Automation in Construction 173 (2025) 106023

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

# Comprehensive floor plan vectorization with sparse point set representation

Jici Xing [a,b], Longyong Wu [c], Tianyi Zeng [a], Yijie Wu [c], Jianga Shang [a,*]

[a] School of Computer Science, China University of Geosciences, Wuhan, 430074, China
[b] School of Software Engineering, Anyang Normal University, Anyang, 455000, China
[c] Faculty of Architecture, The University of Hong Kong, Pokfulam, 999077, Hong Kong Special Administrative Region of China

ARTICLE INFO

*Keywords:*
Floor plan
Dataset
Vectorization

ABSTRACT

Floor plan vectorization in complex scenarios poses significant challenges due to the intricate and diverse nature of design elements. This paper capitalizes on the inherent characteristics of architectural elements, eliminating the requirement for semantic segmentation processes. This paper proposes a method with a specially designed representation, employing a quartet of points to accurately capture a wide range of shapes with minimal parameters. Furthermore, a comprehensive dataset is proposed, consisting of large-scale images that contain a significantly higher number of elements and encompass diverse floor plan styles. Experimental results demonstrate the robust performance and substantial improvement of the proposed method in boundary delineation. These results indicate the ease of parameterization and notable practical potential of the proposed method in real-world scenes.

## 1. Introduction

Architectural floor plans are graphical representations that integrate design functions to depict the structure, distribution, and layout of the buildings. Professional engineers usually create these plans in a vectorized format, but they are often rasterized for consumer use in the final output. While these prints provide an intuitive understanding, this process removes semantic and geometric metadata, significantly hindering post-processing tasks such as model analysis and reconstruction. The vectorization process aims to detect the geometric properties of elements and identify semantic units that convey higher-level information. Fig. 1 demonstrates the purpose of this work by extracting key elements from rasterized images and converting them into well-defined primitives.

Previous studies addressed this issue through low-level image processing such as edge detection and symbol recognition [1]. However, these methods are limited in handling complex floor plans with varying levels of detail and noise. Recently, several data-driven approaches have shown promising results in automatic floor plan recognition [2–7]. These methods typically employ object detection [8] or segmentation [9,10] frameworks to recognize architectural symbols. Compared to commonly used benchmarks such as Microsoft COCO [11] and HRSID [12], floor plans exhibit distinct features, including geometry, topology, and semantics. Geometry refers to the physical structure, such as the layout of walls and windows. Topology provides the spatial relationships and connectivity among these elements. Semantics includes additional attributes such as text and icons. In addition to evaluating the accuracy of recognition metrics, automatic floor plan analysis should consider defining morphology and preserving topology.

Another barrier is the rare diversity of datasets. Acquiring floor plans is often costly and restricted. While many works have achieved promising results [13,14], existing public datasets mainly focus on residential buildings such as houses or apartments. However, the scope of construction and design is much broader. For example, commercial floor plans are generally more complex, yet these entities remain underrepresented in current research. Including such samples would broaden the scope of analysis and enable a more diverse range of data sources for academia and industry.

This study introduces a method to alleviate the challenges in complex and realistic scenarios. Our approach centers on accurate boundary representation while maintaining sparsity. We employ the point set representation [15] and analyze various instances to determine the minimum number of points required to preserve their morphology. Our findings indicate that four vertices are sufficient to locate most elements and semantic descriptions. Doors consist of arcs and lines usually requiring a segmentation module [5] or additional points along their boundary. The radius of a door is constant, and the center aligns with the arc, allowing the center and endpoints of the radius to parametrize the sectorial region. This observation enables a uniform representation of regular and most irregular elements using minimal points.

* Corresponding author.
*E-mail addresses:* jicixing@cug.edu.cn (J. Xing), wulongyong@connect.hku.hk (L. Wu), zty1@cug.edu.cn (T. Zeng), yijiewu@connect.hku.hk (Y. Wu), jgshang@cug.edu.cn (J. Shang).

https://doi.org/10.1016/j.autcon.2025.106023
Received 16 May 2024; Received in revised form 26 January 2025; Accepted 27 January 2025
Available online 27 February 2025

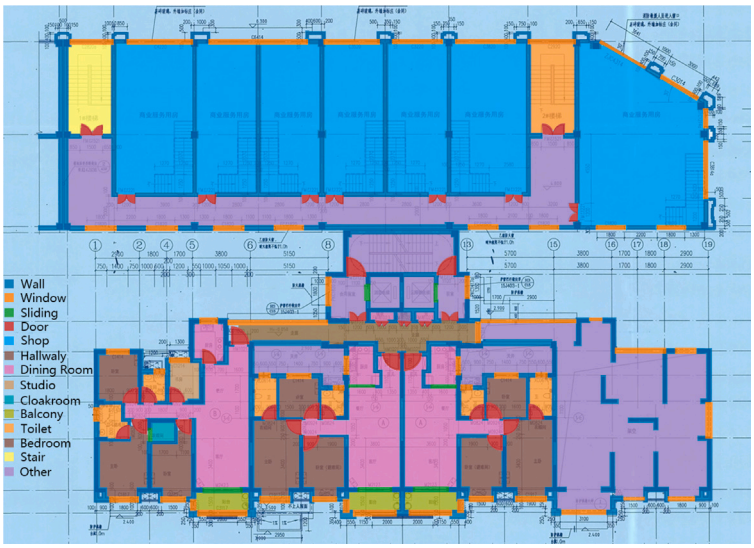



**Fig. 1.** Illustration of floor plan recognition and vectorization.

Another contribution of this work is the Comprehensive Floor Plan (CFP) dataset. Classical datasets like R2V [2,4] have been thoroughly studied, and many works have achieved satisfactory results [13,14,16]. The CFP dataset is designed to extend research boundaries by providing more diverse and complex samples. It features scans of actual floor plans, emphasizing villas, markets, and other large buildings more intricate than typical apartment or residential samples. The dataset offers instance-level annotations to support various research pipelines, empowering researchers to investigate new methods and techniques beyond the constraints of existing annotation strategies. We summarize our main contributions as follows:

1. We propose a sparse and precise representation using a quartet of points that uniformly encapsulates various architectural elements, from regular to irregular forms.
2. We introduce the CFP dataset, which includes diverse and intricate floor plan samples.
3. The sparsity and uniformity enable state-of-the-art performance on the existing CubiCasa5K [17], and the proposed CFP dataset also demonstrates superior efficiency.

## 2. Related works

### 2.1. Rule-based method

Early methods for recognizing floor plans involved locating walls, doors, and rooms by identifying graphic patterns such as lines, arcs, and contours. For instance, Ryall et al. [18] used a semi-automatic model for room identification, while Yu et al. [19] converted floor plans to vector graphics to generate 3D models. Ahmed et al. [20] separated text from graphics and extracted lines of multiple thicknesses, using thicker lines to represent walls and thinner lines for symbols. Similarly, Gimenez et al. [21] used heuristics to recognize floor plans and generate 3D models from detected elements. Despite this progress, traditional methods still require considerable effort to select suitable processing operations, tune parameters, and craft rules for various drawing styles.

### 2.2. Learning-based method

More recently, learning-based approaches have been applied to parse floor plan images.

#### 2.2.1. Semantic-based method

Semantic segmentation involves classifying each pixel in an image, focusing on specific categories rather than merely distinguishing individual boundaries or symbols. These models can accurately identify the shape and position of various elements, such as walls, doors, windows, and stairs. The extracted data can be converted into vector graphics for developing BIM and related applications. Yamasaki et al. [3] trained a fully convolutional network [22] to classify pixels into several categories, using the classified pixels to form a graph for retrieving rooms with similar structures. Zeng et al. [4] applied the semantic segmentation approach and used room boundary features to guide pixel prediction. Lu et al. [14] utilized the VGG-16 [23] to extract features and the U-Net [24] to segment architectural elements. Additionally, the SSD [25] was employed to detect text for differentiating room types. However, these pipelines may produce smeared effects and unclear boundaries. Even regions with similar features or textures may represent entirely different rooms, posing a challenge for these models in accurately identifying such areas [7].

#### 2.2.2. Object-based method

Object detection identifies symbols using predefined bounding boxes. Wang et al. [26] employed the YOLOv3 model [27] to detect elements and used a decision tree to classify different types of rooms. Lv et al. [13] integrated multi-modal information from floor plans, including room structure, type, symbols, text, and scale, to identify and reconstruct building layouts. They utilized multiple YOLOv4 models [28] to recognize room types, sizes, text, numbers, and symbols. Khade et al. [29] introduced a scale-invariant algorithm for removing doors and windows, segmenting walls, and outputting floor plan shapes while using Faster R-CNN [30] to detect 12 categories of furniture objects. However, since bounding boxes define objects based on their center and height–width offset, their recognition capability is limited to horizontal or vertical elements. This representation is inadequate for accurately describing inclined, curved, or irregular structures.

#### 2.2.3. Junction-based method

Junction-based methods depend on identifying cross junctions and their types rather than directly predicting the elements. Liu et al. [2] used ResNet-152 [31] to extract features to form several semantic maps, where walls contain I, L, T, and X types, and doors are represented as straight lines with two endpoints. Other symbols are defined by their vertices. These junctions are trained with the room regions to create a multi-class semantic map. An Integer Programming post-processing

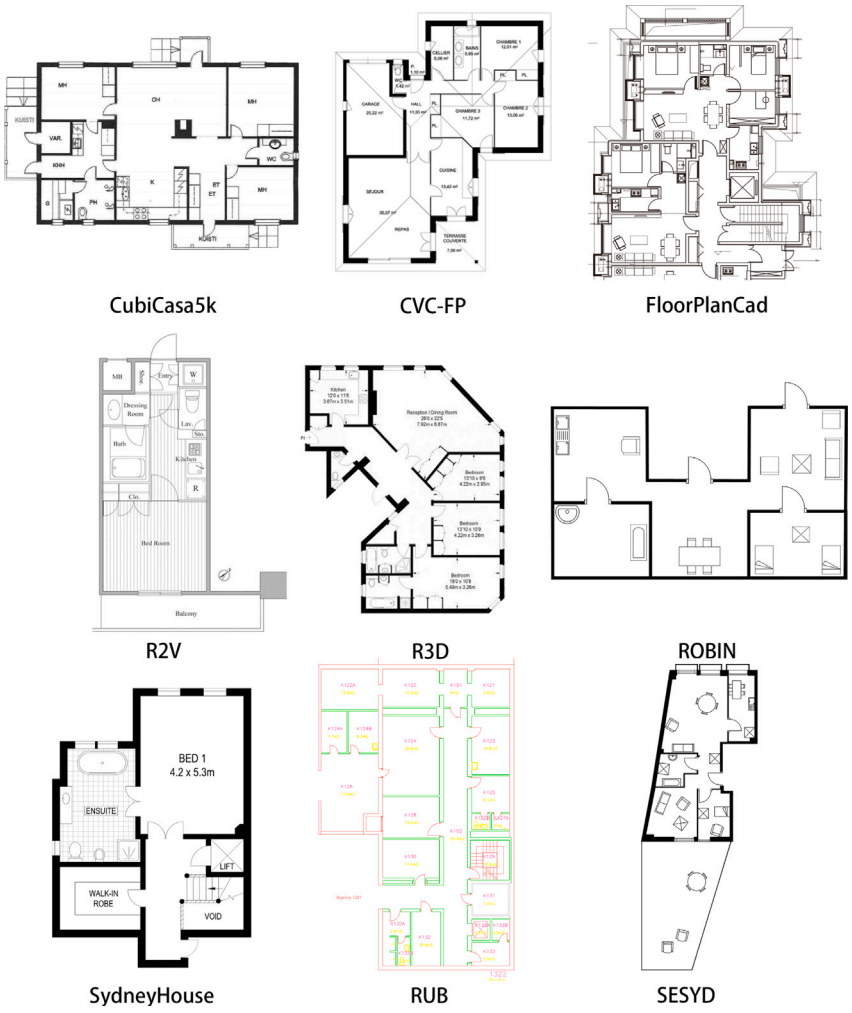



**Fig. 2.** Examples of floor plan images from existing datasets.

is designed to ensure element pairing, mutual exclusivity, and the creation of closed loops. The predicted results are further refined by aligning neighbors and eliminating parallel lines. Kalervo et al. [17] extended the work of Liu et al. [2] by proposing a modified ResNet-152 model to detect walls, rooms, and building symbols and a multi-task training strategy [32] to adjust the relative weights between the loss terms. The vectorization process also uses the Integer Programming method [2] to optimize the global topology. One issue with these methods is that they rely on the Manhattan assumption and cannot recognize irregular structures.

#### 2.2.4. Instance-based method

Instance segmentation obtains appropriate boundaries by performing mask segmentation within bounding boxes. These models have shown promising results in accurately representing individual objects. Wu et al. [5] utilized Mask-RCNN [33] to detect walls, doors, windows, and stairs and then applied constraint equations to align the elements and close gaps. Xing et al. [7] developed a morphology template that defines rules for correcting pseudo samples in self-supervised training. However, Room generation through instance-based and object-based approaches hinges on the consistency of other elements. Deficiencies like missing or misaligned components may cause spatial discontinuities, leading to incorrect room merging.

### 2.3. Dataset

Table 1 presents various datasets categorized by their origins, accessibility, annotations, and the number of samples. As shown in Fig. 2, examples from these datasets illustrate a range of styles in color, texture, room design, furniture types, and symbolic representations. However, current datasets primarily focus on residential buildings like houses and apartments. This narrow focus overlooks the broader architectural design scope. Including more diverse and complex structures would provide a richer resource for academic research and practical industry applications. Another limitation of existing datasets [2,4] is the representation of arcs, which are often depicted as simple rectangles, failing to reflect their orientation accurately. Although recent work [7] has introduced the use of Bezier curves to represent doors, this practice has not been widely adopted.

### 2.4. Application

In construction applications, identifying elements is essential for defining structural frameworks and locating key components such as doors and beams. This process is fundamental to architecture, engineering, and construction, providing crucial information for design optimization, structural analysis, and cost estimation. For example, automated recognition techniques transform two-dimensional floor plans into three-dimensional models for BIM reconstruction [49–51]. Additionally, recognized floor plans can serve as foundational designs, improving floor tile planning and promoting sustainability within the industry [52–54]. Cartography also benefits from integrating BIM, particularly in creating detailed map descriptions [55]. Furthermore, virtual and augmented reality applications are enhanced by incorporating architectural elements [56,57]. Indoor navigation systems utilize floor

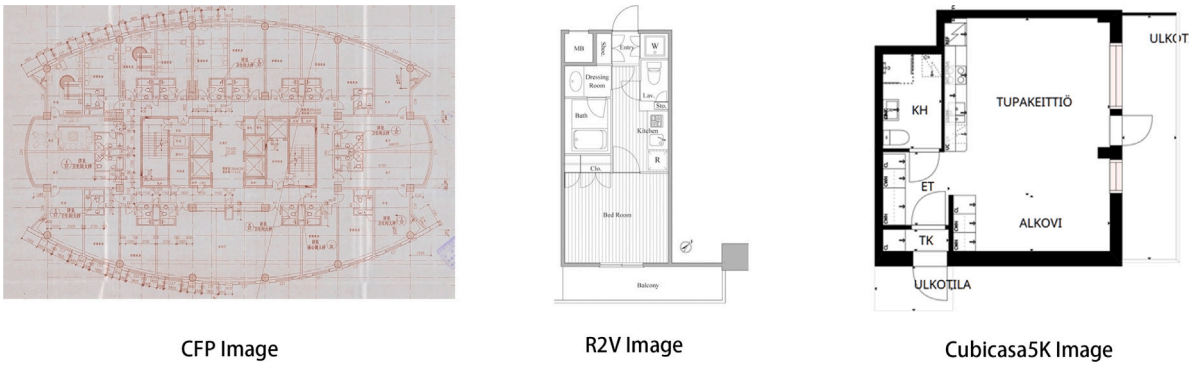



**Fig. 3.** Comparison of floor plan images from existing and CFP datasets.

**Table 1**
Summary of existing floor plan datasets.

| Dataset | Accessible | Annotation | Sample |
|---|---|---|---|
| FPLAN-POLY [34] | ✓ | wall, door window and furniture | 42 |
| SESYD [35] | ✓ | wall, door, window and 6 different furniture | 1000 |
| CVC-FP [36] | ✓ | wall, door, window, room without types | 122 |
| R3D [37] | ✓ | wall, door, window, room | 215 |
| SydneyHouse [38] | ✓ | wall, door and window in multi-unit plans | 174 |
| R-FP – Rakuten [39] | ✓ | wall | 500 |
| ROBIN [40] | ✓ | synthetic apartment room | 510 |
| ROBIN++ [40] | ✓ | hand-drawn sketch of ROBIN | 510 |
| R2V [4] | ✓ | wall, opening room | 815 |
| CubiCasa5K [17] | ✓ | wall, door, window, etc. 80 categories | 5000 |
| RPLAN [41] | ✓ | wall, room, boundaries and mask | 80 788 |
| HouseExpo [42] | ✓ | wall | 3512 |
| R3D++ [43] | ✓ | re-labeling based on R3D; adding 7 types of rooms | 215 |
| RUB [44] | ✓ | doors, non-door | 74 |
| FloorPlanCAD [45] | ✓ | wall, door, window, stair etc. 35 categories | 15,000 |
| BRIDGE [46] | ✗ | door, window and 14 other object categories | 13,000 |
| EAIS [47] | ✗ | wall, window | 450 |
| ZSCVFP [48] | ✗ | wall, room, entrance, door, window, balcony | 10,800 |
| RFP [13] | ✗ | wall, door, window, porch, room | 7000 |
| RuralHomeData [14] | ✗ | wall, door, window, stair, slope, text, room | 800 |

plans or BIM models to obtain valuable contextual information, thereby improving navigational accuracy and user experience [5,47,58].

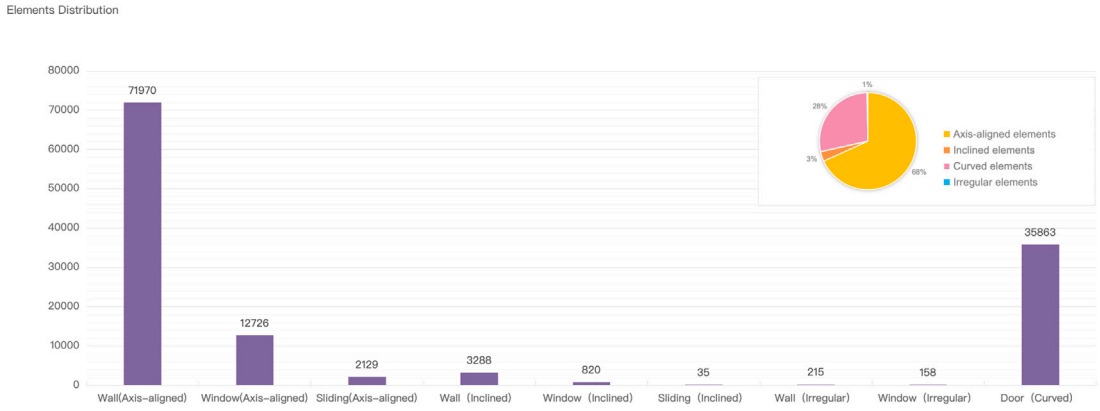
## 3. Comprehensive floor plan dataset

This study aims to alleviate the limitations of existing datasets by presenting the Comprehensive Floor Plan Dataset (CFP) dataset. This dataset contains 1062 samples of diverse floor plan styles, including villas, malls, and other comprehensive buildings. Each sample is derived from construction plans, processed by a high-resolution scanner, and covers private information such as stamps, signatures, and institutions. These scanned images are of a quality comparable to posters, which are commonly used in physical advertising campaigns. To our knowledge, this is the first dataset to include such media. The CFP dataset comprises large-scale images with a significantly higher number of elements. Additionally, the prints are often underlined, which leads to creases during handling and storage, introducing additional noise. The scanning process sometimes results in uneven placement, causing regular elements to tilt slightly, further exacerbating irregularities. Due to the complexity and detail of the source plans, annotating a single image can take between 60 and 240 min. Each sample has been meticulously annotated according to a predefined protocol, with the annotation process facilitated using the Labelme toolbox [59]. A semi-automated verification process was employed for each sample to ensure consistency and precision. Initially, the annotator reviewed any inconsistencies in the annotated floor plan. This process was followed by a secondary verification that generated rooms based on the labeled elements and manually affirmed their correct placement. Finally, the integrity of the rooms was checked to ensure all components formed a closed loop, with any errors corrected.

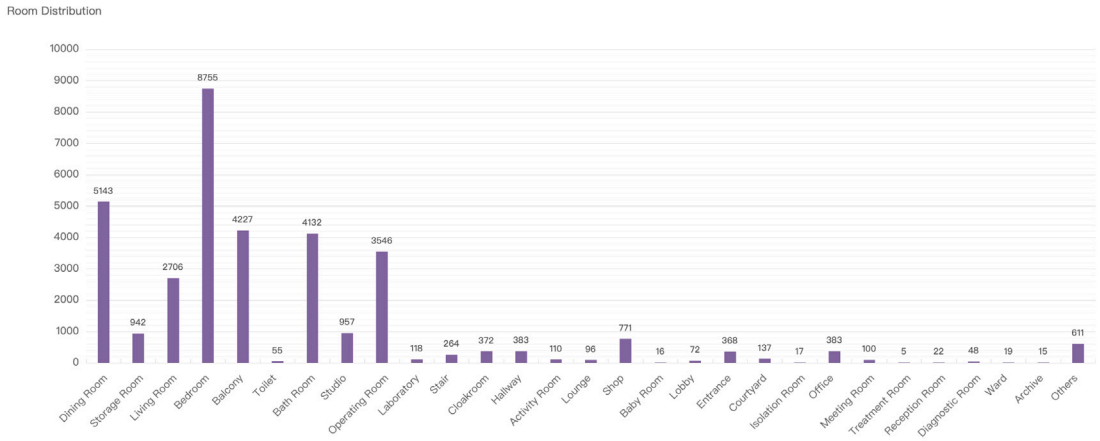
Fig. 3 showcases a representative sample from the CFP dataset alongside other existing datasets. Table 2 outlines the statistical characteristics of these datasets. The CFP dataset averages 180 elements per image, indicating a complexity level 2 to 4 times greater than its counterparts, distinguished by high resolution and dense element aggregation. Fig. 4 illustrates the distribution of elements and room types within the CFP dataset. Specifically, Fig. 4(a) depicts the distribution of various architectural elements, categorized into axis-aligned, inclined, curved, and irregular shapes. The dataset comprises approximately 68% regular and 32% irregular shapes. The higher prevalence of irregular instances in the CFP dataset compared to other datasets presents significant challenges to traditional methods that rely on the Manhattan assumption [2]. Additionally, Fig. 4(b) shows the distribution of room types, revealing an extremely uneven categorization. This long-tail distribution introduces complexity to the classification process.

## 4. Method

This section delineates the proposed method. The motivation for this work is initially explained. Subsequently, a representation strategy employing a quartet of points to parametrize diverse shapes is presented. The network architecture and associated training objectives are then detailed. Finally, the vectorization process is introduced, converting predictions into precise graphics and ensuring structural integrity and accurate room layouts.




(a) Distribution of architectural elements.




(b) Distribution of room types.

**Fig. 4.** Element and room distribution on CFP dataset.

**Table 2**
Statistical comparison of datasets.

| Dataset | Samples | Instances | Size | Density | Annotations |
|---|---|---|---|---|---|
| R2V [2] | 870 | 38,634 | 96–1,920 pix | 44 | wall, door, window, closet, bathroom, living room, bedroom, hall, balcony |
| CubiCasa5k [17] | 5,000 | 352,577 | 50–8,000 pix | 70 | wall, door, window, etc. (80 categories) |
| CFP | 1,062 | 180,693 | 4,924–12,892 pix | 180 | wall, door, window, sliding, 29 types of rooms |

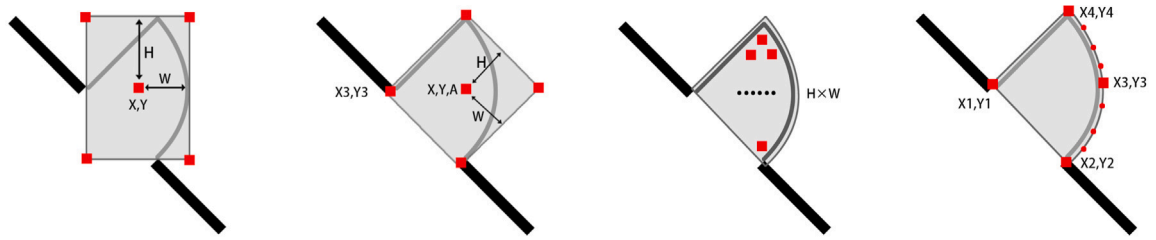
### 4.1. Motivation

The primary objective of our research is to develop a representation that accurately captures the boundaries of most elements while maintaining boosted speed. Our representation draws inspiration from the point set [15,60–62] by the following observations: Floor plans vastly consist of regular elements, interspersed with some inclined or curved ones that require specialized processing. Applying a complex procedure to all elements would introduce unnecessary computational overhead and reduce accuracy. Despite the apparent diversity in architectural drawings, they generally follow underlying rules and patterns. For instance, the interval of a door must conform to a standardized circular trajectory. Moreover, the difference between inclined and perpendicular walls is a matter of rotational adjustment. The point set method is a robust strategy for consistently representing these elements. Our approach involves selectively sampling critical points along the boundary of each element to capture their fundamental geometric characteristics.

### 4.2. Representation

The rationale of an appropriate representation is crucial in model design. Existing methods commonly base their functions on object detection or segmentation paradigms, employing common forms like bounding boxes or masks, as illustrated in Fig. 5.

The representation of the bounding box (Fig. 5(a)) is a standard practice in object detection [8], which is defined by its center along with the height–width offset. Nevertheless, this design encounters difficulties when dealing with elements possessing angular or irregular shapes. The oriented bounding box, frequently employed in aerial object detection [63], augments the conventional bounding box by incorporating an angular dimension, thereby allowing it to represent angled elements. However, it remains insufficient when depicting more complex shapes like sectorial doors. Semantic (Fig. 5(b)) and instance (Fig. 5(c)) segmentation methods utilize masks to delineate elements, providing the flexibility to encompass a diverse array of arbitrary shapes. Despite their versatility, they may yield ragged or smeared outcomes due to hollow interiors or sparse features. Moreover, instance segmentation hinges on the pre-determined bounding box, and errors




(a) Bounding box representation (b) Oriented bounding box representation (c) Semantic segmentation representation (d) Our proposed representation

**Fig. 5.** Comparison of different representation methods.

in this stage can adversely affect the subsequent masks. Meanwhile, segmenting axis-aligned elements is superfluous and might result in a reduction of precision.

This paper explores the structure of architectural forms, focusing on minimal vertices as the basis of our representation. These vertices are the cornerstones for both regular and highly irregular elements. For example, doors follow standardized circular paths, where angular span and radius variations determine the curvature. The diverse shapes of rooms present another particular challenge. To address this, elements are enclosed to reconstruct room shapes. Our approach utilizes the point set as the foundation and extends its application to include sectorial elements. The original point set involves regressing key points based on semantic features to identify objects within a specific space. The innovation lies in adapting this approach to the particular demands of architectural contexts. Our rationale focuses on minimal points as the basis of our representation. These points are the cornerstones for both regular and highly irregular elements. The following expression describes the point set function:

$$\mathcal{R} = \left\{\left(x_k, y_k\right)\right\}_{k=1}^{n} \tag{1}$$

where $\mathcal{R}$ is the point set, $k$ is the specific index and $n$ is the total number of points. The optimization process can be expressed as:

$$\mathcal{R}_o = \left\{\left(x_k + \Delta x_k, y_k + \Delta y_k\right)\right\}_{k=1}^{n} \tag{2}$$

where $\left\{\left(\Delta x_k, \Delta y_k\right)\right\}_{k=1}^{n}$ are the predicted offsets of the new points concerning the old ones. The following conversion function transforms the point set into the corresponding geometry:

$$\mathcal{C} = \mathrm{G}(\mathcal{R}) \tag{3}$$

where $\mathcal{C}$ denotes the converted shape derived from the learned point set $\mathcal{R}$. $\mathrm{G}(\cdot)$ represents the conversion function [61]. The MinAreaRect computes an object's orientation and identifies the smallest rotated rectangle encompassing the point set. The NearestGTCorner leverages the nearest ground truth point for each predicted point within a group, thereby forming the boundary. The ConvexHull determines the smallest convex polygon that encloses a set of points, ensuring that each point lies either inside the polygon or on its boundary. The ConvexHull outlines shapes requiring multiple points to delineate the boundary, thus necessitating sophisticated sampling and post-processing strategies [60]. Simple structures, such as walls and windows, can be accurately represented with only four points. Text elements are also oriented either horizontally or vertically. In contrast, complex shapes like doors require more points to describe their form accurately. However, using excessive points for all elements leads to unnecessary computational overhead. This paper proposes the TopNHull, which employs sparse points to represent most elements without sacrificing precision. The TopNHull samples $N$ points that significantly affect the area. Four points are sufficient for regular elements, while additional points do not yield further improvement. Room acquisition is achieved by enclosing other elements for irregular elements, with doors being the exception. The TopNHull is used in both the training and prediction phases. Fig. 6 illustrates the fabrics of TopNHull. During training, the Top4Hull (TopNHull with $N$ set to 4) simplifies door classes for uniform representation with other elements. At the inference stage, doors are parameterized specifically and then restored to sectors. Fig. 7 illustrates the sectorial parameterization process. The Top3Hull is used to locate the $\triangle OAB$, and the Top4Hull finds point $C$ that maximizes the area contribution, with point $C$ situated in the arc. After parameterization, sectors can be recovered, benefiting from sparse and efficient training and inference like regular elements.

The remaining task involves creating room layouts. Our method generates rooms by enclosing other elements, which requires precisely aligning these components. Any gaps or omissions could compromise the integrity of the rooms. Prior works [4,5,13,14] managed this issue through specific post-processing. However, these methods are often tailored to a particular fashion and do not transfer easily across different styles. Moreover, these approaches rely solely on fixed predictions without extra assistance. In this study, we introduce linkage points that can be generated from the endpoints of elements without incurring extra labor costs. This concept is inspired by a scene text detection research [64], which merged characters into words by predicting affinity scores between characters using a Gaussian heat map. Although a Gaussian heat map could be utilized to characterize connection relationships, this approach requires a global segmentation branch to score pixels, which is inconsistent with our sparse design philosophy. Instead, we categorize linkage points as a distinct class with fixed sizes, treating all of them uniformly regardless of their scale or shape. Our method uniformly captures architectural elements, texts, and supportive linkages, balancing precision, efficiency, and consistency. It overcomes the limitations of existing methods by accurately defining structural boundaries while minimizing computational complexity. This approach allows for consistent processing of diverse elements within floor plans, eliminating the complicated process of different shapes.

### 4.3. Network architecture

The network architecture is depicted in Fig. 8. We utilize ResNet-50 [31] with a Feature Pyramid Network (FPN) [65] as the backbone and maintain consistency with the approach of our counterpart [5]. This combination has been shown to enhance performance in recognition tasks [8]. The architecture consists of four stages, with convolutional blocks arranged in a 3:4:6:3 ratio to control the feature map dimensions and depths. Each block includes standard convolutional layers, batch normalization layers, and ReLU activation functions, all incorporating skip connections to enhance gradient flow. Specifically, given an image $I \in \mathbb{R}^{H\times W\times 3}$ with width $W$ and height $H$, the backbone network extracts a series of hierarchical features $\mathcal{I}_i \in \mathbb{R}^{\frac{H}{R_i}\times\frac{W}{R_i}\times D_i}$ for $i \in \{2,3,4,5\}$. Here, $R_i \in \{4,8,16,32\}$ represents the downscaling factor at each stage, and $D_i \in \{256,512,1024,2048\}$ denotes the number of channels. In the feature fusion stage, the deepest feature undergoes a $1 \times 1$ convolution and is progressively integrated with higher-level features through a top-down pathway:

$$\mathcal{F}_i = \mathrm{Conv}(\mathcal{I}_i) + \mathrm{Upsample}(\mathrm{Conv}(\mathcal{I}_{i+1})) \tag{4}$$

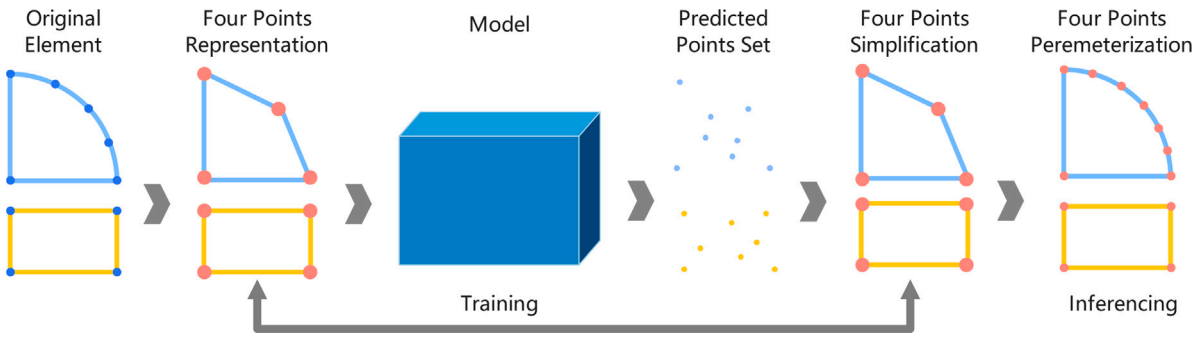



**Fig. 6.** Simplification and parameterization using TopNHull.

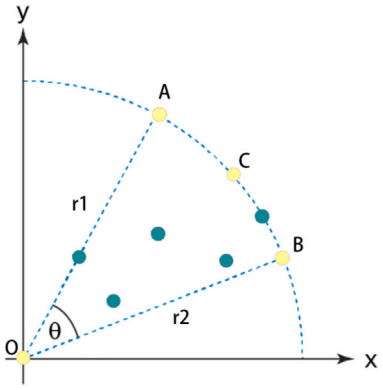



**Fig. 7.** Illustrative example of TopNHull.

where $\mathcal{F}_i$ denotes the fused feature at level $i$, and $\mathcal{I}_i$ represents the feature map output by the backbone network at level $i$. The Conv(·) operation refers to a $1 \times 1$ convolution operation, which is utilized to adjust the number of feature channels, thereby ensuring consistency across different levels. The Upsample(·) operation is the bilinear interpolation that serves to increase the spatial dimensions of the convolved higher-level feature map $\mathcal{I}_{i+1}$ to match those of the current level. The fused feature $\mathcal{F}_i$ contains comprehensive multi-scale information, thus improving the ability to perform vision tasks with higher robustness [65].

The point head corresponds to structure, linkage, and text initialization. The geometry set reconstructs building contours, the linkage set closes gaps between elements, and the text set differentiates room types. The calculation of these point sets is as follows:

$$\mathcal{R}_{init} = \text{Conv}(\mathcal{F}_i), \quad \mathcal{F}_{loc} = \text{Conv}(\mathcal{F}_i), \quad \mathcal{F}_{cls} = \text{Conv}(\mathcal{F}_i) \tag{5}$$

where $\mathcal{F}_{loc}$ and $\mathcal{F}_{cls}$ are utilized for localization and classification purposes, respectively. The point set processing unfolds in two stages. In the first stage, the initial point set $\mathcal{R}_{init} \in \mathbb{R}^{N\times 2}$ is generated via a convolutional layer, thereby yielding a preliminary prediction of the point set. The parameter $N$ denotes the number of feature points within each point set. Deformable convolution DConv(·) [66] is incorporated in the second stage to refine the receptive field. This technique augments standard convolution by learning supplementary offsets, allowing the convolutional kernel to adjust its shape and position, thereby effectively capturing intricate spatial structures. The coordinate adjustment and category determination are presented as follows:

$$\begin{aligned} &\mathcal{R}_{ref} = \text{DConv}(\mathcal{F}_{loc}, \mathcal{R}_{init}), \quad \mathcal{R}_{cls} = \text{DConv}(\mathcal{F}_{cls}, \mathcal{R}_{init}), \\ &\mathcal{R}_{loc} = \text{TopNHull}(\mathcal{R}_{cls}, \mathcal{R}_{ref}) \end{aligned} \tag{6}$$

Where $\mathcal{R}_{ref}$ denote the optimized coordinates of the point set, and $\mathcal{R}_{cls}$ represent the corresponding category. The parameter $\mathcal{R}_{loc}$ refers to the coordinates obtained through TopNHull parameterization, which selects the four points with the highest influence on the area from a set of $N$ points. This operation is uniformly applied across all feature stages $\mathcal{F}_i$. The final prediction is derived by rescaling the original size according to $R_i$.

### 4.4. Training objective

The training objective of the network comprises three components: the Focal Loss [67] for classification and the GIoU Loss [68] for initial and refined localization. The overall loss function is expressed as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{init} + \lambda_2 \mathcal{L}_{refine} + \lambda_3 \mathcal{L}_{cls} \tag{7}$$

$$\mathcal{L}_{init,refine} = 1 - GIoU \tag{8}$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ are balance coefficients that weigh the importance of each component, with values of 0.5, 1.0, and 1.0, respectively. $\mathcal{L}_{init}$ and $\mathcal{L}_{refine}$ correspond to the initial and refined localization losses, while $\mathcal{L}_{cls}$ represents the classification loss. GIoU Loss is an alternative of the IoU Loss [69] that takes into consideration non-overlapping regions to reflect the overlap degree of the predicted region $P$ and ground truth $G$:

$$GIoU = IoU - \frac{Area(C \backslash (P \cup G))}{Area(C)} \tag{9}$$

where $C$ is the minimum enclosing region containing $P$ and $G$, in the original work, $P$, $G$, and $C$ are represented as bounding boxes. In this work, they are defined as the boundary of the corresponding point set.

To alleviate the problem of aspect ratio distribution, we employed the shape adaptive strategy [62] to select different *IoU* thresholds for sampling elements. This strategy reveals the correlation between the *IoU* threshold and object aspect ratios. Specifically, a low *IoU* threshold is more effective for objects with larger aspect ratios due to the varied sensitivity of *IoU* values to localization errors for different shapes. For the $i$th ground truth element, the *IoU* threshold $\mathcal{T}_i$ is calculated as:

$$\mathcal{T}_i = e^{-\frac{\gamma_i}{\omega}} * (\mu + \sigma) \tag{10}$$

where $\gamma_i$ represents the aspect ratio of the ground truth corresponding to the prediction. The weight should decrease as the aspect ratio increases. Consequently, $e^{(\cdot)}$ is used as a monotonic decreasing function for weighting elements. $\omega$ is a hyperparameter that corresponds to the irregularity of datasets, which is set to 6. The mean and variance $(\mu, \sigma)$ of the object are also taken into consideration to weight its bias:

$$\mu = \frac{1}{J}\sum_{j=1}^{J} \mathcal{I}_{i,j}, \quad \sigma = \sqrt{\frac{1}{J}\sum_{j=1}^{J} \left(\mathcal{I}_{i,j} - \mu\right)^2} \tag{11}$$

where $J$ represents a number of predictions and $j$ is a sample from $J$ corresponding to the ground truth $i$. $\mathcal{I}_{i,j}$ is the *IoU* value between $i$ and $j$.

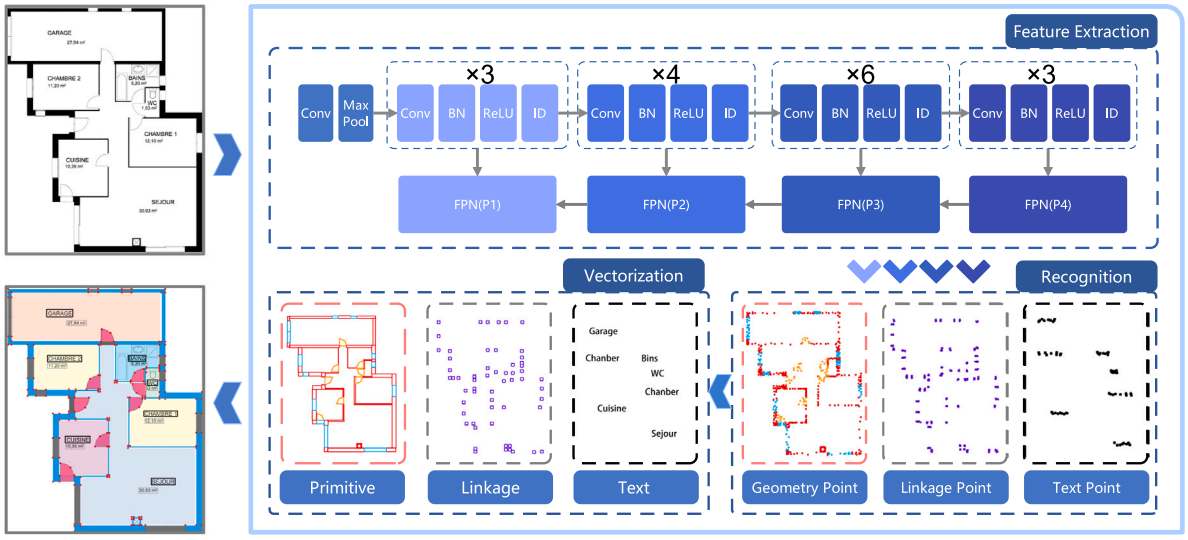



**Fig. 8.** Overview of the proposed model pipeline.

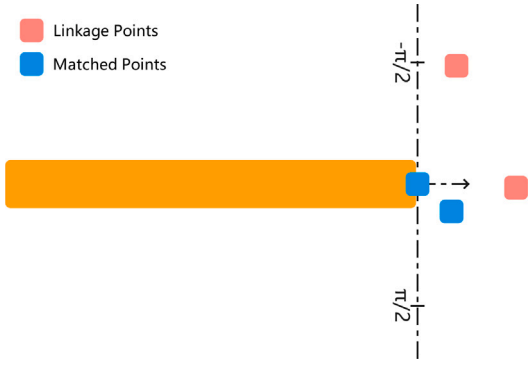



**Fig. 9.** Illustration of search domain for element pairing.

### 4.5. Vectorization

This section transforms predictions into vector graphics, including structural elements, text, and symbols. Our approach generates points directly, avoiding the complex procedures that convert pixels into vertices. Unlike previous methods, our pipeline drops the segmentation branch to maintain a sparse and uniform process while preventing smearing effects. Nevertheless, this operation results in the absence of room regions. Consequently, higher detection accuracy is required to ensure room generation. The details are outlined in the following section.

#### 4.5.1. Linkage integration

Generating rooms necessitates precise alignment of structural components, including walls, doors, and windows, as their integrity is critical to achieving correct room configurations. Any omission in these components can result in erroneous room delineations. Despite predictions closely approximating the ground truth, gaps often arise, particularly at junctions, leading to inaccurate results. We propose a linkage mechanism to mitigate this issue and preserve global topological integrity. Linkage points are derived from the ground truth by generating a rectangular region, 15 pixels in width, around the contact side of each neighboring junction. Non-maximum suppression is applied to eliminate duplicates. These points are categorized as a distinct class, contributing only a single channel to the detection head, thereby incurring minimal computational overhead. The strategic placement of linkage points enhances architectural topology by providing flexibility in selecting confidence thresholds, controlling adjustment areas, and filling gaps. Conventional approaches typically employ a static threshold to balance precision and recall, necessitating manual fine-tuning to optimize performance. In contrast, our proposed linkage strategy adopts a dynamic approach, evaluating the reliability of each structural component. This method ensures seamless integration through a grid of linkage points, typically ranging from two to four per element. This dynamic filtering is effective when certain elements have confidence levels below the pre-determined static threshold, allowing these elements to be reconsidered. Their inclusion in the final structure depends on two main factors: their alignment with the established linkage points, ensuring structural coherence and continuity, and their role in increasing the total room count. This approach allows for a more nuanced and context-sensitive evaluation, potentially saving elements that would fall below the static confidence threshold. Linkage points also preserve original information, preventing the introduction of artifacts that may occur with complex repair procedures. Changes are allowed only around these regions, freezing most of the image area. Additionally, linkage points provide prior information for topological integrity. A valid element should have at least two neighbors, forming a closed loop. Elements with fewer neighbors are placed in a repair queue, which addresses topological gaps caused by missed detections or insufficient size. A search process within a defined angular domain of $[-\pi/2, \pi/2]$ (Fig. 9) locates elements that intersect with one another, based on the geometric principle that two fragments should ideally align straight across from each other. The search algorithm selects the pair of elements that minimize spatial distance, ensuring accurate alignment and continuity.

#### 4.5.2. Room classification

Room classification is another issue of floor plan analysis. This information is typically conveyed through printed text, which is more straightforward and less susceptible to the complexities and multilingual challenges associated with scene text [70], thereby reducing processing difficulty. Consequently, we employed a reliable text recognition model [71] to recognize these printed annotations accurately.

In some cases, spatial enclosures may lack textual identifiers. Heuristic inference can be made based on situational symbols. For example, a toilet usually suggests a bathroom, while a fridge typically indicates a kitchen. These iconographic elements can function similarly to textual descriptions, though they are not always present. Enclosures with dual descriptions or ambiguous symbols require more complex identification

**Table 3**
Quantitative performance comparison on CubiCasa5K dataset.

| Method | Class accuracy | | | | | | | MIoU | Speed (fps) |
|---|---|---|---|---|---|---|---|---|---|
| | Wall | Door | Opening | Window | Railing | Room | Mean | | |
| DeepFloorPlan | **0.93** | 0.79 | 0.71 | 0.72 | 0.49 | 0.75 | 0.73 | 0.70 | 15 |
| IMM | 0.83 | **0.91** | 0.86 | 0.88 | 0.59 | **0.78** | 0.81 | 0.75 | 6.5 |
| **Ours** | 0.88 | 0.90 | **0.89** | **0.89** | **0.62** | 0.77 | **0.82** | **0.79** | **19.6** |

**Table 4**
Boundary analysis on CubiCasa5K dataset..

| Method | BIoU with different boundary ratios | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | 1.0 | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| DeepFloorPlan | 0.51 | 0.70 | 0.68 | 0.65 | 0.61 | 0.56 | 0.50 | 0.44 | 0.38 | 0.32 | 0.25 |
| IMM | 0.57 | 0.75 | 0.72 | 0.69 | 0.70 | 0.67 | 0.61 | 0.56 | 0.41 | 0.35 | 0.28 |
| **Ours** | **0.62** | **0.79** | **0.77** | **0.75** | **0.72** | **0.69** | **0.64** | **0.58** | **0.49** | **0.39** | **0.33** |

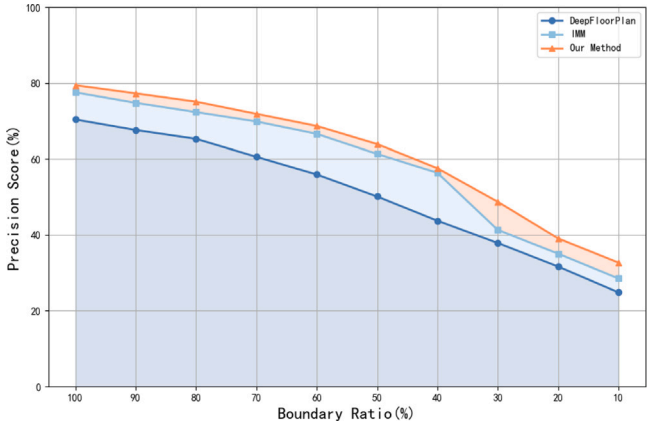



**Fig. 10.** Boundary analysis on CubiCasa5K dataset.

rules, which are beyond the scope of this research. We focus on extracting geometric primitives from rasterized images and ignore these intricate cases.

## 5. Experiment

In this section, we conduct comprehensive experiments to evaluate the proposed method. We outline the implementation details and then comprehensively assess our method on the CubiCasa5k [17] and CFP Datasets. Finally, we perform ablation studies to analyze the impact of critical components.

### 5.1. Implementation details

The proposed model uses the Stochastic Gradient Descent (SGD) optimizer, setting the learning rate to 0.001, momentum to 0.9, and weight decay to 0.0001. We conduct the training over 60 epochs with a batch size of 4. The experiments are performed in a controlled environment using four NVIDIA RTX 3090 GPUs and the PyTorch 1.5 framework. All models are trained and tested under consistent conditions to ensure the reliability and comparability of the results.

DeepFloorPlan [4] and IMM [5] are chosen in our experiment due to their demonstrated effectiveness within respective domains. DeepFloorPlan is a semantic segmentation model, and IMM is an instance segmentation model. Both methods can recognize elements of arbitrary shapes. Given that the original IMM model lacks room-categorizing ability, we improve this functionality by integrating descriptions from our results.

Pixel Accuracy (PA) and Mask Intersection over Union (MIoU) are adopted as the primary evaluation metrics. Boundary IoU (BIoU) [72] metric is applied to evaluate the vectorization ability in capturing boundary details. BIoU is highly effective in assessing boundary preservation and delineation, especially in architectural plans where clear boundaries are essential for usability. The evaluation includes multiple boundary retention ratios to adapt BIoU to this specific task, facilitating a comprehensive assessment of boundary details. Additionally, t-tests were conducted to assess statistical significance, with both p-values and t-values for the boundary evaluation being reported to confirm the reliability of the delineation.

### 5.2. Evaluation on CubiCasa5K dataset

The CubiCasa5K dataset comprises 5000 samples and is divided into training, validation, and test sets with 4200, 400, and 400 images, respectively. It includes the Scalable Vector Graphics (SVG) format, which facilitates instance extraction and supports flexible applications.

Table 3 compares the performance of our method with DeepFloorPlan and IMM across several evaluation metrics. Our approach shows improvements in mean PA, scoring 0.82, representing an increase of 1 and 9 percentage points over IMM and DeepFloorPlan, respectively. Similar gains are observed in MIoU, with 4 and 9 percentage points improvements. By eliminating the traditional segmentation branch, our method achieves an inference speed of 19.6 frames per second, making it approximately 3 times faster than IMM and 1.3 times faster than DeepFloorPlan.

Table 4 and Fig. 10 present the results of the boundary analysis, where our method consistently outperforms contemporary approaches. This analysis focuses on hollowing out the interior of elements and increasing the significance of boundaries, progressively evaluating the clarity of vectorized elements. Our method excels across all boundary ratios, outperforming IMM and DeepFloorPlan by 5 and 9 percentage points, respectively. In the t-test, our method shows t-values of 3.08 and 7.27 compared to IMM and DFP, respectively, with p-values of $2.61 \times 10^{-3}$ and $1.33 \times 10^{-5}$. These results highlight the effectiveness of our approach in accurately preserving boundary details.

Fig. 11 illustrates a comparative analysis of the models. DeepFloorPlan preserves the overall architectural structure but demonstrates suboptimal performance in capturing finer details. As depicted in Fig. 12, a closer examination reveals that DeepFloorPlan produces blurred boundaries, with many regions appearing jagged, underscoring the inherent challenges of accurately capturing intricate architectural features. While IMM generally shows improvements, it fails to address the issue of boundary blurriness due to limitations in its segmentation design. Both methods rely on global or local segmentation operations, which can result in smeared effects. Since the defining characteristics of architectural elements are often concentrated at their boundaries, segmentation errors in hollow or blank spaces can aggravate these issues. In practice, regions with similar features or textures may represent distinct rooms, further complicating accurate regional classification. Our method shows robustness in handling most elements that uniformly

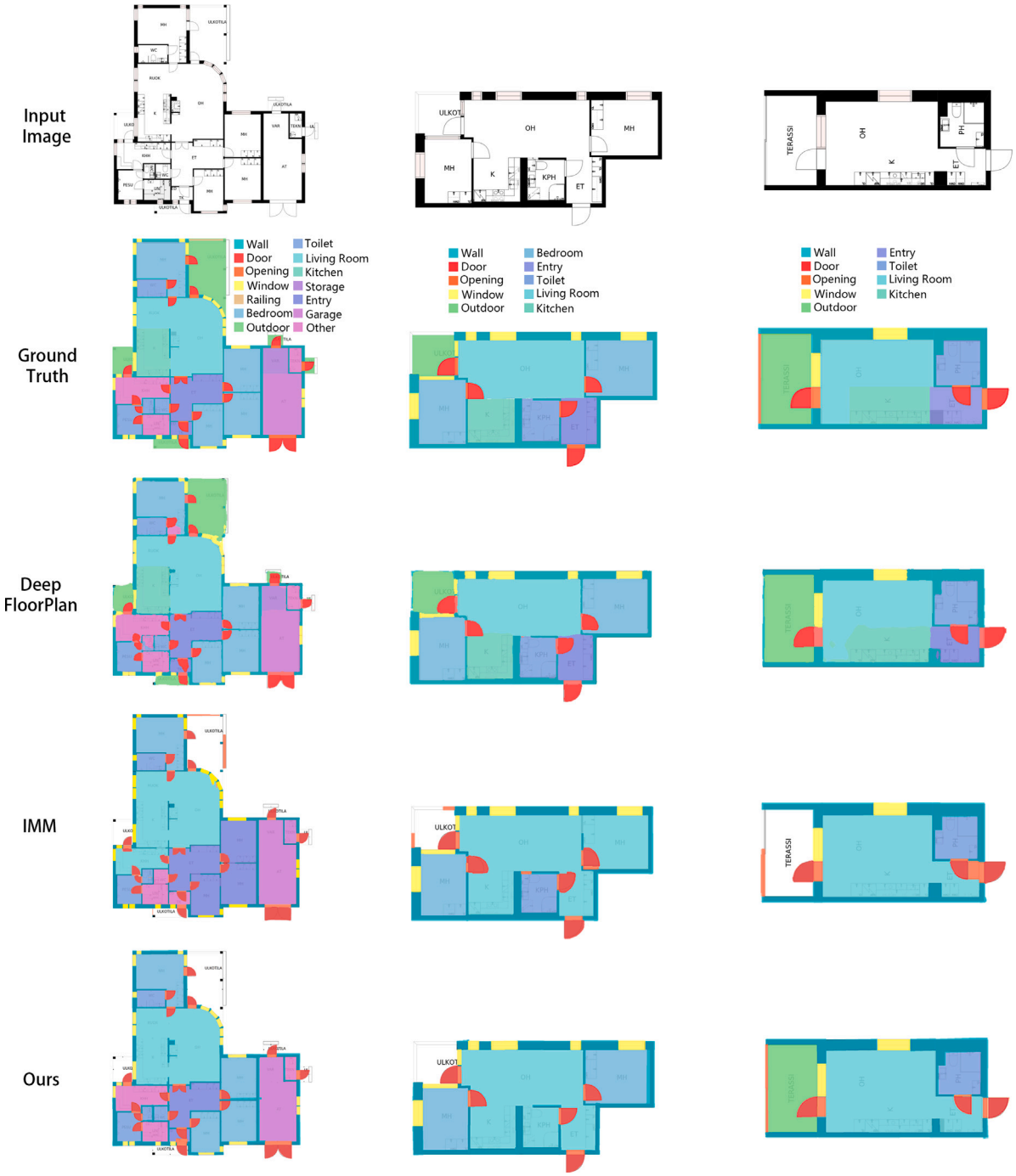



**Fig. 11.** Visual comparison on CubiCasa5K dataset.

perform parameterization using a quartet of points. The TopNHull regulation ensures that boundaries are rendered sharply and clearly, which is crucial for precise architectural delineation. However, instance-based methods, including our approach and IMM, face difficulties in identifying rooms that are not fully enclosed or lack distinguishable text, as illustrated at the bottom of Fig. 11.

### 5.3. Evaluation on CFP dataset

The CFP dataset comprises 1062 samples, divided into training, validation, and test sets containing 800, 100, and 162 images. These samples were initially downscaled to accommodate GPU memory constraints. This operation significantly reduced the size of elements, posing a challenge in identifying tiny objects. Empirical observations confirmed this limitation, as neither IMM nor the proposed model achieved convergence on the downscaled images. To address this issue, we implemented a multi-scale training strategy that divided images into patches to avoid reducing the regions of interest to tiny scales, thus preserving the recognizability of features. However, DeepFloorPlan employs a global semantic segmentation with room boundary attention. Splitting images into patches results in the disintegration of room entities, undermining the holistic structural integrity and generating

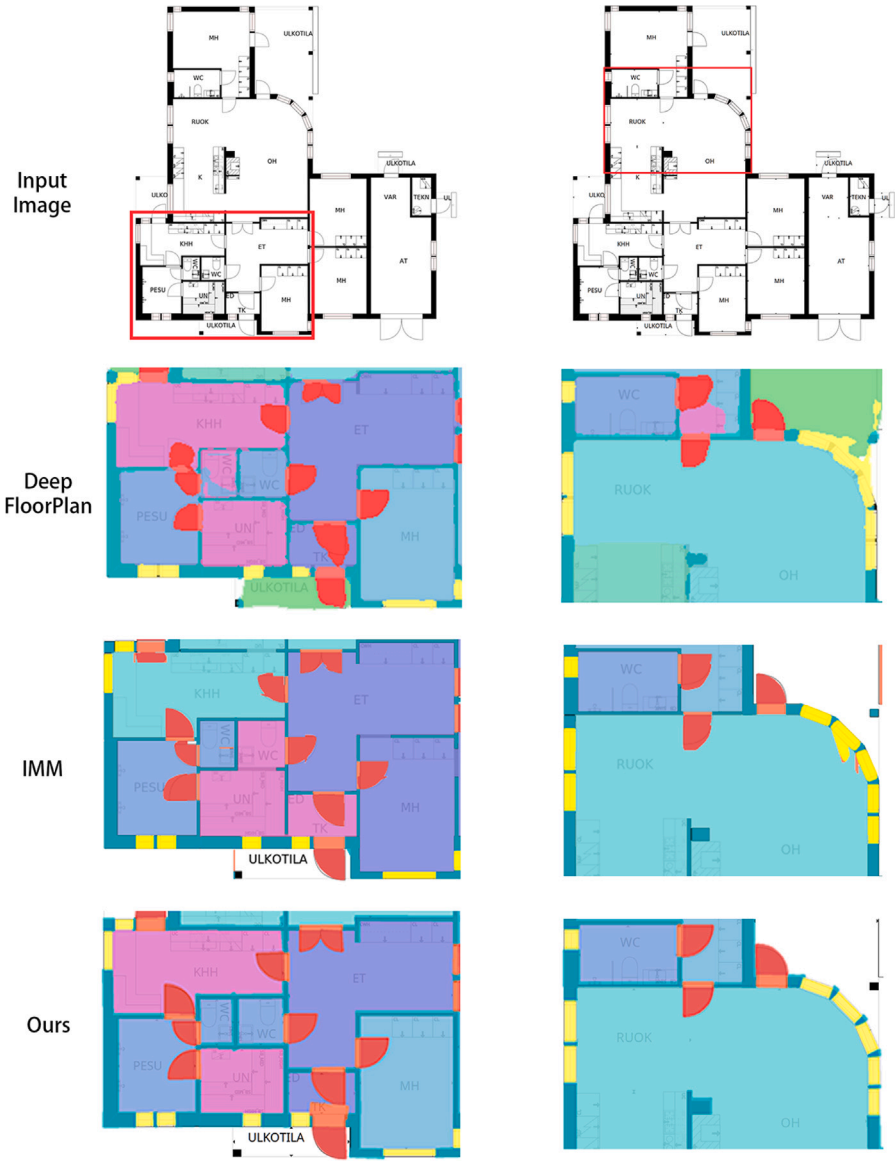



**Fig. 12.** Detailed comparison on CubiCasa5K dataset.

**Table 5**
Quantitative performance comparison on CFP dataset.

| Method | PA | | | | | | MIoU | Speed (fps) |
|---|---|---|---|---|---|---|---|---|
| | Wall | Door | Window | Sliding | Room | Mean | | |
| DeepFloorPlan | **0.98** | 0.77 | 0.39 | 0.45 | 0.54 | 0.62 | 0.54 | **15** |
| IMM | 0.75 | **0.87** | **0.67** | **0.77** | 0.77 | **0.77** | 0.67 | 2.1 |
| **Ours** | 0.90 | 0.80 | 0.61 | 0.64 | **0.88** | **0.77** | **0.68** | 6.5 |

many fragments without semantics. This outcome stands in stark opposition to the design principles of DeepFloorPlan. Therefore, the patching operation was not adopted in DeepFloorPlan.

Table 5 compares the performance metrics of our method against DeepFloorPlan and IMM. The complexity of CFP images leads to a noticeable decline in performance for all methods. Our approach still performs better than the others across most metrics. In terms of mean PA, Our method scores at 0.77, equivalent to IMM, and surpasses DeepFloorPlan by 15 percentage points. Regarding MIoU, our method leads by 1 and 14 percentage points, respectively.

Table 6 and Fig. 13 present the boundary evaluation results, where the proposed method consistently shows a more apparent advantage. Our method outperforms IMM by 5 and DeepFloorPlan by 11 percentage points on average. In the t-test, our method shows t-values of 17.77 and 6.79 compared to IMM and DFP, with p-values of $1.43 \times 10^{-32}$ and $8.36 \times 10^{-10}$, respectively.

Fig. 14 visually compares our approach and counterparts. Compared to the existing dataset, these samples include more irregular elements, imposing more stringent requirements for recognition. DeepFloorPlan continuously exhibits a blurring effect when recognizing most structures. This phenomenon is primarily due to the structural

**Table 6**
Boundary analysis on CFP dataset.

| Method | BIoU with different boundary ratios | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | 1.0 | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| DeepFloorPlan | 0.39 | 0.54 | 0.51 | 0.48 | 0.45 | 0.41 | 0.37 | 0.34 | 0.29 | 0.27 | 0.20 |
| IMM | 0.47 | 0.67 | 0.64 | 0.60 | 0.57 | 0.53 | 0.44 | 0.39 | 0.33 | 0.30 | 0.23 |
| **Ours** | **0.52** | **0.68** | **0.66** | **0.63** | **0.61** | **0.58** | **0.52** | **0.44** | **0.40** | **0.36** | **0.30** |

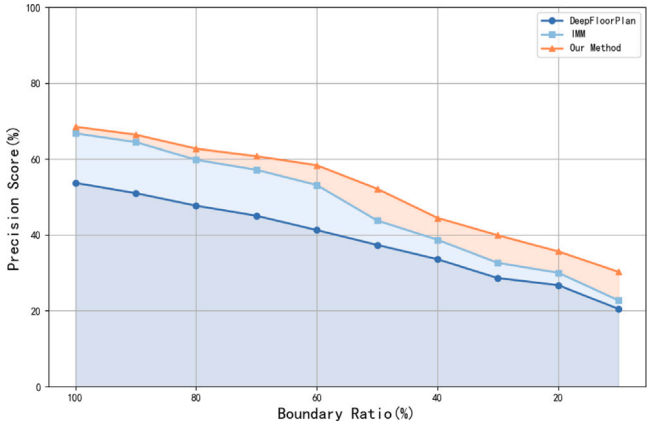



**Fig. 13.** Boundary analysis on CFP dataset.

complexities of the CFP dataset, which includes more than 30 types. A notable feature of this dataset is the infrequent occurrence of specific room types, resulting in a long-tail distribution that poses difficulties for such samples. Additionally, many rooms present remarkable similarities with backgrounds where large blanks abound in internal spaces, bringing extra challenges to the segmentation process. IMM demonstrates acceptable performance with horizontally and vertically oriented elements, a strength attributed to its predefined bounding box. However, this function struggles with angled or irregular elements, also resulting in jagged or smeared edges. These findings highlight the inherent limitations of segmentation approaches, mainly when applied to complex structures. In contrast, our proposed method excels in processing complex architectural layouts, accurately illustrating most elements while consistently maintaining clear and sharp boundaries. Additionally, it effectively bridges gaps, ensuring structural integrity and consistency in the output, thereby demonstrating its advanced capabilities in handling diverse architectural forms. Fig. 15 compares our method and the predictions from IMM and DeepFloorPlan, emphasizing the limitations of segmentation-based approaches. Human visual perception struggles to differentiate room types based on blank or similar local features, which largely contributes to the significant errors seen in the DeepFloorPlan method. In contrast, incorporating textual information helps reduce these ambiguities, leading to more accurate room classification.

### 5.4. Ablation study

#### 5.4.1. Ablation of representation

We initially removed the TopNHull algorithm, reducing the model to a conventional rotated object detector. In this simplified form, elements are depicted as rotated bounding boxes, typically represented by a 5-dimensional vector $[X, Y, W, H, A]$. This vector includes the centroid $(X, Y)$, dimensions $(W, H)$, and rotation angle $(A)$ of the bounding box. Fig. 16 illustrates the results following this ablation study.

As expected, this simplified representation reduced elements to quadrilaterals, which could not accurately capture sectors. However, regular or inclined elements remained unaffected. During training, the costs associated with the $[X, Y, W, H, A]$ representation (Fig. 5(c)) and our method (Fig. 5(d)) were found to be similar, as both approaches ultimately require conversion into a convex region.

Our method preserves polygonal profiles and accounts for the sectorial regions. In contrast, the $[X, Y, W, H, A]$ representation oversimplifies intervals of doors, which can impact a room's layout. While this representation may be sufficient for tasks that do not require precise identification of door orientations, our approach provides a more comprehensive representation of architectural elements.

**Table 7**
Ablation study of linkage integration..

| Strategy | Room accuracy |
|---|---|
| – | 0.82 |
| Linkage Integration | **0.88** |

#### 5.4.2. Ablation of linkage integration

We conducted another ablation study on the linkage integration strategy during the vectorization process. Removing this strategy led to gaps between elements, causing rooms to merge with the background. This issue is particularly pronounced in instance-based approaches, where achieving perfect alignment between elements is inherently challenging. Table 7 shows a decrease in room accuracy of 6 percentage points after removing linkage integration, underlining the significance of this strategy.

## 6. Limitations

The primary limitation lies in the reliance on a quartet of definitions to represent all shapes, which inherently excludes the parameterization of higher-order curves.

Another challenge is the distribution of feature points. As shown in Fig. 7, four points can represent the sectorial region well, but an asymmetric distribution of these points can make it difficult to determine orientation. For instance, when $A$, $B$, and $C$ are collinear, the Top3Hull configuration becomes indistinguishable from Top4Hull. If $\triangle OAB$ forms an isosceles triangle, the arc side $AB$ can be identified. However, the orientation is lost as the region becomes an equilateral triangle.

Bias in the annotation process is another limitation. Several students carried out the annotation, which inevitably introduced personal considerations. Although our experiments prove the usability of the dataset, further refinement is required.

The last limitation is the incomplete use of semantic information. Elements like walls and windows sometimes look visually identical. Therefore, supportive text or descriptions are required for accurate category determination. However, this operation is complicated because of the inconsistent placement of descriptive text, which might be near the element or indicated by an arrow.

## 7. Conclusions

This paper introduced a new approach and dataset to advance floor plan vectorization research, particularly in handling complex architectural layouts. The proposed method overcomes notable limitations by employing a sparse, uniform representation, facilitating robust and efficient processing of intricate floor plans. The CFP dataset contributes diverse samples that broaden analytical capabilities and enable future

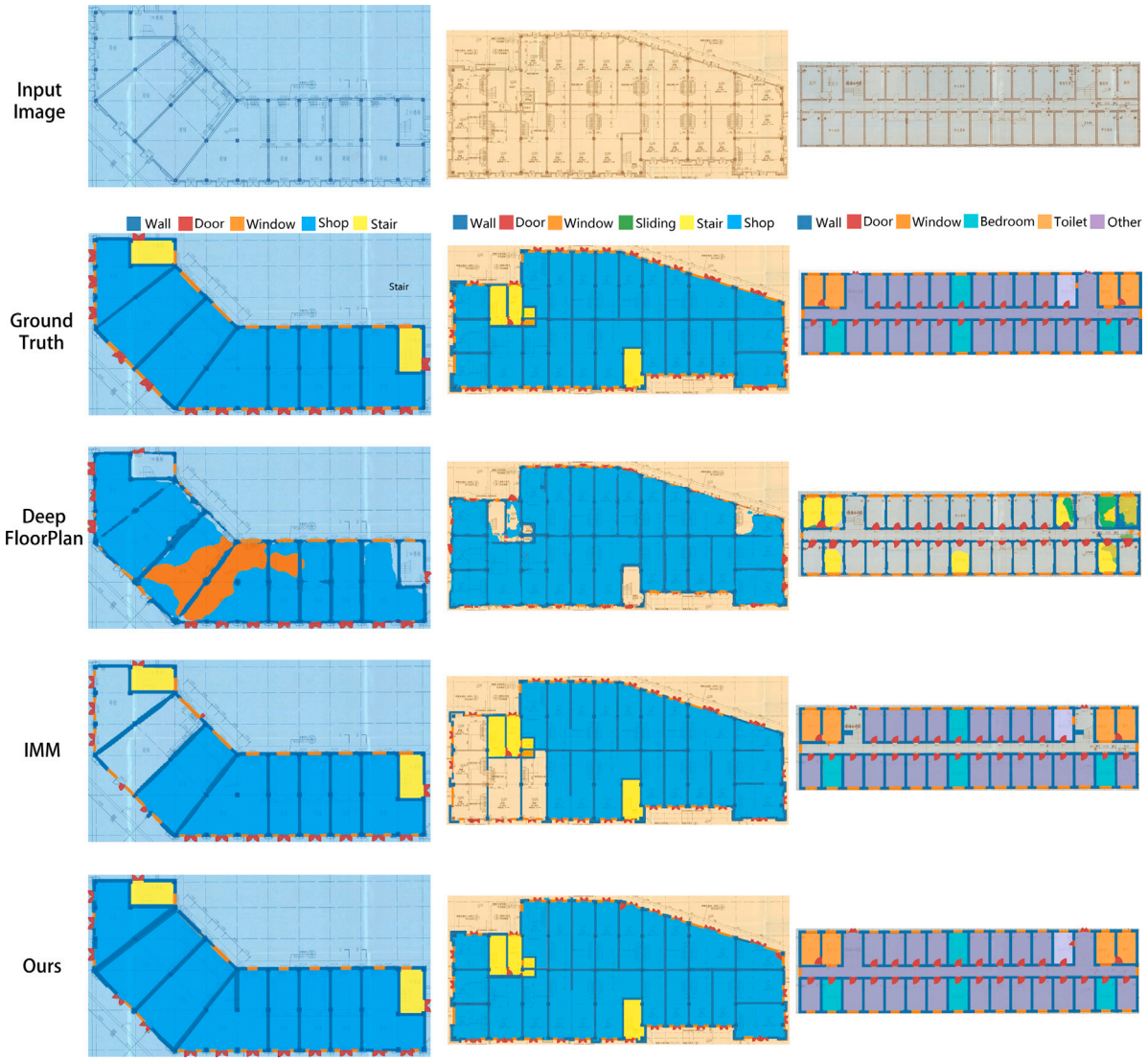



**Fig. 14.** Visual comparison on CFP dataset.

advancements in automated recognition research. Quantitative evaluations on the CubiCasa5K and CFP datasets reveal the superiority of our approach compared to established methods, including DeepFloorPlan and IMM, across most metrics. Specifically, our method significantly improves boundary delineation accuracy and computational efficiency. The enhanced performance underscores the model's suitability for capturing varied architectural forms, including inclined and irregular geometries that challenge conventional methods. The broader implications of this work extend to professional applications in architecture and construction, such as digital building reconstruction, indoor navigation, and intelligent infrastructure systems. These applications benefit from the enhanced accuracy and reliability of the vectorization method, addressing crucial demands in engineering workflows. Future research could refine the framework by incorporating advanced parameterization for high-order curves and integrating semantic context to improve classification accuracy for visually similar elements. Such advancements would further enhance the model's utility in diverse architectural scenarios.

## CRediT authorship contribution statement

**Jici Xing:** Writing – original draft, Supervision, Resources, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **Longyong Wu:** Visualization, Validation, Conceptualization. **Tianyi Zeng:** Writing – original draft. **Yijie Wu:** Writing – review & editing, Conceptualization. **Jianga Shang:** Writing – review & editing.

## Declaration of competing 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.

## Acknowledgments

This study was supported by the National Key R&D Program of China (No. 26602) supported this study, and YiKun Data Information Technology Co., Ltd provided the data.

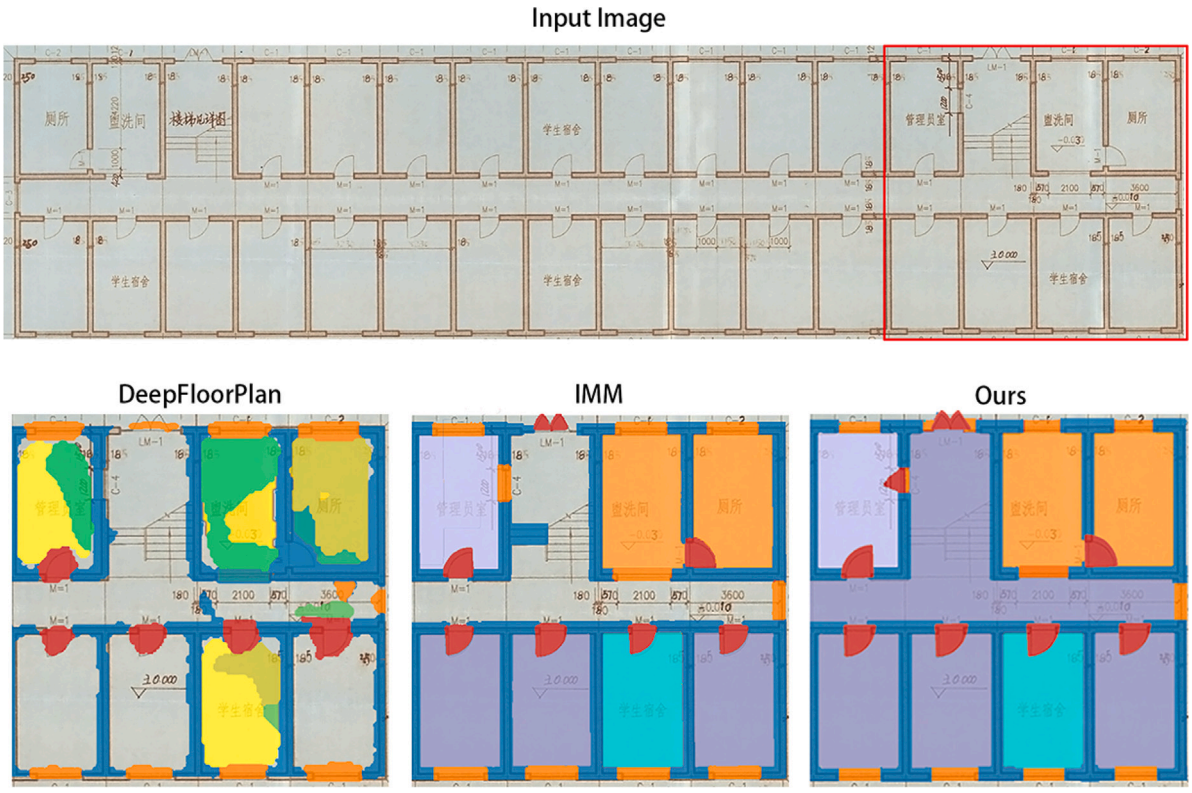



**Fig. 15.** Detailed comparison on CFP dataset.

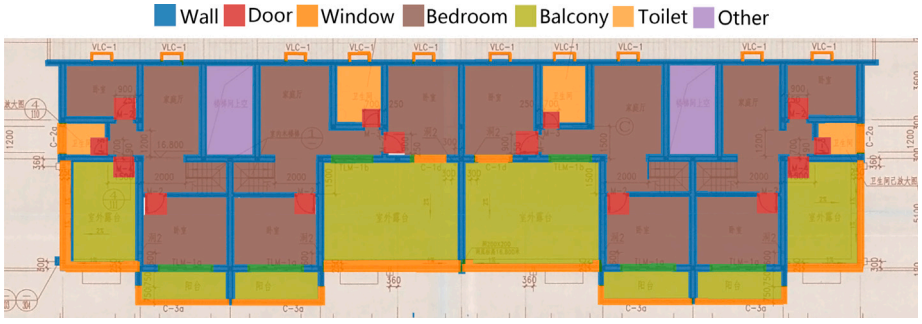



**Fig. 16.** Ablation of representation.

## Data availability

The CFP dataset will be accessible exclusively for academic pursuits.

## References

[1] P.N. Pizarro, N. Hitschfeld, I. Sipiran, J.M. Saavedra, Automatic floor plan analysis and recognition, Autom. Constr. 140 (2022) 104348, http://dx.doi.org/10.1016/j.autcon.2022.104348.

[2] C. Liu, J. Wu, P. Kohli, Y. Furukawa, Raster-to-vector: Revisiting floorplan transformation, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2195–2203, http://dx.doi.org/10.1109/iccv.2017.241.

[3] T. Yamasaki, J. Zhang, Y. Takada, Apartment structure estimation using fully convolutional networks and graph model, in: Proceedings of the 2018 ACM Workshop on Multimedia for Real Estate Tech, 2018, pp. 1–6, http://dx.doi.org/10.1145/3210499.3210528.

[4] Z. Zeng, X. Li, Y.K. Yu, C.-W. Fu, Deep floor plan recognition using a multi-task network with room-boundary-guided attention, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 9095–9103, http://dx.doi.org/10.1109/iccv.2019.00919.

[5] Y. Wu, J. Shang, P. Chen, S. Zlatanova, X. Hu, Z. Zhou, Indoor mapping and modeling by parsing floor plan images, Int. J. Geogr. Inf. Sci. 35 (6) (2020) 1205–1231, http://dx.doi.org/10.1080/13658816.2020.1781130.

[6] I.Y. Surikov, M.A. Nakhatovich, S.Y. Belyaev, D.A. Savchuk, Floor plan recognition and vectorization using combination UNet, faster-RCNN, statistical component analysis and ramer-douglas-peucker, in: Communications in Computer and Information Science, Springer, 2020, pp. 16–28, http://dx.doi.org/10.1007/978-981-15-6648-6_2.

[7] J. Xing, Q. Luo, Y. Wu, J. Shang, Self-optimization for parsing floor plans, J. Comput. Civ. Eng. 36 (6) (2022) 04022037, http://dx.doi.org/10.1061/(asce)cp.1943-5487.0001048.

[8] Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, Proc. the IEEE (2023) 257–276, http://dx.doi.org/10.1109/jproc.2023.3238524.

[9] S. Hao, Y. Zhou, Y. Guo, A brief survey on semantic segmentation with deep learning, Neurocomputing 406 (2020) 302–321, http://dx.doi.org/10.1016/j.neucom.2019.11.118.

[10] W. Gu, S. Bai, L. Kong, A review on 2D instance segmentation based on deep neural networks, Image Vis. Comput. 120 (2022) 104401, http://dx.doi.org/10.1016/j.imavis.2022.104401.

[11] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: Lecture Notes in Computer Science, Springer, 2014, pp. 740–755, http://dx.doi.org/10.1007/978-3-319-10602-1_48.

[12] S. Wei, X. Zeng, Q. Qu, M. Wang, H. Su, J. Shi, HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation, IEEE Access 8 (2020) 120234–120254, http://dx.doi.org/10.1109/access.2020.3005861.

[13] X. Lv, S. Zhao, X. Yu, B. Zhao, Residential floor plan recognition and reconstruction, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 16712–16721, http://dx.doi.org/10.1109/cvpr46437.2021.01644.

[14] Z. Lu, T. Wang, J. Guo, W. Meng, J. Xiao, W. Zhang, X. Zhang, Data-driven floor plan understanding in rural residential buildings via deep recognition, Inf. Sci. 567 (2021) 58–74, http://dx.doi.org/10.1016/j.ins.2021.03.032.

[15] Z. Yang, S. Liu, H. Hu, L. Wang, S. Lin, RepPoints: Point set representation for object detection, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 9656–9665, http://dx.doi.org/10.1109/iccv.2019.00975.

[16] J. Sun, Y. Bie, Y. Zhang, A. Bie, Adversarial sample generation for traffic sign recognition deep neural network models based on attack space generative adversarial network, 2024, http://dx.doi.org/10.2139/ssrn.4976577, arXiv preprint arXiv:2001.11194.

[17] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, J. Kannala, CubiCasa5K: A dataset and an improved multi-task model for floorplan image analysis, in: Lecture Notes in Computer Science, Springer, 2019, pp. 28–40, http://dx.doi.org/10.1007/978-3-030-20205-7_3.

[18] K. Ryall, S. Shieber, J. Marks, M. Mazer, Semi-automatic delineation of regions in floor plans, in: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 2, IEEE, 1995, pp. 964–969, http://dx.doi.org/10.1109/icdar.1995.602062.

[19] Y.K. Yu, S.H. Or, K.H. Wong, M. Chang, Accurate 3-D motion tracking with an application to super-resolution, in: 18th International Conference on Pattern Recognition, ICPR'06, IOS Press Amsterdam, The Netherlands, 2006, pp. 730–733, http://dx.doi.org/10.1109/icpr.2006.202.

[20] S. Ahmed, M. Liwicki, M. Weber, A. Dengel, Improved automatic analysis of architectural floor plans, in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 864–869, http://dx.doi.org/10.1109/icdar.2011.177.

[21] L. Gimenez, S. Robert, F. Suard, K. Zreik, Automatic reconstruction of 3D building models from scanned 2D floor plans, Autom. Constr. 63 (2016) 48–56, http://dx.doi.org/10.1016/j.autcon.2015.12.008.

[22] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 3431–3440, http://dx.doi.org/10.1109/cvpr.2015.7298965.

[23] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, http://dx.doi.org/10.1109/cvpr.2016.182.

[24] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Lecture Notes in Computer Science, Springer, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single shot MultiBox detector, in: Lecture Notes in Computer Science, Springer, 2016, pp. 21–37, http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[26] W. Wang, S. Dong, K. Zou, W. sheng L.I., Room classification in floor plan recognition, in: 2020 4th International Conference on Advances in Image Processing, 2020, pp. 48–54, http://dx.doi.org/10.1145/3441250.3441265.

[27] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1804, CVPR, Springer Berlin/Heidelberg, Germany, 2017, pp. 6517–6525, http://dx.doi.org/10.1109/cvpr.2017.690.

[28] S.R. Dirisala, D. Padidem, Comparative study of the different object detection algorithms: YOLOv4, SSD, and RCNN based on accuracy and speed, Int. J. Sci. Res. (IJSR) (2023) 1560–1565, http://dx.doi.org/10.21275/sr231020230143.

[29] R. Khade, K. Jariwala, C. Chattopadhyay, U. Pal, A rotation and scale invariant approach for multi-oriented floor plan image retrieval, Pattern Recognit. Lett. 145 (2021) 1–7, http://dx.doi.org/10.1016/j.patrec.2021.01.020.

[30] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2017) 1137–1149, http://dx.doi.org/10.1109/tpami.2016.2577031.

[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778, http://dx.doi.org/10.1109/cvpr.2016.90.

[32] R. Cipolla, Y. Gal, A. Kendall, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7482–7491, http://dx.doi.org/10.1109/cvpr.2018.00781.

[33] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2961–2969, http://dx.doi.org/10.1109/iccv.2017.322.

[34] M. Rusiñol, A. Borràs, J. Lladós, Relational indexing of vectorial primitives for symbol spotting in line-drawing images, Pattern Recognit. Lett. 31 (3) (2010) 188–201, http://dx.doi.org/10.1016/j.patrec.2009.10.002.

[35] M. Delalandre, E. Valveny, T. Pridmore, D. Karatzas, Generation of synthetic documents for performance evaluation of symbol recognition spotting systems, Int. J. Doc. Anal. Recognition (IJDAR) 13 (3) (2010) 187–207, http://dx.doi.org/10.1007/s10032-010-0120-x.

[36] S. Macé, H. Locteau, E. Valveny, S. Tabbone, A system to detect rooms in architectural floor plan images, in: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, 2010, pp. 167–174, http://dx.doi.org/10.1145/1815330.1815352.

[37] C. Liu, A.G. Schwing, K. Kundu, R. Urtasun, S. Fidler, Rent3D: Floor-plan priors for monocular layout estimation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 3413–3421, http://dx.doi.org/10.1109/cvpr.2015.7298963.

[38] H. Chu, S. Wang, R. Urtasun, S. Fidler, HouseCraft: Building houses from rental ads and street views, in: Lecture Notes in Computer Science, Springer, 2016, pp. 500–516, http://dx.doi.org/10.1007/978-3-319-46466-4_30.

[39] S. Dodge, J. Xu, B. Stenger, Parsing floor plan images, in: 2017 Fifteenth IAPR International Conference on Machine Vision Applications, MVA, IEEE, 2017, pp. 358–361, http://dx.doi.org/10.23919/mva.2017.7986875.

[40] D. Sharma, N. Gupta, C. Chattopadhyay, S. Mehta, DANIEL: A deep architecture for automatic analysis and retrieval of building floor plans, in: 2017 14th IAPR International Conference on Document Analysis and Recognition, Vol. 1, ICDAR, IEEE, 2017, pp. 420–425, http://dx.doi.org/10.1109/icdar.2017.76.

[41] W. Wu, X.-M. Fu, R. Tang, Y. Wang, Y.-H. Qi, L. Liu, Data-driven interior plan generation for residential buildings, ACM Trans. Graph. 38 (6) (2019) 1–12, http://dx.doi.org/10.1145/3355089.3356556.

[42] T. Li, D. Ho, C. Li, D. Zhu, C. Wang, M.Q.H. Meng, HouseExpo: A large-scale 2D indoor layout dataset for learning-based algorithms on mobile robots, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 5839–5846, http://dx.doi.org/10.1109/iros45743.2020.9341284.

[43] M. Vidanapathirana, Q. Wu, Y. Furukawa, A.X. Chang, M. Savva, Plan2Scene: Converting floorplans to 3D scenes, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 10728–10737, http://dx.doi.org/10.1109/cvpr46437.2021.01059.

[44] C.P. Simonsen, F.M. Thiesson, M.P. Philipsen, T.B. Moeslund, Generalizing floor plans using graph neural networks, in: 2021 IEEE International Conference on Image Processing, ICIP, IEEE, 2021, pp. 654–658, http://dx.doi.org/10.1109/icip42928.2021.9506514.

[45] Z. Fan, L. Zhu, H. Li, X. Chen, S. Zhu, P. Tan, FloorPlanCAD: A large-scale CAD drawing dataset for panoptic symbol spotting, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 10108–10117, http://dx.doi.org/10.1109/iccv48922.2021.00997.

[46] S. Goyal, V. Mistry, C. Chattopadhyay, G. Bhatnagar, BRIDGE: Building plan repository for image description generation, and evaluation, in: 2019 International Conference on Document Analysis and Recognition, ICDAR, IEEE, 2019, pp. 1071–1076, http://dx.doi.org/10.1109/icdar.2019.00174.

[47] H. Jang, K. Yu, J. Yang, Indoor reconstruction from floorplan images with a deep learning approach, ISPRS Int. J. Geo- Inf. 9 (2) (2020) 65, http://dx.doi.org/10.3390/ijgi9020065.

[48] S. Dong, W. Wang, W. Li, K. Zou, Vectorization of floor plans based on EdgeGAN, Inf. 12 (5) (2021) 206, http://dx.doi.org/10.3390/info12050206.

[49] L. Gimenez, J.-L. Hippolyte, S. Robert, F. Suard, K. Zreik, Review: reconstruction of 3D building information models from 2D scanned plans, J. Build. Eng. 2 (2015) 24–35, http://dx.doi.org/10.1016/j.jobe.2015.04.002.

[50] Q. Lu, S. Lee, A semi-automatic approach to detect structural components from CAD drawings for constructing As-Is BIM objects, in: Computing in Civil Engineering 2017, 2017, pp. 84–91, http://dx.doi.org/10.1061/9780784480823.011.

[51] Y. Zhao, X. Deng, H. Lai, Reconstructing BIM from 2D structural drawings for existing buildings, Autom. Constr. 128 (2021) 103750, http://dx.doi.org/10.1016/j.autcon.2021.103750.

[52] S. Wu, N. Zhang, X. Luo, W.Z. Lu, Multi-objective optimization in floor tile planning: Coupling BIM and parametric design, Autom. Constr. 140 (2022) 104384, http://dx.doi.org/10.1016/j.autcon.2022.104384.

[53] S. Wu, N. Zhang, X. Luo, W.Z. Lu, Intelligent optimal design of floor tiles: A goal-oriented approach based on BIM and parametric design platform, J. Clean. Prod. 299 (2021) 126754, http://dx.doi.org/10.1016/j.jclepro.2021.126754.

[54] H. Liu, C. Sydora, M.S. Altaf, S. Han, M. Al-Hussein, Towards sustainable construction: BIM-enabled design and planning of roof sheathing installation for prefabricated buildings, J. Clean. Prod. 235 (2019) 1189–1201, http://dx.doi.org/10.1016/j.jclepro.2019.07.055.

[55] S. Goyal, C. Chattopadhyay, G. Bhatnagar, Knowledge-driven description synthesis for floor plan interpretation, Int. J. Doc. Anal. Recognition (IJDAR) 24 (1–2) (2021) 19–32, http://dx.doi.org/10.1007/s10032-021-00367-3.

[56] G. Gerstweiler, L. Furlan, M. Timofeev, H. Kaufmann, Extraction of structural and semantic data from 2D floor plans for interactive and immersive VR real estate exploration, Technol. 6 (4) (2018) 101, http://dx.doi.org/10.3390/technologies6040101.

[57] Y. Song, R. Koeck, S. Luo, Review and analysis of augmented reality (AR) literature for digital fabrication in architecture, Autom. Constr. 128 (2021) 103762, http://dx.doi.org/10.1016/j.autcon.2021.103762.

[58] E. Lewandowicz, P. aw Lisowski, Methodology to generate navigation models in building, J. Civ. Eng. Manag. 24 (8) (2018) 619–629, http://dx.doi.org/10.3846/jcem.2018.6599.

[59] B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, LabelMe: A database and web-based tool for image annotation, Int. J. Comput. Vis. 77 (2007) 157–173, http://dx.doi.org/10.1007/s11263-007-0090-8.

[60] Z. Yang, Y. Xu, H. Xue, Z. Zhang, R. Urtasun, L. Wang, S. Lin, H. Hu, Dense RepPoints: Representing visual objects with dense point sets, in: Lecture Notes in Computer Science, Springer, 2020, pp. 227–244, http://dx.doi.org/10.1007/978-3-030-58589-1_14.

[61] W. Li, Y. Chen, K. Hu, J. Zhu, Oriented RepPoints for aerial object detection, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 1819–1828, http://dx.doi.org/10.1109/cvpr52688.2022.00187.

[62] L. Hou, K. Lu, J. Xue, Y. Li, Shape-adaptive selection and measurement for oriented object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, (no. 1) 2022, pp. 923–932, http://dx.doi.org/10.1609/aaai.v36i1.19975.

[63] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, DOTA: A large-scale dataset for object detection in aerial images, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3974–3983, http://dx.doi.org/10.1109/cvpr.2018.00418.

[64] Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 9357–9366, http://dx.doi.org/10.1109/cvpr.2019.00959.

[65] T.Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 2117–2125, http://dx.doi.org/10.1109/cvpr.2017.106.

[66] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 764–773, http://dx.doi.org/10.1109/iccv.2017.89.

[67] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2980–2988, http://dx.doi.org/10.1109/iccv.2017.324.

[68] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 658–666, http://dx.doi.org/10.1109/cvpr.2019.00075.

[69] J. Yu, Y. Jiang, Z. Wang, Z. Cao, T. Huang, UnitBox, in: Proceedings of the 24th ACM International Conference on Multimedia, 2016, pp. 516–520, http://dx.doi.org/10.1145/2964284.2967274.

[70] F. Naiemi, V. Ghods, H. Khalesi, Scene text detection and recognition: a survey, Multimed. Tools Appl. 81 (14) (2022) 20255–20290, http://dx.doi.org/10.1007/s11042-022-12693-7.

[71] PaddlePaddle, PaddleOCR: Awesome multilingual OCR toolkits based on PaddlePaddle, 2024, GitHub repository. URL https://github.com/PaddlePaddle/PaddleOCR.

[72] B. Cheng, R. Girshick, P. Dollar, A.C. Berg, A. Kirillov, Boundary IoU: Improving object-centric image segmentation evaluation, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2021, http://dx.doi.org/10.1109/cvpr46437.2021.01508.