FloorPlanFormer: Multi-Task Transformer Network for Floor Plan Recognition with Outer-to-Inner Feature Refinement

Yun Liang[1]*, ZiHao Wu[1], Run Zheng[1], Shuai Xie[1], Bo Hong[1], Yishen Lin[1]

[1]College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China
yliang@scau.edu.cn, wzhwzh@stu.scau.edu.cn, run@stu.scau.edu.cn

Abstract

Floor plan recognition requires accurate segmentation and classification of entrance doors, outer contours (walls and windows) and inner contours (various room types), despite strong spatial dependencies and large stylistic differences between different datasets. To overcome these challenges, we propose FloorPlanFormer, a multi-task learning network divided into three phases: the first phase introduces a Swin Transformer backbone with a pixel decoder to extract fine-grained pixel-level semantics; the second phase employs prompt encoder and mask decoder, and a novel Global Contextual Attention Module (GCAM) is designed to generate clear, high-quality outer contour masks; the third stage uses mask transformer decoder to recognize targets and designs a Masked Feature Refinement Module (MFRM) to accurately delineate the inner contour by modeling the relationship between the local inner and outer contours. Finally, we constructed FloorPlan8K, a dataset containing 8200 images and 77434 instances, on which our model was trained and evaluated, and the results greatly outperformed the state-of-the-art general segmentation methods and specialized methods.

Code — https://github.com/LTayfaker/FloorPlanFormer

1 Introduction

The digital transformation of the real estate industry is critical to automated floor plan analysis (McGuire and Schiffer 1983; Liu et al. 2017). Traditional floor plan recognition typically relies on image processing techniques such as binarization, Hough line detection, and Canny edge extraction. All these traditional methods (Tombre et al. 2002; Ahmed et al. 2011) depend heavily on manually set thresholds and see their accuracy plummet when confronted with extreme or non-standard floor plan layouts.

With the advent of deep convolutional networks such as AlexNet and ResNet (He et al. 2016), CNN-based architectures have become prevalent in floor plan recognition. The method (Liu et al. 2015) formulates the floor plan reconstruction problem as a Markov random field inference problem. A multi-task hourglass-style ResNet model simultaneously segments rooms and icons (Kalervo et al. 2019). Subsequent studies (Zeng et al. 2019; Pizarro et al. 2022) have employed VGG or ResNet encoders as well as dual decoders for boundary and room types.

The advent of transformer in 2017 pushed deep learning to a new stage. Soon after, researchers (Yue et al. 2022) applied a Transformer encoder–decoder with CNN backbone for end-to-end floor plan segmentation; building on this (Fan et al. 2021), (Fan et al. 2022) introduced panoramic symbol recognition by tokenizing graphical primitives for ViT input and adding a dual-branch head. The method (Xie et al. 2025) combines the Swin Transformer with an FPN and the semi-supervised fixmatch framework, though it relies on specific datasets. Most transformer-based methods focus on specific elements or layouts, but they validate the transformer's effectiveness for floor plan recognition.

To address these challenges, we introduce a transformer-based floor plan recognition method called FloorPlanFormer, which begins by organizing the elements of the floor plan into a hierarchical label scheme: walls and windows form one category, while inner contours are subdivided by room function—for detailed Fig.1. We propose a three-stage multi-task network for floor plan recognition. First, a Swin Transformer backbone extracts multi-scale features, and a pixel decoder produces high-resolution embeddings. Second, the outer contour branch uses visual prompting, a cross-attention decoder, and a Global Contextual Attention Module combining self-attention and convolution to refine polygon masks. Third, the inner contour branch employs a masked decoder and a Masked Feature Refinement Module, which uses the outer mask as an attention and directional convolution to capture room-boundary structures. Finally, we introduce FloorPlan8K with 8200 images and 77434 finely annotated instances. Our main contributions are as follows:

- We propose FloorPlanFormer, a novel three-stage floor plan recognition method, which employs a multi-task learning strategy and transformer architecture components to build a network that can effectively decouple the recognition targets and segment them efficiently.
- We propose a novel GCAM module and an efficient MFRM module that can efficiently capture global information and target local spatial relationships for full-map interaction and cross-task interaction.
- We construct FloorPlan8K, a floor plan dataset with rich

*Corresponding author.

Figure 1: Floor plan styles and hierarchical labeling scheme for our proposed FloorPlan8K.

scale and variety for model training and evaluation, with fine-grained labeling.

2 Related Works

Floor plan recognition builds upon image segmentation, a fundamental task in computer vision that provides essential support for automating the analysis of architectural layouts. In the following, we review related work from two perspectives: general image segmentation and floor plan recognition.

2.1 Image Segmentation

Traditional methods use task-specific models and losses. Instance segmentation generates binary masks per object and is done via two-stage (detect then segment) or one-stage (direct mask) approaches. In the early two-stage segmentation method (Cai and Vasconcelos 2019; Chen et al. 2019) , the segmentation region is determined based on the bounding box predicted by the detection model, and then the instance targets within the region are predicted. One-stage methods (Bolya et al. 2019; Wang et al. 2020a,b) remove the candidate region generation and feature re-pooling steps, which simplifies model training and achieves relatively high real-time performance. Semantic segmentation classifies each pixel into a semantic category, a problem first addressed by FCN (Long, Shelhamer, and Darrell 2015) and later refined by models such as DeepLab (Chen et al. 2017a) and Rethinking Atrous Convolution for Semantic Image Segmentation (Chen et al. 2017b). Panorama segmentation (Kirillov et al. 2019; Wang et al. 2021) are then a combination of these two segmentation tasks, to segment foreground instances and background semantics together. DETR (Carion et al. 2020) introduces end-to-end mask prediction with transformers, and Mask2former (Cheng et al. 2022) extends this with mask attention to excel at three segmentation tasks.

2.2 Floor Plan Recognition

Automatic floor plan recognition is a challenging problem, and the problem can affect other domains (Qiuchen Lu et al. 2019; Gaitanis et al. 2023). Researchers first tackle floor plan recognition using classical segmentation and classification methods (Koutamanis and Mitossi 1992). After Alexnet, the great potential of deep segmentation networks was demonstrated (Haskins, Kruger, and Yan 2020). A learnable module (Liu et al. 2017) was first added to the segmentation network to balance the contribution of multi-task losses. A deep multi-task neural network (Zeng et al. 2019) was designed to recognize floor plan elements using VGG, and there are some subsequent approaches (Xu et al. 2021, 2025) that refer to this idea. Transformer architectures have recently demonstrated strong promise for image segmentation. As Vit (Dosovitskiy et al. 2020) and Swin Transformer (Liu et al. 2021) have made a big splash in the vision field, (Fan et al. 2022; Huang et al. 2023; Xie et al. 2025) have applied them to various tasks in the floor plan field. While (Yue et al. 2022) method using a standard transformer structure. Existing transformer-based methods face two main issues: CNN-based models miss fine edge details due to lim-

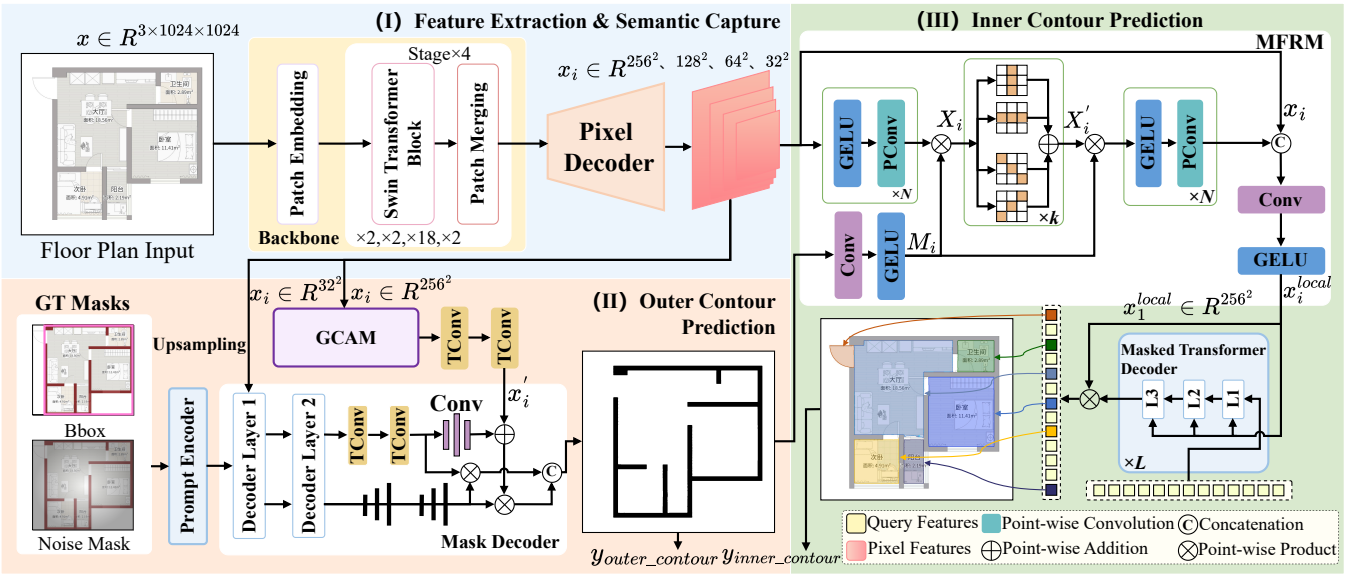


Figure 2: Overview of FloorPlanFormer, which is based on a multi-task learning strategy to recognize the inner contours (including the entrance door) and outer contours of a floor plan in three stage.

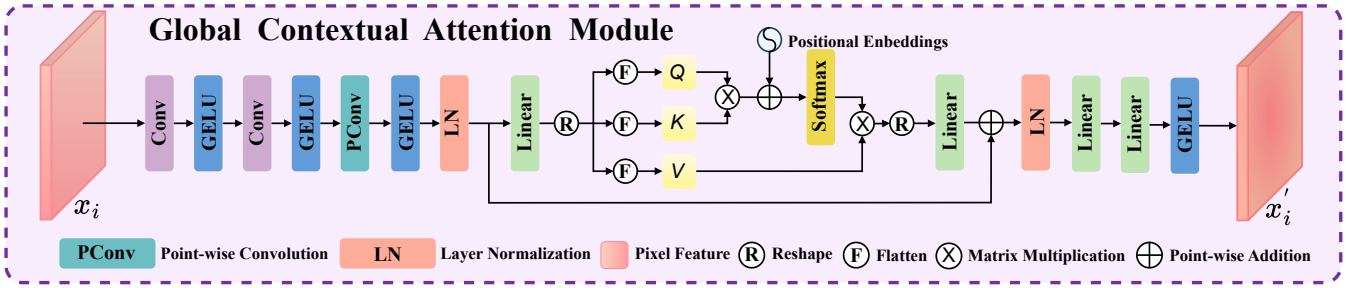


Figure 3: Overview of Global Contextual Attention Module, which is used to extract global semantic associations from early features.

ited complexity and generalization, while pure Transformer models overlook local spatial relations, leading to overlapping elements and irregular boundaries.

3 Methods

3.1 Goals and Problem

This study aims to accurately identify and organize multi-dimensional spatial semantic elements in floor plans by classifying components like walls, doors, and rooms, filtering transitional doors to find entrances, and categorizing rooms hierarchically.

Challenges include ambiguous boundaries causing misclassification and complex spatial dependencies, such as rooms near load-bearing walls, which both aid and complicate feature modeling.

3.2 Overall Network Architecture

Our overall network structure is shown in Fig.2. The network first employs a transformer-based visual encoder for feature extraction and downsampling of the input. Subsequently, a pixel encoder is introduced to refine the above multi-scale features to enhance the semantic associations between the features at each level. Then, these refined features are decoded to output the corresponding segmentation prediction results.

Feature Extraction and Semantic Capture. In the first stage of the network (top left of Fig.2), we introduce the Swin Transformer (Liu et al. 2021) as the backbone to learn hierarchical, cross-region feature associations: the input $x \in \mathbb{R}^{C\times H\times W}$ is first divided by patch embedding into $\frac{H}{4} \times \frac{W}{4}$ non-overlapping patches of size 4×4, producing a token sequence that is processed by four BasicLayers, each comprising an even number of Swin Transformer blocks followed by a Patch Merging module for downsampling.

After these layers, we obtain multi-scale feature maps $x_1 \in \mathbb{R}^{C\times \frac{H}{4}\times \frac{W}{4}}$, $x_2 \in \mathbb{R}^{C\times \frac{H}{8}\times \frac{W}{8}}$, $x_3 \in \mathbb{R}^{C\times \frac{H}{16}\times \frac{W}{16}}$, and $x_4 \in \mathbb{R}^{C\times \frac{H}{32}\times \frac{W}{32}}$. These maps are then fed into a pixel decoder (Cheng, Schwing, and Kirillov 2021) : we apply 1×1 convolutions to x_2, x_3, and x_4, flatten them into

embeddings, concatenate the results, and input them to a multi-scale deformable attention Transformer encoder for fine-grained representation of features at the pixel level.

Outer Contour Prediction. In the second stage (bottom left of Fig.2) , we first refine $x_1 \in \mathbb{R}^{C\times\frac{H}{4}\times\frac{W}{4}}$ via the GCAM module to obtain $x_1' \in \mathbb{R}^{C\times\frac{H}{4}\times\frac{W}{4}}$ through transposed convolution. Simultaneously, we upsampling $x_4 \in \mathbb{R}^{C\times\frac{H}{32}\times\frac{W}{32}}$ and feed it into a two-layer mask decoder (Kirillov et al. 2023; Ke et al. 2024; Ravi et al. 2024) , guided by bounding box and noise mask prompts from the ground true mask, to predict the outer contour. Within each decoder layer, the upsampled x_4 is combined with positional encodings to form keys and values for cross-attention, while prompt tokens serve as queries, producing class logits y_{cls} and mask embeddings $y_{mask} \in \mathbb{R}^{C\times\frac{H}{16}\times\frac{W}{16}}$. We then transposed convolution y_{mask} to produce $y'_{mask} \in \mathbb{R}^{C\times\frac{H}{4}\times\frac{W}{4}}$, apply two padded convolutions to inflate and recompress its channels, and fuse this result with x_1' to obtain x_1''. Finally, y_{cls} is decoded by a two-layer MLP and concatenated with both y'_{mask} and x_1'' to generate the final outer contour mask $y_{outer\_contour}$.

Inner Contour Prediction. In the third phase (right of Fig.2) , the four-layer multi-scale features and the stage two output $y_{outer\_contour}$ are fed into the MFRM, yielding refined features $x_1^{local} \in \mathbb{R}^{C\times\frac{H}{4}\times\frac{W}{4}}$, $x_2^{local} \in \mathbb{R}^{C\times\frac{H}{8}\times\frac{W}{8}}$, $x_3^{local} \in \mathbb{R}^{C\times\frac{H}{16}\times\frac{W}{16}}$, $x_4^{local} \in \mathbb{R}^{C\times\frac{H}{32}\times\frac{W}{32}}$.These features then enter a masked transformer decoder comprising L blocks, each with three decoder layers corresponding to scales $\frac{H}{8}$, $\frac{H}{16}$, and $\frac{H}{32}$. Within each layer, a learnable positional encoding is added to x_i^{local} $(i = 2, 3, 4)$ to serve as the key and value for cross-attention. After stacking these layers L times—from smallest to largest resolution, a learnable output query, together with x_4^{local}, is passed through the prediction head to generate an initial attention mask, which the masked attention mechanism refines into the final inner-contour output. We formulate the computation of mask-based attention as follows:

$$\mathbf{F}_l = \mathbf{F}_{l-1} + softmax(\mathbf{Q}_l\mathbf{K}_l^{\mathrm{T}} + \mathcal{M}_{l-1})\mathbf{V}_l. \quad (1)$$

Moreover, $\mathcal{M}$ is the attention mask on the feature, which can be expressed as:

$$\mathcal{M}_l(x,y) = \begin{cases} 0 & if \quad M_l(x,y) = 1 \\ -\infty & other \end{cases}. \quad (2)$$

Here, l denotes the layer index. In the l^{th} layer, $\mathbf{F}_l \in \mathbb{R}^{N\times C}$ represents $N \times C$ query features, while $\mathbf{K}^l, \mathbf{V}^l \in \mathbb{R}^{H_lW_l\times C}$ are the image embedding keys and values. The decoder input $\mathbf{Q}_l \in \mathbb{R}^{N\times C}$ is obtained by passing $\mathbf{F}_{l-1}$ through a feed-forward network. Each layer predicts a binary mask $M_l \in \{0,1\}^{N\times H_lW_l}$, with M_0 derived directly from $\mathbf{F}_0$.

3.3 Masked Feature Refinement Module

The floor plan outer contours and inner contour have strong local dependencies. To capture these dependencies, we introduce the Masked Feature Refinement Module, shown on the top right of Fig.2.

The Masked Feature Refinement Module has two branches: the top branch takes in learned multi-scale image embeddings, while the bottom branch receives the input outer contour mask. The bottom branch applies convolutions to downsample this mask in multi-scale attention maps $M_1 \in R^{C\times\frac{H}{4}\times\frac{W}{4}}$, $M_2 \in \mathbb{R}^{C\times\frac{H}{8}\times\frac{W}{8}}$, $M_3 \in \mathbb{R}^{C\times\frac{H}{16}\times\frac{W}{16}}$, and $M_4 \in \mathbb{R}^{C\times\frac{H}{32}\times\frac{W}{32}}$. In the top branch, the embedding of each scale is first compressed along the channel dimension via a GELU activation followed by a pointwise convolution. We then apply each attention map twice—denoted by "X" in the figure—to these compressed features, producing refined outputs $X_1 \in \mathbb{R}^{C\times\frac{H}{4}\times\frac{W}{4}}$, $X_2 \in \mathbb{R}^{C\times\frac{H}{8}\times\frac{W}{8}}$, $X_3 \in \mathbb{R}^{C\times\frac{H}{16}\times\frac{W}{16}}$ The "X" operation itself is defined as:

$$X_i = x_i \cdot M_i, \quad i = 1, 2, 3, 4. \quad (3)$$

We then apply multi-directional convolution to extract and fuse edge information from each X_i. Specifically, we use four fixed 3×3 kernels—horizontal C_h, vertical C_v, diagonal C_d, and anti-diagonal C_f—to capture localized edges in each orientation. This set of convolutions is repeated k times to progressively refine the boundary features. Denoting the result as X_i', the operation over k iterations can be written as:

$$\sum_k (C_h + C_v + C_d + C_f) * X_i. \quad (4)$$

Finally, each refinement feature $X_i^{'}(i = 1, 2, 3, 4)$ is reweighted by its corresponding attention mask M_i and then concatenated with the original multiscale features along the channel dimension. Subsequent 1×1 convolutions recover the full channel semantics, and the final channel convolution fuses these combined features before passing them into the masked transform decoder.

3.4 Global Contextual Attention Module

To address Swin Transformer's limited ability to model full long-range dependencies, we design a module that enhances global feature interaction beyond local and cross-window attention (Fig.3 details our Global Contextual Attention Module) .

We first downsample the input high-resolution feature $x_1 \in \mathbb{R}^{C\times\frac{H}{4}\times\frac{W}{4}}$ fourfold through strided convolutions, then apply a 1×1 convolution to expand channel dimensions—shrinking the spatial embedding for efficiency while preserving feature richness. These compressed features are projected into query, key, and value vectors using linear layers. We scale the query by the downsampling factor and compute the attention scores via a query-key dot product, apply softmax with positional embeddings, and weight the value vectors to produce globally informed representations. A final linear projection and convolution yield the global context features, which are fed into the outer contour mask decoder to inject strong, long-range semantics, empirically improving outer contour accuracy.

3.5 FloorPlan8K Dataset

We gather nearly 10000 floor plans of Chinese buildings from various property companies and obtain 8,200 representative and high-quality floor plans after filtering, cleaning

Metrics	Categories	DFPR	Mask2former	Ours	**Ours++**	Mask2former	Ours	**Ours++**
mIoU	all categories	0.793	0.915	0.919	**0.935**	*Unique Coverage Accu.*		
Total Accu.	all categories	0.857	0.957	0.951	**0.966**	0.899	0.943	**0.962**
	outer_contour	0.890	0.948	**0.953**	0.944	0.588	**0.953**	0.944
	kitchen	0.916	0.968	0.981	**0.987**	0.961	0.975	**0.986**
	restaurant	0.569	0.727	0.811	**0.840**	0.692	0.764	**0.819**
	bathroom	0.905	0.975	0.966	**0.984**	0.967	0.963	**0.983**
	bedroom	0.912	0.969	0.971	**0.979**	0.968	0.964	**0.979**
	cloakroom	0.241	0.674	0.625	**0.894**	0.585	0.625	**0.894**
Class Accu.	living_room	**0.977**	0.903	0.908	0.934	0.890	0.897	**0.918**
	wardrobe	0.526	0.938	0.957	**0.975**	0.854	0.919	**0.933**
	master_bedroom	0.890	**0.975**	0.945	0.954	**0.964**	0.929	0.954
	verandah	0.901	0.970	0.962	**0.976**	0.969	0.960	**0.974**
	guest_restaurant	0.897	**0.989**	0.968	**0.989**	0.961	0.964	**0.988**
	other_room	0.663	0.890	0.883	**0.947**	0.866	0.851	**0.932**
	entrance_door	0.744	**0.964**	0.943	0.963	0.945	0.932	**0.958**

Table 1: We compare FloorPlanFormer with current general segmentation methods and specialized floor plan recognition methods for each metric. Top performances are marked in **blod**.

and labeling, which contain three types of forms: two-color (black and white), undecorated line drawings, and fully decorated floor plans. Our hierarchical labeling scheme covers three element types: (1) outer contours: walls and windows are combined into one category and labeled as an ordered set of points encoding shape, position, and thickness (following the COCO guideline (Lin et al. 2014)); (2) entrance doors: labeled as a fan shape to indicate the position and swing range of the door while capturing the direction of entrance of the occupants and the door geometry; and (3) inter contours: rooms enclosed by outer contours, categorized into ten common types (kitchens, bathrooms, bedrooms, etc.), rarely labeled as "room", "kitchen", "bathroom", etc. (see Fig.1).

3.6 Training

Single-task loss. In the second stage, we supervise mask prediction using cross-entropy loss and dice loss (Milletari, Navab, and Ahmadi 2016), defining the final branch loss as:

$$\mathcal{L}_o = \lambda_{oce}\mathcal{L}_{ce} + \lambda_{odice}\mathcal{L}_{dice}. \tag{5}$$

In the third stage, mask prediction is supervised by cross-entropy loss and dice loss (Milletari, Navab, and Ahmadi 2016), and the overall branch loss combines these mask losses with classification loss:

$$\mathcal{L}_{rd} = \lambda_{rce}\mathcal{L}_{ce} + \lambda_{rdice}\mathcal{L}_{dice} + \lambda_{cls}\mathcal{L}_{cls}. \tag{6}$$

Multi-task loss. As the two tasks are independent of each other but potentially linked, it is critical to balance the importance between the tasks. We compute the corresponding weights based on the number of pixel points contained in each task in the image, and finally obtain the final loss of the overall network by joint weighted summation. Specifically, the joint multi-task loss can be expressed as:

$$\mathcal{L} = \lambda_{rd}\mathcal{L}_{rd} + \lambda_o\mathcal{L}_o. \tag{7}$$

Here, λ_{rd} and λ_o can also be expressed as:

$$\lambda_{rd} = \frac{N_o}{N_{rd} + N_o} \quad \text{and} \quad \lambda_o = \frac{N_{rd}}{N_{rd} + N_o}. \tag{8}$$

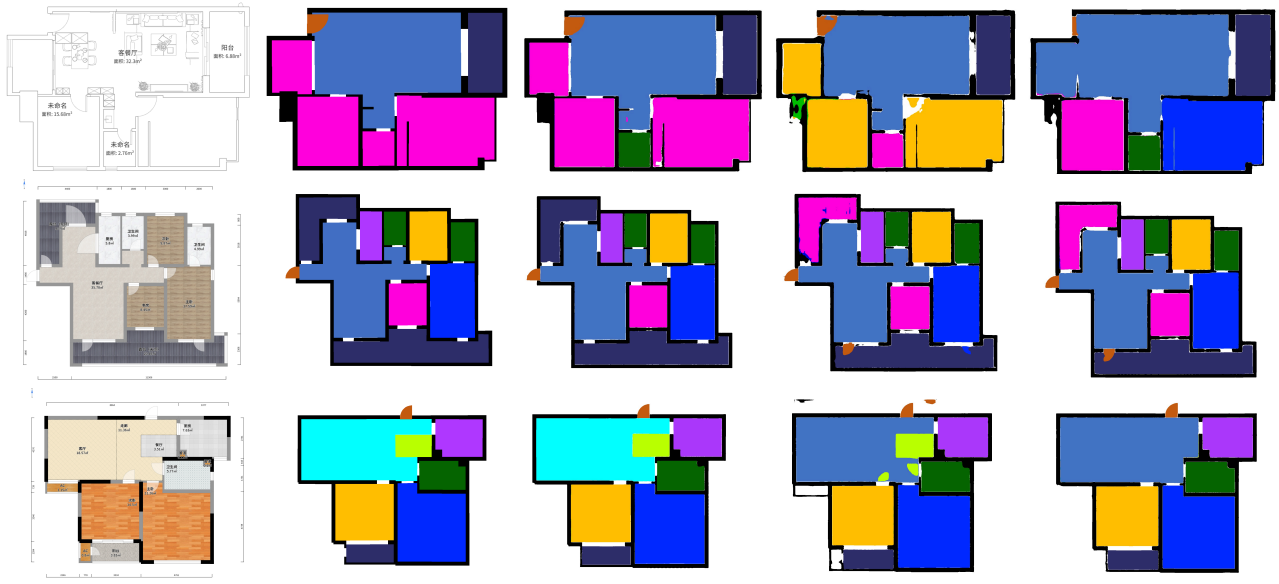
where N_{rd} is the total number of pixel points in the inner contour and entrance door in the original image. N_o is the total number of pixel points in the outer contour.

4 Experiments

We conduct extensive experiments on the FloorPlan8K dataset, benchmarking our approach against state-of-the-art methods and performing ablation studies to assess the contribution of each module.

4.1 Implementation Details

Module configuration. We adopt SwinT-S as our backbone to maximize feature accuracy, and employ a multi-scale deformable-attention transformer (Zhu et al. 2020) as the pixel decoder. In the third stage, our masked transformer encoder uses three layers—each with three attention scales—100 queries by default, and embeds the attention mask into intermediate outputs for guided learning. Within the Masked Feature Refinement Module, we apply channel convolution with GELU activation and repeat directional convolutions ($N = 3$, $k = 2$) to extract precise room-boundary details.



(a) Input (b) Ground turth (c) FloorPlanFormer (ours) (d) DeepFloorplan (e) Mask2former

Figure 4: Visualisation of FloorPlanFormer on the FloorPlan8K with the floor plan recognition results of other comparative methods.

Training. We train for 40 epochs on the FloorPlan8K training split using a single NVIDIA A6000 GPU with a batch size of 4. Model parameters are optimized via AdamW (Loshchilov and Hutter 2017) with an initial learning rate of 1e-4, decayed by a factor of 0.1 at epochs 20 and 30. Baseline methods are retrained using the hyperparameter settings reported in their original papers.

4.2 Comparison with DeepFloorplan

We first compare with DeepFloorplan (Zeng et al. 2019), a leading floor plan recognition method. To ensure fairness, we adopt its training protocol and optimize its hyperparameters as described in (Zeng et al. 2019). Fig.4 presents side-by-side visual results: FloorPlanFormer consistently delivers crisper and more accurate segmentation of walls, doors, and rooms than DeepFloorplan, particularly in areas with complex layouts and overlapping elements.

For quantitative assessment, we use two common metrics(Long, Shelhamer, and Darrell 2015; Zeng et al. 2019; Lv et al. 2021) for this direction:

$$Accu = \frac{\sum_i N_i}{\sum_i \hat{N}_i}, \tag{9}$$

$$Accu(i) = \frac{N_i}{\hat{N}_i}. \tag{10}$$

Here, N_i and $\hat{N}_i$ denote the number of correctly predicted pixels and the total ground-truth pixels for the i^{th} category, respectively. Tab.1 reports quantitative results on the FloorPlan8K dataset, comparing FloorPlanFormer with DeepFloorplan; our method achieves higher accuracy in most target classes.

4.3 Comparison with Segmentation Methods

We compare Mask2former (Cheng et al. 2022) with our method on the FloorPlan8K dataset, and present the visual results side by side in Fig.4.

Current generic segmentation models often over-segment connected regions—e.g., splitting one room into multiple correctly classified fragments, but their union can still match the ground truth closely enough so that standard accuracy metrics (Zeng et al. 2019) fail to expose errors. So we introduce Unique Coverage Accuracy metrics, which evaluate segmentation contiguity rather than per-fragment correctness. The unique coverage(UC) overall accuracy and unique coverage(UC) per-class accuracy are defined as:

$$Accu^{UC} = \frac{\sum_i (N_i - N_i^{overlap})}{\sum_i \hat{N}_i}, \tag{11}$$

$$Accu^{UC}(i) = \frac{N_i - N_i^{overlap}}{\hat{N}_i}. \tag{12}$$

Here, N_i and $\hat{N}i$ denote the number of correctly predicted pixels and the total ground truth pixels for the i^{th} class, respectively, while $Ni^{overlap}$ represents the count of pixels that the model predicts more than once for that class. Tab.1 presents a quantitative comparison on the FloorPlan8K dataset: FloorPlanFormer consistently outperforms Mask2former (Cheng et al. 2022) and other baselines across most semantic categories, achieving higher per-class and overall recognition accuracy.

4.4 Ablation Experiments

Next, we experimentally analyze the network structure, the Global Contextual Attention Module and the Masked Feature Refinement Module.

Settings			Metrics		
GCAM	MFRM(L)	MFRM(S)	$Total\ Accu.$	$mIoU$	$UC\ Total\ Accu.$
-	-	-	0.951	0.919	0.943
✓	-	-	0.959	0.931	0.954
-	✓	-	0.960	0.929	0.949
-	-	✓	0.964	0.932	0.956
✓	-	✓	**0.966**	**0.935**	**0.962**

Table 2: Ablation experiments of Global Contextual Attention Module and Masked Feature Refinement Module.

Multi-tasking network. We train our multi-task network on FloorPlan8K without the Global Contextual Attention Module and the Masked Feature Refinement Module to benchmark against Mask2former (Cheng et al. 2022). Tab.1 shows that even this base version outperforms Mask2former, confirming that our architecture more effectively addresses floor plan recognition.

Module analysis. We analyze the contributions of the Global Contextual Attention Module and the Masked Feature Refinement Module, as detailed in Fig.2 and Fig.3, by conducting ablation studies on the FloorPlan8K dataset.

We extract edge features from multi-scale embeddings via multi-directional convolution in two ways:

- Large kernels: a single set of oversized, multi-scale filters processes all embedding scales.
- Small kernels: multiple sets of smaller filters independently operate at each embedding scale.

From Tab.2, we see that adding both the Global Contextual Attention Module and the Masked Feature Refinement Module, together with small-kernel multi-directional convolutions, improves all performance metrics. This result shows that capturing the interaction between global context and inner-outer contours is essential for our multi-task network.

Loss functions analysis. We evaluate four loss configurations in Tab.3. Keeping the classification loss fixed, we first remove the dice loss (Milletari, Navab, and Ahmadi 2016), which causes the largest drop in all metrics, indicating its primary importance. Removing the cross-entropy loss yields a smaller decline, showing it serves as a supportive role. Finally, adding the joint multi-task loss achieves the best overall performance by automatically balancing each task's contribution. We confirm that dice loss is indispensable, cross-entropy loss provides secondary supervision, and the joint loss further enhances metric consistency across tasks.

Settings				Metrics		
$\mathcal{L}_{cls}$	$\mathcal{L}_{ce}$	$\mathcal{L}_{dice}$	$\mathcal{L}_{multi\_task}$	$TotalAccu.$	$mIoU$	$UC\ TotalAccu.$
✓	✓	-	-	0.951	0.919	0.943
✓	-	✓	-	0.960	0.917	0.953
✓	✓	✓	-	0.964	0.922	0.951
✓	✓	✓	✓	**0.966**	**0.935**	**0.962**

Table 3: Loss function analysis. We conducted analytical experiments on the loss function combinations used by FloorPlanFormer to demonstrate their effectiveness.

Settings		Metrics		
N	k	$Total\ Accu.$	$mIoU$	$UC\ Total\ Accu.$
1	1	0.959	0.919	0.946
2	1	0.957	0.924	0.951
3	1	0.959	0.930	0.952
1	2	0.963	0.931	0.955
2	2	0.958	0.922	0.946
3	2	**0.966**	**0.935**	**0.962**
3	3	**0.966**	0.934	0.958
4	2	0.964	0.933	0.958

Table 4: Experiments with the Mask Feature Refinement Module. We performed experiments comparing the number of overlaps k and N for multi-directional convolution and channel convolution.

Study on Masked Feature Refinement Module. We perform ablation experiments on the convolutions of the Masked Feature Refinement Module: repeating multi-direction convolution k times and channel expansion convolution N times. Tab.4 shows that $k = 2$ and $N = 3$ yield the best performance. Repeating multi direction convolution twice expands the receptive field and strengthens boundary correlations, while three channel expansion convolutions prevent excessive channel scaling and preserve high frequency details

4.5 Discussion

Application. We extract concave and convex corners from predicted contours to define the building's footprint, apply geometric rules to inner and outer masks to refine protrusions and recesses, and feed these precise outputs into downstream tasks like 3D modeling, rendering, layout automation, and facility management.

Limitation. Our network struggles with open plan layouts, such as combined kitchens and dining areas—because it depends on clear boundaries to separate rooms, and it misclassifies unlabeled or undecorated rooms due to scarce training examples and low feature variability for those types.

5 Conclusion

We introduce FloorPlanFormer, a three-stage transformer network that efficiently recognizes floor plan elements. We build FloorPlan8K, a dataset of 8200 finely annotated floor plans, to train and evaluate our model. We employ a multi-task learning framework that jointly predicts outer contours, inner contours, and entrance doors. We enhance outer-contour learning with a Global Contextual Attention Module that captures long-range structural dependencies. We refine inner-contour predictions using a Masked Feature Refinement Module that leverages local relationships between outer and inner contours. Extensive experiments show our model consistently outperforms existing methods, and we next focus on optimizing it for real-time deployment.

Acknowledgments

This research was supported by the National Key Research and Development Program of China (2024YFC2814901), Natural Science Foundation of Guangdong Province (2025A1515010803), and Key R&D Project of Guangzhou Science and Technology Plan (2025B01J0001).

References

Ahmed, S.; Weber, M.; Liwicki, M.; and Dengel, A. 2011. Text/Graphics Segmentation in Architectural Floor Plans. In *2011 International Conference on Document Analysis and Recognition*.

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9157–9166.

Cai, Z.; and Vasconcelos, N. 2019. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5): 1483–1498.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. 2019. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4974–4983.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4): 834–848.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017b. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention Mask Transformer for Universal Image Segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cheng, B.; Schwing, A.; and Kirillov, A. 2021. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34: 17864–17875.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*.

Fan, Z.; Chen, T.; Wang, P.; and Wang, Z. 2022. CADTransformer: Panoptic Symbol Spotting Transformer for CAD Drawings. *CVPR*.

Fan, Z.; Zhu, L.; Li, H.; Chen, X.; Zhu, S.; and Tan, P. 2021. FloorPlanCAD: A Large-Scale CAD Drawing Dataset for Panoptic Symbol Spotting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Gaitanis, A.; Lentzas, A.; Tsoumakas, G.; and Vrakas, D. 2023. Route planning for emergency evacuation using graph traversal algorithms. *Smart Cities*, 6(4): 1814–1831.

Haskins, G.; Kruger, U.; and Yan, P. 2020. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31(1): 8.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, L.; Wu, J.-H.; Wei, C.; and Li, W. 2023. MuraNet: Multi-task Floor Plan Recognition with Relation Attention. In *International Conference on Document Analysis and Recognition*, 135–150. Springer.

Kalervo, A.; Ylioinas, J.; Häikiö, M.; Karhu, A.; and Kannala, J. 2019. Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis. In *Image Analysis: 21st Scandinavian Conference, SCIA 2019, Norrköping, Sweden, June 11–13, 2019, Proceedings 21*, 28–40. Springer.

Ke, L.; Ye, M.; Danelljan, M.; Tai, Y.-W.; Tang, C.-K.; Yu, F.; et al. 2024. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9404–9413.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Koutamanis, A.; and Mitossi, V. 1992. Automated recognition of architectural drawings. In *1992 11th IAPR International Conference on Pattern Recognition*, volume 1, 660–663. IEEE Computer Society.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, C.; Schwing, A. G.; Kundu, K.; Urtasun, R.; and Fidler, S. 2015. Rent3D: Floor-plan priors for monocular layout estimation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, C.; Wu, J.; Kohli, P.; and Furukawa, Y. 2017. Raster-to-Vector: Revisiting Floorplan Transformation. In *2017 IEEE International Conference on Computer Vision (ICCV)*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully Convolutional Networks for Semantic Segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. *Learning,Learning*.

Lv, X.; Zhao, S.; Yu, X.; and Zhao, B. 2021. Residential floor plan recognition and reconstruction. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

McGuire, R. H.; and Schiffer, M. B. 1983. A theory of architectural design. *Journal of anthropological archaeology*, 2(3): 277–303.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*.

Pizarro, P. N.; Hitschfeld, N.; Sipiran, I.; and Saavedra, J. M. 2022. Automatic floor plan analysis and recognition. *Automation in Construction*, 140: 104348.

Qiuchen Lu, V.; Parlikad, A. K.; Woodall, P.; Ranasinghe, G. D.; and Heaton, J. 2019. Developing a dynamic digital twin at a building level: Using Cambridge campus as case study. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*, 67–75. ICE Publishing.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Tombre, K.; Tabbone, S.; Pélissier, L.; Lamiroy, B.; and Dosch, P. 2002. Text/graphics separation revisited. In *Document Analysis Systems V: 5th International Workshop, DAS 2002 Princeton, NJ, USA, August 19–21, 2002 Proceedings 5*, 200–211. Springer.

Wang, H.; Zhu, Y.; Adam, H.; Yuille, A.; and Chen, L.-C. 2021. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5463–5474.

Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; and Li, L. 2020a. Solo: Segmenting objects by locations. In *European conference on computer vision*, 649–665. Springer.

Wang, X.; Zhang, R.; Kong, T.; Li, L.; and Shen, C. 2020b. Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems*, 33: 17721–17732.

Xie, H.; Ma, X.; Mei, Q.; and Chui, Y. H. 2025. A semi-supervised approach for building wall layout segmentation based on transformers and limited data. *Computer-Aided Civil and Infrastructure Engineering*, 40(10): 1295–1313.

Xu, Z.; Yang, C.; Alheejawi, S.; Jha, N.; Mehadi, S.; and Mandal, M. 2021. Floor plan semantic segmentation using deep learning with boundary attention aggregated mechanism. In *Proceedings of the 2021 4th International Conference on Artificial Intelligence and Pattern Recognition*, 346–353.

Xu, Z.; Yang, C.; Alheejawi, S.; Jha, N.; Mehadi, S.; and Mandal, M. 2025. Automatic floor plan analysis using a boundary attention-based deep network. *International Journal on Document Analysis and Recognition (IJDAR)*, 28(1): 19–30.

Yue, Y.; Kontogianni, T.; Schindler, K.; and Engelmann, F. 2022. Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. *2023CVPR*.

Zeng, Z.; Li, X.; Yu, Y. K.; and Fu, C.-W. 2019. Deep Floor Plan Recognition Using a Multi-Task Network with Room-Boundary-Guided Attention. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*.