature map is chosen to be 1/2 rather than 1/4 of the original image after scaling. Considering that the locations of some key points in the floor plan are close to each other, the use of smaller grids in lieu of pixels will obviously have a significant enhancement regarding the recall rate. The neural network in this paper predicts the probability of connecting points for each smaller grid, and the output of each grid is denoted as Eq. (1):

$$C^{'}(p) = \begin{cases} C(p), & C^{'}(p) = \max_{p^{'} \in N(p)} C(p^{'}) \\ 0, & other \quad conditions \end{cases} \tag{1}$$

4.3. KAF

In Openpose, there exists a Part Affinity Fields (PAF) to label the limbs, which are 2D vectors for each limb in the body, while maintaining the positional and orientation information between the limb regions. The human skeleton detection can be done in this way is due to its specificity, where each limb of the body is connected by a fixed class of key points. For example, the left knee and the left ankle may form the calf, and the left knee and the left hip may form the thigh these body parts, and the 2D vector can then be represented as the direction vector from the left knee to the left ankle and the left knee to the left hip. Therefore, human skeleton detection can be characterized by using PAF. In contrast, in the process of floor plan recognition, there is no such thing as a key point belonging to a fixed category, and a key point belonging to one category may not be connected to a key point in another category only.If there exists a line between two key points, then this line is all 1s between them, so this paper simplifies the PAF, and does not need to keep the direction information of the limb regions, but only keeps the position information and semantic information between them (i.e., the KAF is no longer a 2dimensional vector, it is a 1dimensional data). Suppose $x_{j1,k}$ and $x_{j2,k}$ denote the two endpoints of the k^{th} wall segment, respectively, and if a point p falls within the range of the point set of this segment, then the value of $L_k^*(p)$ is 1, and for all other points, the value of $L_k^*(p)$'s is equal to 0. The point set of a line segment can be defined as the points within the range of distances of the line segment, which means that it satisfies the following condition shown in Eq. (2):

$$0 \leq v \cdot (p - x_{j1,k}) \leq l_k \quad and \quad |v_{\perp} \cdot (p - x_{j1,k})| \leq \sigma_l \tag{2}$$

where, $v = (x_{j1,k} - x_{j2,k}) / \left\| x_{j1,k} - x_{j2,k} \right\|$, and σ_l is used for the pixel-level distance, $l_k = \left\| x_{j1,k} - x_{j2,k} \right\|$, $v_{\perp}$ is orthogonal to v.

4.4. Multi-tasking Loss

In this paper, the output of Initial and Refinement modules are supervised using the key point labeling map and KAF map of structural elements. Total loss is the sum of Initial module and Refinement module as shown in Eq. (3).Both Initial and Refinement have two branches of output composition, the key point output heatmap and the key point composition output KAF, where the weight of KAF output is two, see Eq. (4).

$$L = L_{Initialstage} + L_{Refinementstage} \tag{3}$$

$$L_{Initialstage} = L_{Refinementstage} = L_{heatmap} + 2 * L_{kaf} \tag{4}$$

4.4.1. Key Point Regression Loss

The set $C = (C_1, C_2, \cdots, C_k)$ represents the heatmaps of the K categories, $C_k \in R^{w \times h}$,w,h the output of the feature maps respectively the width and height respectively, see Eq. (5).

$$L_{heatmap} = \sum_{q} \sum_{k}^{K} (\|C_k(q) - C_k^*(q)\|_2^2 \cdot weight\_mask_{k,q}) \tag{5}$$

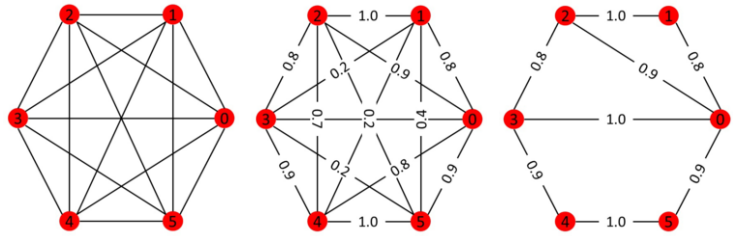
In Eq. (5), $C_k^*(q)$, $C_k(q)$represent the real heatmap labeled by category k at position $q \in R^2$and the predicted heatmap, respectively. The $weight\_mask_{k,q}$ represents the weight of category k at position q. For each category, the corresponding heatmap $C_k^*(q)$ is generated. $x_{i,k} \in R^2$ represents the position of the i^{th} key point of the k^{th} category. The value of position$p \in R^2$ in $C_k^*(q)$ is defined as Eq. (6),

$$C_k^*(q) = \max_i exp(-\frac{\|q - x_{i,k}\|_2^2}{\sigma^2}) \tag{6}$$

where σ controls the spread of the peak. In order to distinguish between peak and peak-edge values, the maximum value of a single heatmap is used in this paper as the true heat map value. Because of the relatively large proportion of the background when generating the labels for the heatmaps, the positive and negative samples will be unbalanced. In order to mitigate the problem caused by sample imbalance, the loss function needs to be weighted. The label value of the heatmap is a matrix with values ranging from 0 to 1. Let the label value be multiplied by a parameter w (with value 10) plus 1, which becomes a range of values from 1 to $w+1$. Samples with value 1 correspond to a weight of $w+1$, samples with value 0.5 correspond to a weight of $0.5 * w + 1$, samples with value 0 correspond to a weight of 1.

4.4.2. KAF Loss

Cross-entropy loss [29] is commonly used as a loss function for semantic segmentation and classification tasks to measure the degree of discrepancy between the true distribution of the data and the predicted probability distribution, the smaller the value the smaller the discrepancy and the more accurate the model. However, the standard cross entropy lacks the ability to discriminate pixels located in the vicinity of different cate-



(a) Connection relationship (b) Connection strength (c) Final result

Figure 7. Graph matching

gories, so this paper proposes to increase their corresponding weights for pixel points in the vicinity of multiple categories. y_i is the label of the i_{th} pixel, and $p_i(c)$ denotes the predicted probability of the i_{th} pixel under the category c. w_i denotes the weight of the pixel point i, which corresponds to a larger weight if the pixel point is in the vicinity of multiple categories. During the training process, due to the imbalance of positive and negative samples, the pixel points of negative samples are randomly selected according to the Bernoulli distribution in each iteration to ensure the balance of positive and negative samples. The loss function of KAF is defined as Eq.(7).

$$L_{kaf} = \sum_{i}\sum_{c}^{C} -w_i y_i log p_i(c) \tag{7}$$

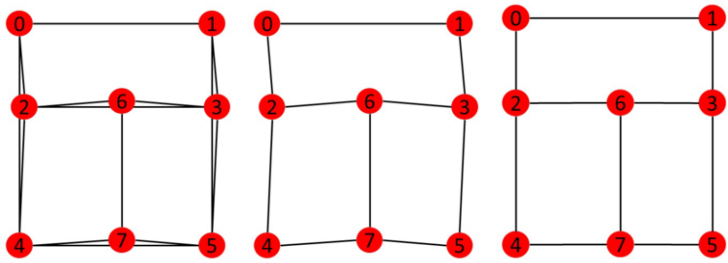
4.5. Post Process

For each channel category, this paper will get a series of candidate point sets based on the heatmap values obtained from the prediction according by applying NMS(Non-Maximum Suppression). For the category of wall, this paper will obtain a set of candidate points based on the wall channel, and then determine the connection relationship between each point and all other points to form a completely un-directed graph, as shown in Figure 7(a). For each edge in this graph, the connection strength of each edge needs to be calculated, and the connection strength is obtained based on the result of using the integration of the line segments over the KAF, and the result is shown in Figure 7(b). The result after filtering according to the threshold is shown in Figure 7(c). The line segment integral is calculated as Eq. (8):

$$E = \int_{t=0}^{t=1} L_k(p(t)) \cdot \frac{d_{j2} - d_{j1}}{\left\| d_{j2} - d_{j1} \right\|} dt \tag{8}$$

$$p(t) = (1-t)d_{j1} + t d_{j2} \tag{9}$$

where d_{j1}, d_{j2} represents two candidate keypoints position, $p(t)$ interpolates the position of the two keypoints d_{j1} and d_{j2}, $L_k(P(t))$ is the product of $L_k^*(P(t))$(the KAF value) and the unit vector of $p(t)$, see Eq. (9). Of course there may be a large number of re-



(a) with redundant lines (b) with longer lines (c) Final result

Figure 8. Posting processing

dundant line segments, as shown in Figure 8(a). For example, node 1 is connected to node 3, node 3is connected to node 5, and node1 is connected to node 5. This connecting relationship between node 1 and node 5 is redundant. In this paper, we will perform a non-shortest suppression of these connectivity relations, i.e., if two line segments have an identical point and the angle between the two line segments is less than a certain threshold value θ_1, then the shorter line segment will be retained. Of course, this threshold can be dynamically adjusted, if the longer of the two line segments is less than the length ρ_1, this threshold θ_1 will be adjusted to θ_2. If the distance between the two line segments within the threshold range is less than the threshold ρ_2 and the minimum value of the angle of the longer line segment to the vector $(0,1),(0,-1),(-1,0),(1,0)$ is less than θ_3, then the longer line segment is retained as shown in Figure 8(b). In this paper, $\theta_1 = 25, \theta_2 = 5, \rho_1 = 20, \rho_2 = 4, \theta_3 = 5$. For the consideration of aesthetic design of floor plans, most of the floor plans will have more orthogonal line segments, and the obtained line segments need to be optimized. If $\left|\cos(\bar{p_j p_i}, \bar{p_i p_k})\right|$ is larger than a threshold, these three points are co-linear, and p_j, p_k are the two connection points of p_i . As in Figure 8(c), nodes 4, 5, 7, 2, 3, 6, etc. should be colinear. If the point p_i has and only has two connecting relations p_j,p_k, p_i will be removed and p_j,p_k will be co-linear. When there are more than 2 connecting relationships, this point needs to be retained.

5. Experiments and Results

The model in this paper was trained on NVIDIA GeForce RTX 3090 GPUs for a total of 100 rounds of training. The model is optimally trained using the Adam [30] optimizer with an initial learning rate of $1 \times e^{-3}$, decaying to $1 \times e^{-4}$. The input image is first filled with the edges of the short edges according to the size of the long edges, and the fill value is the maximum value of the image pixels, and then the image is scaled to a uniform size of 512*512.In this paper, three evaluation metrics, recall, precision and F1, are used to compare the results. TP represents correct prediction, positive sample. TN means prediction is correct, sample is negative. FP means prediction is wrong, the sample is predicted to be positive but is actually negative. FN means wrong prediction, the sample was predicted to be negative, but it was actually positive. Recall = TP/(TP+FN) reflects the probability that the model correctly classifies all samples that are actually positive. Precision rate Precision = TP/(TP+FP) reflects the degree to which samples classi-

fied as positive are guaranteed to be correct. F1 = 2*Precision*Recall/(Precision+Recall) combines precision and recall to indicate the comprehensive performance of the model. The comparison methods are Raster-to-Vector[7] and Lightweight-Openpose [28], because the Raster-to-Vector method implies the Manhattan assumption, so in this paper, we only compare the straight wall these orthogonal household types in the comparison process.The original model of Raster-to-Vector has a part about semantic segmentation, while the data annotation in this paper does not have this information, so this part of the Raster-to-Vector model is not processed. And Lightweight-Openpose (abbreviated as LWO) needs to predict the direction information between key points, so in this paper, when constructing its corresponding dataset, the direction vector of the wall line is the point closer to the origin pointing to the point farther from the origin. For key points, if a predicted point has the minimum distance to some target point and this distance value is less than some threshold τ_α, then this predicted point is correct. The distance used here is the Euclidean distance. If the two endpoints of a line segment obtained from the prediction are both less than some threshold value τ_α from the two endpoints of the target line segment and the angles of the two line segments are similar, then this predicted line segment is correct. Table 3 and Table 4 represent the comparison of the quantization results for the inflection points and vectorized line segments of the wall doors and windows, respectively (bold font marks the best results).

Table 3. Comparison of results of inflection point experiments

	Wall Inflection			Door Inflection			Window Inflection		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
R2V	0.97	0.9	0.93	0.91	**0.83**	0.87	0.88	0.75	0.81
LWO	**0.99**	0.95	0.97	0.95	0.82	0.88	0.86	0.82	0.84
Ours	0.98	**0.98**	**0.98**	**0.96**	**0.83**	**0.89**	**0.9**	**0.87**	**0.89**

Table 4. Comparison of experimental results for vectorized line segments

	Wall Line			Door Line			Window Line		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
R2V	0.94	0.88	0.91	0.91	0.82	0.86	0.87	0.76	0.81
LWO	0.93	0.93	0.93	0.94	0.81	0.87	0.84	0.8	0.82
Ours	**0.98**	**0.97**	**0.97**	**0.96**	**0.83**	**0.89**	**0.91**	**0.87**	**0.89**

From Table 3, it can be seen that this paper's method is slightly inferior to the LWO algorithm in terms of the accuracy of the wall inflection points, but it is superior to other methods in all other indicators. For the vectorization results, it can be seen from Table 4that this paper's method outperforms the other methods in all the indicators.

We also performed a comparison of the results under different PCKh thresholds, as shown in Tables5 and Table 6, where s denotes the normalized distance of PCKh.

We also compared the comparison of mAP and mAR results after using different RefineStages, see Figure 9. There is a significant improvement in the results from the 1st to the second stage, and from the 2nd to the 4th stage there is an improvement, but it is

Table 5. Comparison of keypoint AP results at different PCKh thresholds

	AP@s=0.1	AP@s=0.2	AP@s=0.3	AP@s=0.4	AP@s=0.5
Wall	0.41	0.82	0.95	0.98	0.98
Window	0.38	0.65	0.83	0.88	0.9
Door	0.43	0.75	0.91	0.94	0.96

Table 6. Comparison of AP results for line segments at different PCKh thresholds

	AP@s=0.1	AP@s=0.2	AP@s=0.3	AP@s=0.4	AP@s=0.5
Wall line	0.26	0.61	0.86	0.94	0.98
Windows line	0.24	0.44	0.7	0.84	0.91
Door line	0.28	0.58	0.77	0.9	0.96

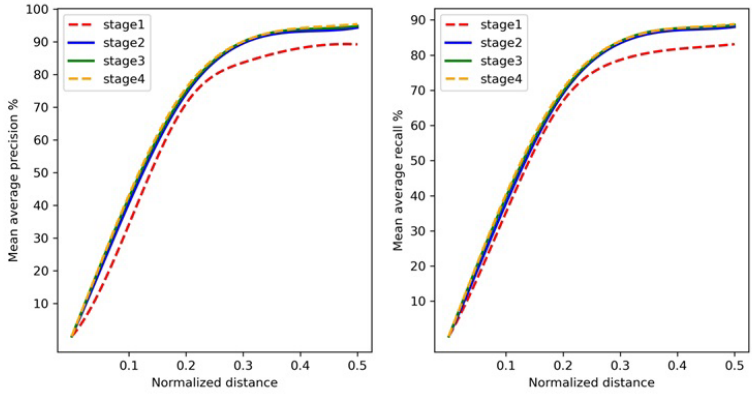


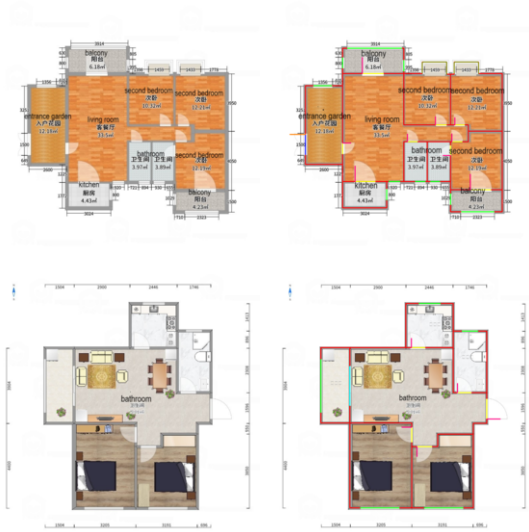
Figure 9. mAP and mAR curves at different PCKh thresholds

very small, considering the fact that the more stages we use, the more parameters and computations of the model, so we chose 2 RefinementStages in our experiments.

Some of the recognition results for different house plans in different styles and with or without the presence of furniture are presented in Figure 10, with the original images on the left and the recognized result images on the right. Categories such as bay window and door orientation have been added to the list. It can be seen from Figure 10 that the algorithm proposed in this article can accurately identify the house type even when the household appliances and furniture are found in the floor plan.

6. Conclusion and Outlook

In this paper, we propose a framework for automatic floor plan recognition based on key point detection, which uses 2D floor plan as input, detects the main body of the floor plan to reduce a large amount of background information, and then outputs the key point



(a) Original image (b) Recognition result image

Figure 10. Floor plans and corresponding floor plan recognition results

information of the main body of the floor plan as well as the semantic information of the walls, doors, and windows through the model, and finally obtains the vector data results based on the optimization of the post-processing of this information. This method can reduce the workload from image to editable floor plan for further reconstruction and provide the basis for 3D reconstruction of house type and furniture layout. Due to the limited nature of data annotation and the relatively large differences in the styles of house type images, the model is not very effective in recognizing some of these images. In view of this, we can subsequently optimize the model iteratively by increasing the size of the training set and the coverage of image styles with a view to alleviating this problem.

References

[1] Macé S, Locteau H, Valveny E, Tabbone S. A system to detect rooms in architectural floor plan images. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems; 2010. p. 167-74.

[2] Ahmed S, Liwicki M, Weber M, Dengel A. Improved automatic analysis of architectural floor plans. In: 2011 International conference on document analysis and recognition. IEEE; 2011. p. 864-9.

[3] Gimenez L, Robert S, Suard F, Zreik K. Automatic reconstruction of 3D building models from scanned 2D floor plans. Automation in Construction. 2016;63:48-56.

[4] Huang W, Zheng H. Architectural drawings recognition and generation through machine learning. In: Proceedings of the 38th annual conference of the association for computer aided design in architecture, Mexico City, Mexico; 2018. p. 18-20.

[5] Yamasaki T, Zhang J, Takada Y. Apartment structure estimation using fully convolutional networks and graph model. In: Proceedings of the 2018 ACM Workshop on Multimedia for Real Estate Tech; 2018. p. 1-6.

[6] Zeng Z, Li X, Yu YK, Fu CW. 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; 2019. p. 9096-104.

[7] Liu C, Wu J, Kohli P, Furukawa Y. Raster-to-vector: Revisiting floorplan transformation. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 2195-203.
[8] Surikov IY, Nakhatovich MA, Belyaev SY, Savchuk DA. Floor plan recognition and vectorization using combination unet, faster-rcnn, statistical component analysis and ramer-douglas-peucker. In: International Conference on Computing Science, Communication and Security. Springer; 2020. p. 16-28.
[9] Ahmed S, Weber M, Liwicki M, Dengel A. Text/graphics segmentation in architectural floor plans. In: 2011 International Conference on Document Analysis and Recognition. IEEE; 2011. p. 734-8.
[10] De Las Heras LP, Ahmed S, Liwicki M, Valveny E, Sánchez G. Statistical segmentation and structural recognition for floor plan interpretation: Notation invariant structural element recognition. International Journal on Document Analysis and Recognition (IJDAR). 2014;17(3):221-37.
[11] Dodge S, Xu J, Stenger B. Parsing floor plan images. In: 2017 Fifteenth IAPR international conference on machine vision applications (MVA). IEEE; 2017. p. 358-61.
[12] Lu Z, Wang T, Guo J, Meng W, Xiao J, Zhang W, et al. Data-driven floor plan understanding in rural residential buildings via deep recognition. Information Sciences. 2021;567:58-74.
[13] Lv X, Zhao S, Yu X, Zhao B. Residential floor plan recognition and reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 16717-26.
[14] Wang Z, Sun N. Offset-Guided Attention Network for Room-Level Aware Floor Plan Segmentation. IEEE Access. 2023.
[15] Zhang Y, He Y, Zhu S, Di X. The direction-aware, learnable, additive kernels and the adversarial network for deep floor plan recognition. arXiv preprint arXiv:200111194. 2020.
[16] Dong S, Wang W, Li W, Zou K. Vectorization of floor plans based on EdgeGAN. Information. 2021;12(5):206.
[17] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. Springer; 2016. p. 483-99.
[18] Cao Z, Simon T, Wei SE, Sheikh Y. Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 7291-9.
[19] Zhang F, Nauata N, Furukawa Y. Conv-mpn: Convolutional message passing neural network for structured outdoor architecture reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 2798-807.
[20] Zhang Y, Sang L, Grzegorzek M, See J, Yang C. BlumNet: Graph component detection for object skeleton extraction. In: Proceedings of the 30th ACM International Conference on Multimedia; 2022. p. 5527-36.
[21] Chen J, Qian Y, Furukawa Y. HEAT: Holistic Edge Attention Transformer for Structured Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022. p. 3866-75.
[22] Yue Y, Kontogianni T, Schindler K, Engelmann F. Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 845-54.
[23] Yang B, Jiang H, Pan H, Xiao J. VectorFloorSeg: Two-Stream Graph Attention Network for Vectorized Roughcast Floorplan Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 1358-67.
[24] Wei H, Chen X, Xie L, Tian Q. Cornerformer: Purifying instances for corner-based detectors. In: European Conference on Computer Vision. Springer; 2022. p. 18-34.
[25] Torralba A, Russell BC, Yuen J. Labelme: Online image annotation and applications. Proceedings of the IEEE. 2010;98(8):1467-84.
[26] Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 7263-71.
[27] Kocabas M, Karagoz S, Akbas E. Multiposenet: Fast multi-person pose estimation using pose residual network. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 417-33.
[28] Osokin D. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. arXiv preprint arXiv:181112004. 2018.
[29] Ke TW, Hwang JJ, Liu Z, Yu SX. Adaptive affinity fields for semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 587-602.
[30] Kinga D, Adam JB, et al. A method for stochastic optimization. In: International conference on learning representations (ICLR). vol. 5. San Diego, California;; 2015. p. 6.