Journal of Science and Technology in Civil Engineering, HUCE, 2026, 20 (4): 68–80

AUTOMATIC SEGMENTATION AND ARCHITECTURAL SPACE RECOGNITION FROM COMPLEX DRAWINGS USING DEEP LEARNING

Do Quoc Hoang[a], Nguyen Hoang Hai[a], Tran Van Phuong[c], Tran Anh Binh[a,*], Pham Van Hoan[b], Nguyen Thi Hai Nhu[b]

[a]*Institute of Information Technology in Civil Engineering, Hanoi University of Civil Engineering, 55 Giai Phong road, Bach Mai ward, Hanoi, Vietnam*
[b]*Faculty of Information Technology, Hanoi University of Civil Engineering, 55 Giai Phong road, Bach Mai ward, Hanoi, Vietnam*
[c]*Former Scientist at Amazon, 1100 Enterprise Way, Sunnyvale, CA 94089, United States*

Article history:
Received 16/04/2026, Revised 04/9/2026, Accepted 21/9/2026, Available online 16/10/2026, Version of Record 25/12/2026

Abstract

While Building Information Modeling (BIM) is becoming a mandatory standard in the construction industry, the automated conversion of legacy 2D technical drawings into structured 3D BIM models remains a critical bottleneck. A primary challenge in this 2D-to-BIM pipeline is the initial semantic extraction: translating unstructured, cluttered graphical data into clean, topologically accurate pixel-level representations. Traditional automated segmentation methods frequently fail to maintain strict geometric accuracy, producing fragmented physical boundaries and topological errors that hinder subsequent vectorization and 3D reconstruction pipelines. To address these extraction challenges, this research proposes an improved solution based on a multi-task semantic segmentation architecture to simultaneously recognize physical boundaries and spatial functions. To overcome the limitations of traditional neural networks in maintaining geometric accuracy, we introduce an Edge-aware Loss function based on the Sobel operator. This function acts as a spatial regularizer, forcing the model to produce sharp boundaries that strictly adhere to the drawing's topological structure. Furthermore, the study experimentally evaluates the replacement of various feature extraction backbones, including VGG16, ResNet, and ConvNeXt. Experimental results on the standard CubiCasa5K dataset show that the model using the ConvNeXt-tiny backbone combined with Edge-aware Loss achieves superior performance with an mIoU score of 78.57% for the boundary task and for the room space task of 72.27%. Compared to the original VGG16-based architecture, the proposed configuration improves Boundary mIoU by 9.8 percentage points (from 68.78% to 78.57%) and Room mIoU by 17.6 percentage points (from 54.70% to 72.27%). These results confirm the practical value of the method for producing high-quality segmentation maps that can serve as input for vectorization and automatic reconstruction of building models.

Keywords: semantic segmentation; floor plan recognition; edge-aware loss; ConvNeXt; deep learning; CubiCasa5K; multi-task learning; boundary detection.

1. Introduction

As Building Information Modeling (BIM) becomes a mandatory standard in the lifecycle of construction projects [1, 2], the need to automate the extraction of information from 2D drawings into a structured 3D model format (2D-to-BIM) is increasingly critical. In the early stages, research often approached the problem using heuristic methods based on line drawing rules [3]. However, these

*Corresponding author. *E-mail address:* binhta@huce.edu.vn (Binh, T. A.)
https://doi.org/10.31814/stce.huce2026-20(4)-05 p-ISSN 1859-2996; e-ISSN 2734-9268
© 2026 The Author(s). Published by Hanoi University of Civil Engineering. Licensed under CC BY-NC-ND 4.0.

traditional techniques struggle to generalize and often fail when faced with real-world drawings that have high noise levels and diverse design styles. The rapid development of Convolutional Neural Networks (CNNs) [4] has driven a new generation of deep learning solutions on benchmark datasets like CubiCasa5K [5]. We recognize that a semantic segmentation approach remains the most optimal and practical solution. By focusing on accurately classifying each pixel, segmentation models not only preserve complex geometric details but also provide a reliable probability map for subsequent processing steps.

To address the open boundary problem and automate the vectorization process, this research proposes an improved deep learning model for extracting local features from high-resolution images, serving as a prerequisite for combining with graph optimization algorithms in the post-processing step. Among recent deep learning-based approaches, Zeng et al. [6] introduced a multi-task neural network for floor plan recognition, in which physical boundaries and room types are predicted simultaneously. Their room-boundary-guided attention mechanism effectively exploits boundary information to improve spatial function recognition, making the model more suitable for complex floor plans than conventional single-task segmentation approaches. However, two limitations still remain. First, the original framework mainly optimizes pixel-wise segmentation accuracy through a cross-and-within-task weighted loss. Although this design addresses class imbalance, it does not explicitly constrain the geometric quality of the predicted boundaries. As a result, errors such as blurred wall edges, jagged contours, and locally fragmented line segments may still appear, which can negatively affect subsequent vectorization and BIM reconstruction. Second, the original architecture relies on VGG16 as the feature extraction backbone. While VGG16 is a classical CNN architecture, its limited receptive field and relatively high memory consumption restrict its ability to capture long-range spatial context in large and complex architectural drawings.

Motivated by these gaps, this study improves the framework of Zeng et al. [6] from two complementary aspects. First, an Edge-aware Loss based on the Sobel operator is introduced to explicitly enhance boundary sharpness, continuity, and geometric consistency. Second, the original VGG16 backbone is replaced and compared with more modern feature extraction networks, including ResNet34 [7] and ConvNeXt-Tiny [8], to improve spatial representation and computational efficiency.

The main contributions of this research are summarized as follows:

- Improved objective function: We propose integrating an Edge-aware Loss function into the existing technical drawing segmentation model architecture. This solution enhances the neural network's boundary perception capabilities, effectively addressing issues of blurred or jagged boundaries, thereby preserving the strict geometric properties of the drawings.

- Experimental study of feature extraction backbone: Conduct a comprehensive performance evaluation of the system by replacing the original VGG16 backbone with more modern convolutional architectures.

- Evaluation and comparison: Provide detailed quantitative and qualitative analyses, demonstrating that the combination of the proposed loss function and advanced backbones delivers superior accuracy compared to the original model.

2. Related work

2.1. Floor plan recognition based on rules

Before the deep learning era, floor plan analysis relied on low-level image processing (binarization, morphological operations) and heuristic rules to extract walls, doors, and symbols [3, 9], with some methods using shape grammars for style-specific reconstruction [10]. These approaches lacked generality, were sensitive to noise, and required heavy manual tuning [11]; they tend to fail on the

complex drawings typical of practical 2D-to-BIM scenarios, motivating the shift to learning-based methods.

2.2. Deep learning-based floor plan recognition

Convolutional neural networks (CNNs) substantially improved floor plan analysis. Liu et al. [12] used a deep segmentation network (ResNet-152) followed by integer programming to assemble primitives into a geometrically consistent vector representation, though this raster-to-vector formulation depends on a rigid optimization stage sensitive to upstream segmentation errors. Going beyond single-task segmentation, Zeng et al. [6] proposed a multi-task network predicting physical boundaries and spatial functions jointly; its room-boundary-guided attention (spatial context module) lets boundary features refine room-type prediction, performing well on complex drawings with curved walls. However, its cross-and-within-task weighted cross-entropy loss addresses class imbalance but does not constrain boundary geometry, so blurred edges, jagged contours, and fragmented segments can persist and degrade downstream vectorization.

More recent studies pursue structured recognition. Lv et al. [13] argue that recognition should support recovery of structured elements for reconstruction, not stop at pixel classification. Graph-based approaches predict vector-like primitives directly: Chen and Pan [14] parse junctions and line segments via a graph neural network, and Hu et al. [15] introduce Raster-to-Graph, an autoregressive attention transformer that predicts the floor plan graph jointly capturing geometry, topology, and semantics. These reduce post-processing but are sensitive to junction-detection errors and require accurate modeling of primitive relationships.

A parallel line of work improves raster-wise recognition through architecture and objective design. Xu et al. [16] add dedicated room-boundary and room-type decoders with a multiscale room-boundary attention model that even embeds directional kernels to handle non-axis-aligned boundaries, yet the network is still trained under a weighted cross-entropy loss that merely balances the two tasks by pixel count; Kratochvila et al. [17] pair an EfficientNet encoder with an affinity field loss - introduced because their unary focal loss lacks spatial discrimination - plus a heatmap term for openings; and De Nardin et al. [18] propose a multibranch, multiattention framework, reporting gains on R2V and R3D while naming ambiguous room boundaries as the dominant error source. These confirm that region-based losses alone are insufficient. Yet their mechanisms differ from what vectorization requires: attention re-weights features toward boundaries and affinity loss regularizes pairwise pixel relations, but neither directly penalizes the geometric fidelity of the predicted edge (sharpness, straightness, continuity). Notably, even FloorNet's directional kernels operate within the attention features rather than the loss, and Kratochvila et al. [17] report residual stair-like artifacts along contours from the discrete mask, while De Nardin et al. [18] likewise single out ambiguous boundaries as the persistent failure mode — reinforcing that awareness of boundary orientation at the feature level does not substitute for an explicit geometric penalty in the objective.

This distinction also separates our approach from PointRend [19], a general-purpose method that reframes segmentation as a rendering problem and sharpens contours by adaptively sampling points in uncertain boundary regions through an inference-time subdivision module. Significantly, its authors note that region-overlap metrics such as mIoU are biased toward object-interior pixels and relatively insensitive to boundary quality — which supports our choice to report a separate Boundary mIoU. Because PointRend refines boundaries via a sampling module attached to the backbone rather than through the training objective, it is complementary to our Edge-aware Loss, which enforces sharpness during training at no additional inference cost.

Transformer backbones address a complementary weakness, the limited receptive field and memory cost of Zeng's VGG16. Xie et al. [20] pair a Swin Transformer encoder with an FPN decoder and FixMatch semi-supervised training, segmenting walls accurately from limited labels. Their intent is orthogonal (attacking annotation scarcity), but the result is instructive: despite a focal loss and boundary-oriented augmentations (erosion, dilation), they report that residual errors cluster precisely at ambiguous, curved, and overlapping wall boundaries, evidence that a strong backbone and augmentation still do not guarantee sharp contours without an explicit geometric penalty on edges.

The remedy is to encode boundary quality into the objective. Kervadec et al. [21] showed that a boundary loss measuring the distance between predicted and ground-truth contours substantially helps highly unbalanced segmentation, confirming the value of an explicit boundary term. However, their distance-map formulation targets smooth, closed anatomical shapes and is costly to compute, whereas architectural drawings demand sharp, straight, orthogonal edges better matched by a lightweight gradient criterion. This motivates our Sobel-based Edge-aware Loss, which penalizes discrepancies between the gradient maps of prediction and ground truth, regularizing boundaries to be sharp and continuous at negligible cost.

In summary, although recent works strengthen backbones and enrich objectives with attention, affinity-field, or augmentation strategies, none explicitly penalizes edge geometry - the first gap our Edge-aware Loss targets. A second, baseline-specific gap is that Zeng et al. [6] still use a VGG16 backbone with limited receptive field and high memory cost, which we address with a modern backbone (ResNet34 / ConvNeXt-Tiny). Both improvements are built on Zeng's room-boundary-guided architecture, combining its spatial-context mechanism with an explicit geometric constraint and a stronger feature extractor.

2.3. Automatic floor plan generation

Automatic floor plan generation is a related but distinct direction. Early methods used graph- or raster-conditioned synthesis, alternating between vector and raster formats and losing geometric detail: Graph2Plan [22] generates plans from a layout graph and building boundary but its raster intermediate can introduce inaccuracies on conversion to CAD/BIM, while House-GAN++ [23] refines layouts via a graph-constrained GAN yet risks unstable training and offers no strict geometric or topological guarantee.

Diffusion-based methods now lead this task by generating vector plans directly: HouseDiffusion [24] preserves parallelism, orthogonality, and corner sharing through discrete and continuous denoising; Kuhn [25] adds conditioning on existing walls; and DiffPlanner [26] eliminates rasterization entirely via conditional diffusion and transformer modeling. This shift from raster synthesis toward vector representations reduces conversion loss and improves controllability. However, these methods create new plans rather than recognize existing drawings, and are thus complementary to ours: for 2D-to-BIM, accurate segmentation of existing drawings remains the prerequisite whose boundary and room quality determines downstream vectorization.

The recent survey of Xu et al. [11] organizes floor plan analysis (2000–2025) into four tasks — room-boundary, room-type, floorplan-object, and floorplan-text detection - noting that 2D-to-BIM systems usually combine several. Our study lies at the room-boundary / room-type intersection, predicting boundaries and functions jointly. The review also notes sensitivity to drawing-style diversity; e.g., Lu et al. [27] address rural-residential styles with a joint boundary-and-text network. This reinforces that accurate room-boundary detection is the foundational step for 3D BIM reconstruction - the stage our work strengthens.

3. Proposed method

3.1. Problem statement

The technical drawing segmentation system is modeled based on a dataset of images $D = \{I_n, Y_n\}_{n=1}^{N}$, where N is the total number of technical drawings in the dataset. For the n-th data sample:

$I_n \in R^{(H*W*C)}$ is the pixel matrix of the input drawing, with H, W being the height and width of the drawing, respectively. C is the number of color channels ($C = 1$ for black and white drawings, $C = 3$ for RGB drawings).

$Y_n \in \{0, 1, \ldots, K-1\}^{H \times W}$ is the corresponding ground truth label matrix.

- Each pixel (i, j) is assigned a label $y_{i,j} \in \{0, 1, \ldots, K-1\}$.

- K represents the total number of object classes to be segmented in the drawing (e.g., lines, symbols, text, or background).

Input: Training dataset $D_{train} \subset D$, a deep learning model with weights θ is initialized randomly, with training hyperparameters (learning rate $\propto$, number of epochs E).

Output: An optimized image segmentation model $f_{\theta^*} : R^{H*W*C} \rightarrow R^{H*W*K}$ representing the probability distribution $\widehat{P}$ for each pixel belonging to different classes. From this, the final prediction matrix $\widehat{Y} = argmax_{k \in Y} \widehat{P_{h,w,k}}$ is derived.

Objective: The objective of the problem is to find the optimal set of weights θ^* to minimize the loss function L_θ over the entire dataset. The objective function is represented as follows:

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{n=1}^{N} l\left(f_\theta\left(I_n\right), Y_n\right) \tag{1}$$

where l is the pixel-level error evaluation function (typically cross-entropy loss) aimed at maximizing the mIoU accuracy.

Constraints: The optimization process is constrained by the specific characteristics of technical drawings:

- Data class imbalance: The number of pixels belonging to the background class is significantly larger than those in the classes containing actual information.

- Structural consistency: The lines representing the structure must maintain spatial continuity without breaks or fragmentation. This ensures that the output data can be converted into graphs and guarantees accuracy during vectorization.

3.2. Motivation

Unlike natural images where boundaries between objects can be soft and flexible, technical drawings possess rigid geometry. Boundaries such as walls, doors, and windows are required to be straight, sharp lines with uniform thickness. Maintaining the "straight" and "clean" nature of these boundaries is a prerequisite for the segmentation results to be utilized in subsequent processing steps, such as vectorization or 3D reconstruction. Building upon [6], we observe that the Cross-Entropy loss function evaluates errors at the individual pixel level completely independently. It is solely concerned with whether the predicted probability of a pixel matches its label, completely ignoring the spatial relationships between neighboring pixels. Consequently, a prediction containing jagged "extra pixels" protruding around a wall edge, or a straight line broken into a sawtooth pattern, is not penalized more heavily by Cross-Entropy than a perfectly smooth line (provided that the total number of misclassified pixels is the same) [21]. This leads to blurred boundaries that contain a significant amount of local noise.

To thoroughly resolve this issue, this study proposes the integration of a loss function based on boundary detection. By applying spatial derivative filters to both the predicted mask and the ground truth mask, this loss function forces the model not only to learn how to correctly classify pixels but also to match the direction and magnitude of the contours. This mechanism acts as a robust regularizer, directly suppressing noisy extra pixels and forcing the neural network to interpolate straight, sharp boundaries that strictly adhere to the topological structure of the original drawing.

3.3. Overview architecture

The model used is inherited from [6]. The model's architecture is highlighted by three core technical pillars described below:

- Multi-task Network with Shared Features: The model utilizes a shared encoder to extract general features from the input drawing image. The data flow then splits into two independent decoder branches to address two parallel tasks: one branch predicts boundaries (walls, doors, windows), and another predicts spatial semantics (bedrooms, living rooms, hallways...).

- Spatial Contextual Module: The key innovation introduced by Zeng et al. [6] is a Boundary-guided Attention Mechanism. This mechanism extracts feature streams from the boundary prediction branch (top branch) and directly transmits them to the room type prediction branch (bottom branch). This forces the model to learn the actual spatial relationships: recognizing partitions and doors as physical boundaries that define a room.

- Cross-and-within-task Weighted Loss: Technical drawings exhibit extreme data imbalance (e.g., the stroke area of a window is miniscule compared to the floor area). This customized loss function is designed to balance the contribution level not only between object classes within the same task but also to balance the importance between the two tasks during backpropagation. Thanks to this combination, the model overcomes the limitations of legacy systems (which often had to rely on orthogonal interpolation rules - the Manhattan assumption), allowing for the successful recognition of drawings with complex architectures, curved walls, uneven line thicknesses, and non-rectangular rooms.

3.4. Edge-aware loss

We propose using an Edge-aware Loss for the boundary prediction branch, serving as a spatial regularizer during the training process. Instead of solely optimizing the label of each individual pixel in isolation, this loss function forces the model to attend to the gradient changes between neighboring pixels, thereby preserving the topological structure of the drawing.

Specifically, to extract boundary information, we apply a convolution operation with spatial derivative filters (the Sobel operator). Let $\widehat{Y}$ be the predicted probability mask extracted from the network, and $\nabla\widehat{y}$ is calculated by:

$$\nabla_x \widehat{Y} = S_x * \widehat{Y}; \quad \nabla_y \widehat{Y} = S_y * \widehat{Y} \tag{2}$$

where $*$ represents the convolution operation, and S_x and S_y are the kernels of the Sobel operator, E for both the predicted and ground truth images is determined using the

L_2 norm of the derivative:

$$E\left(\widehat{Y}\right) = \sqrt{\left(\nabla_x \widehat{Y}\right)^2 + \left(\nabla_y \widehat{Y}\right)^2} \tag{3}$$

$$E(Y) = \sqrt{(\nabla_x Y)^2 + \left(\nabla_y Y\right)^2} \tag{4}$$

The edge-aware loss function L_{edge} is defined as the absolute difference between the predicted edge magnitude and the ground truth edge magnitude:

$$L_{edge} = \frac{1}{M} \sum_{i=1}^{M} \left| E\left(\widehat{Y_i}\right) - E\left(Y_i\right) \right| \tag{5}$$

where M is the total number of pixels in a batch. Finally, the overall loss function (L_{total}) for optimizing the network is a linear combination of the base segmentation loss (L_{seg}) and the Edge-aware Loss:

$$L_{total} = L_{seg} + \lambda L_{edge} \tag{6}$$

where λ is a hyperparameter controlling the influence of the edge-aware branch.

For the floor plan parsing problem, optimizing L_{edge} brings immense value. This function directly "penalizes" predictions with blurred or jagged boundaries, as well as noise adjacent to walls (because they create spurious gradients not present in the original labels). Consequently, the neural network is forced to interpolate spatial segmentations with straight, neat boundaries while maintaining uniform thickness, perfectly accommodating the rigid geometric characteristics of technical drawings.

Furthermore, through a multi-task network architecture featuring a shared features mechanism and a spatial contextual module, this loss function also creates a positive synergistic effect. When the geometric features in the boundary prediction branch are refined to be sharper and more accurate by L_{edge}, this high-quality information flow is propagated directly down to the spatial function prediction branch. As a result, the model indirectly and significantly increases its accuracy in segmenting room types, establishing a clear physical constraint between the room space and its enclosing boundaries.

3.5. Improvement of the feature extraction network

In the architecture of the original model [6], VGG16 is utilized as the core feature extraction network. Despite its foundational role in many computer vision tasks, VGG16 exhibits critical limitations when applied to parsing architectural drawings of large scale and complex structure. By relying solely on stacked convolutional layers with small kernel sizes (3×3), VGG16 possesses a relatively narrow receptive field. This significantly diminishes its ability to capture the global spatial context—a crucial factor for the network to comprehend long-range dependencies between walls or the functions of large-area rooms. Furthermore, its outdated design burdens VGG16 with a massive number of parameters, skyrocketing GPU memory (VRAM) requirements without a commensurate improvement in accuracy.

To comprehensively resolve this issue, this study proposes replacing the VGG16 backbone with the modern convolutional network architecture, ConvNeXt [8]. ConvNeXt represents an evolution of traditional CNNs, modernized by integrating superior design philosophies from the Vision Transformer (ViT) family. Specifically, this architecture employs depthwise convolutions, an inverted bottleneck design, and, notably, the application of large kernel sizes (e.g., 7×7).

This modification yields two breakthrough advantages for the floor plan parsing problem:

- First, enhanced context awareness**:** Large kernel sizes expand the network's receptive field, allowing the model to easily aggregate information from distant regions of the drawing. This facilitates better linking of fragmented wall segments and more accurate delineation of open spaces.

- Second, optimized computational resources**:** ConvNeXt provides feature maps with deeper and richer semantic representations while consuming significantly fewer hardware resources compared to VGG16. This high-quality information flow serves as the perfect input "material" for the decoder branches and the Edge-aware Loss function (discussed in Section 3.4) to maximize their effectiveness in interpolating and sharply refining physical boundaries.

4. Experiments

4.1. Experimental setup

a. Dataset

In this paper, we utilize the CubiCasa5K dataset for model evaluation [5]. This dataset comprises 5,000 distinct technical drawings annotated with over 80 categories of floor plan objects. The data annotation is performed densely and flexibly, utilizing polygons to separate various objects.

b. Data preprocessing

To enhance the model's generalization capability and prevent overfitting, we apply a comprehensive data preprocessing and augmentation pipeline during the training process. Initially, the input images and all corresponding label masks (boundaries, room types, doors) are synchronously resized to a fixed resolution. Subsequently, spatial transformations (flipping, random 90-degree rotations) and color transformations (brightness, contrast adjustments) are applied with a 50% probability to diversify the training data. Notably, to simulate real-world drawings that are often blurred, noisy, or have strokes obscured by text labels and symbols, we additionally integrate Blur, Gaussian Noise, and random occlusion (Coarse Dropout) with a 30% probability. Finally, the input images are normalized according to the ImageNet dataset distribution before being fed into the network.

c. Model training

The comparative models are trained using the following hyperparameters: Adam optimizer, a learning rate of 0.0001, batch size 16 and 50 epochs. In our experiments, λ is fixed at 0.2. All reported results are averaged over five independent training runs.

4.2. Experimental results

In our result, the suffix "wkernel" denotes models trained with the proposed Edge-aware Loss, whereas the remaining models correspond to the standard configurations trained without this additional boundary-aware objective.

Performance is evaluated using Total Loss, Pixel Accuracy, and mean Intersection over Union (mIoU). These metrics are calculated separately for the two task branches: Room Metrics and Boundary Metrics.

Table 1. Overall segmentation performance of the compared models on the CubiCasa5K dataset

Model	Total Loss	Room Acc (%)	Room mIoU (%)	Boundary Acc (%)	Boundary mIoU (%)
VGG16	0.0981	79.76 ± 1.77	54.70 ± 3.13	93.58 ± 0.11	68.78 ± 0.21
ResNet34	0.0797	85.79 ± 0.74	66.59 ± 1.46	94.31 ± 0.58	71.21 ± 1.71
ResNet34 wkernel	0.0842	87.13 ± 0.12	68.82 ± 0.17	96.31 ± 0.12	77.70 ± 0.44
ConvNeXt-tiny	0.0821	88.68 ± 0.27	72.64 ± 0.51	94.89 ± 0.16	73.75 ± 0.62
ConvNeXt-tiny wkernel	0.0801	88.38 ± 0.58	72.27 ± 0.86	96.40 ± 0.37	78.57 ± 1.56

Note: Values are reported as mean ± standard deviation over five independent training runs; Total Loss is dimensionless.

The application of the Edge-aware Loss function (denoted by the wkernel variants) has yielded clear improvements, demonstrating a strong geometric optimization effect on the structure of the floor plans. The statistical results show that the model performs significantly better in detecting wall regions and boundaries when supported by this loss function:

- For the ConvNeXt-Tiny architecture: Integrating wkernel reduced the system's Total Loss from 0.0821 to 0.0801. In particular, Boundary mIoU increased substantially from 73.75% to 78.57%. As shown in Table 2, the model's Boundary Precision also improved significantly, from 77.47% to 82.99%, while maintaining a very high Boundary Recall of 93.08%.

- For the ResNet34 architecture: The impact of Edge-aware Loss on boundary optimization was even more pronounced. Boundary mIoU increased sharply from 71.21% to 77.70%. More notably, Boundary Precision jumped to 83.47%, compared with only 75.26% for the model without wkernel.

The substantial improvements in Boundary mIoU and Boundary Precision demonstrate that Edge-aware Loss plays an essential role in refining local boundary details. By imposing stronger penalties on inaccurate predictions around wall edges—particularly pixel-level noise that produces false gradients—this loss function effectively addresses a key weakness of standard segmentation models. As a result, the system can delineate wall regions more precisely and reduce the misclassification of adjacent empty spaces as walls.

A paired t-test reveals that the proposed ConvNeXt-tiny wkernel model significantly outperforms the VGG16 baseline in both Boundary mIoU ($p = 0.0001$) and Room mIoU ($p = 0.0001$), confirming the overall effectiveness of our framework. Crucially, isolating the impact of the proposed objective function, the addition of the Edge-aware Loss to the ConvNeXt-tiny backbone yields a highly significant improvement in boundary detection refinement (Boundary mIoU: $p = 0.0012$; Boundary Precision: $p = 0.0021$). This directly supports our hypothesis that the Sobel-based penalty effectively regularizes sharp geometric contours and suppresses boundary noise.

Table 2. Precision and Recall of the room and boundary branches of the compared models

Model	Room Precision (%)	Room Recall (%)	Boundary Precision (%)	Boundary Recall (%)
VGG16	70.06 ± 1.71	69.14 ± 3.68	73.27 ± 0.18	91.52 ± 0.42
ResNet34	77.61 ± 1.36	81.50 ± 1.00	75.26 ± 1.68	92.76 ± 0.18
ResNet34 wkernel	80.34 ± 0.99	81.88 ± 0.90	83.47 ± 0.33	90.94 ± 0.82
ConvNeXt-tiny	83.21 ± 0.64	84.39 ± 0.79	77.47 ± 0.82	93.88 ± 0.45
ConvNeXt-tiny wkernel	82.68 ± 0.27	84.68 ± 0.73	82.99 ± 2.03	93.08 ± 0.46

Note: Values are reported as mean ± standard deviation over five independent training runs.

Table 2 details the Precision and Recall metrics, further clarifying the impact of the proposed Edge-aware Loss. While the baseline ConvNeXt-tiny architecture achieves a high Boundary Recall (93.88%), its Boundary Precision is constrained to 77.47%. However, integrating the Edge-aware Loss (ConvNeXt-tiny wkernel) triggers a substantial leap in Boundary Precision to 82.99%, while maintaining a robust Boundary Recall of 93.08%. A similar optimization trend is observed in the ResNet34 wkernel model, where Boundary Precision surges to 83.47%, a marked improvement over the 75.26% achieved by its standard counterpart. This significant boost in precision metrics confirms that the Sobel-based penalty effectively suppresses false positive boundary predictions—such as jagged edges or artificially thickened lines—ensuring the model strictly delineates structural walls without hallucinating non-existent spatial boundaries.

Table 3 presents a detailed analysis of the computational costs of different network architectures, including graphic memory (VRAM) consumption and average inference time per image. From the experimental data, it is evident that the proposed model (ConvNeXt-tiny combined with the Edge-aware Loss) delivers outstanding hardware optimization, accompanied by a completely reasonable trade-off in terms of time.

Table 3. Comparison of average inference time and memory usage for each architecture

Model	VRAM peak (MB)	Time (ms/image)
VGG16	2009.95	58.54 ± 4.30
ResNet34	1590.45	40.13 ± 0.48
ResNet34 wkernel	1590.45	40.26 ± 2.51
ConvNeXt-tiny	1133.36	68.93 ± 2.67
ConvNeXt-tiny wkernel	1133.36	72.44 ± 2.78

Note: Inference time is reported as mean ± standard deviation over five independent runs.

Specifically, the greatest advantage of ConvNeXt-tiny lies in its memory optimization capability. This architecture records the lowest VRAM consumption across the board at only 1133.36 MB, equating to a nearly 50% reduction (approximately 43.6%) compared to the massive 2009.95 MB required by the original VGG16 architecture. This compactness carries crucial practical significance, allowing users to train the model with larger batch sizes for better system convergence, or facilitating easy deployment on server systems and edge devices with limited graphic resources.

Regarding processing speed, the inference time of ConvNeXt-tiny wkernel (72.44 ms/image) shows an increase compared to VGG16 (58.54 ms/image) and ResNet34 (ranging from 40.13 ms to 40.26 ms). However, within the specific context of the architectural drawing digitization problem (2D-to-BIM)—which involves offline static image processing tasks and does not demand strict real-time responses like video processing—a latency of a few dozen milliseconds is entirely negligible.

Furthermore, experiments also indicate that integrating the Edge-aware Loss function (models with the *wkernel* suffix) has an insignificant impact on computational costs. It maintains the exact same VRAM consumption (since the Sobel filter computation only occurs during the training process or is directly aggregated into the weights) and merely adds about 3.51 ms to the inference time for ConvNeXt-tiny.

Synthesizing these findings with the accuracy metrics (mIoU, Precision, Recall) in Table 1 and Table 2, it can be concluded that selecting ConvNeXt-tiny combined with the Edge-aware Loss is a strategic move that yields comprehensive effectiveness: trading off a minuscule amount of inference time in exchange for a significant reduction in hardware burden and an outstanding leap in the quality of spatial segmentation.

The illustration demonstrates performance on two distinct architectural drawing samples:

- First drawing sample: (a) original image, (b) ground truth, (c) VGG16 model, (d) ResNet34 model, (e) ResNet34 + Edge-aware Loss model, (f) ConvNeXt-tiny model, (g) ConvNeXt-tiny + Edge-aware Loss model

- Second drawing sample: (h) original image, (i) ground truth, (j) VGG16 model, (k) ResNet34 model, (l) ResNet34 + Edge-aware Loss model, (m) ConvNeXt-tiny model, (n) ConvNeXt-tiny + Edge-aware Loss model

The prediction results from the VGG16 network clearly expose its weakness in capturing geometric structures. The predicted boundary lines are blurred, fragmented, and contain substantial local sawtooth-like noise around the wall edges. This perfectly aligns with the hypothesis regarding VGG16's narrow receptive field, which renders it incapable of linking distant spatial contexts to maintain stroke continuity.

Fig. 1 shows that the main improvements brought by the Edge-aware Loss are not limited to general visual quality but are specifically related to boundary error reduction. The standard models tend to produce three types of errors: fragmented wall segments, jagged contours around wall edges,

and uneven boundary thickness. These errors are especially visible near corners, door regions, and thin partitions. When the Edge-aware Loss is added, these artifacts are reduced because the Sobel-based objective penalizes mismatches between the predicted and ground-truth edge magnitude maps. Noisy protruding pixels create spurious gradients, while blurred or broken boundaries produce weak or discontinuous edge responses; both cases increase the edge loss during training.

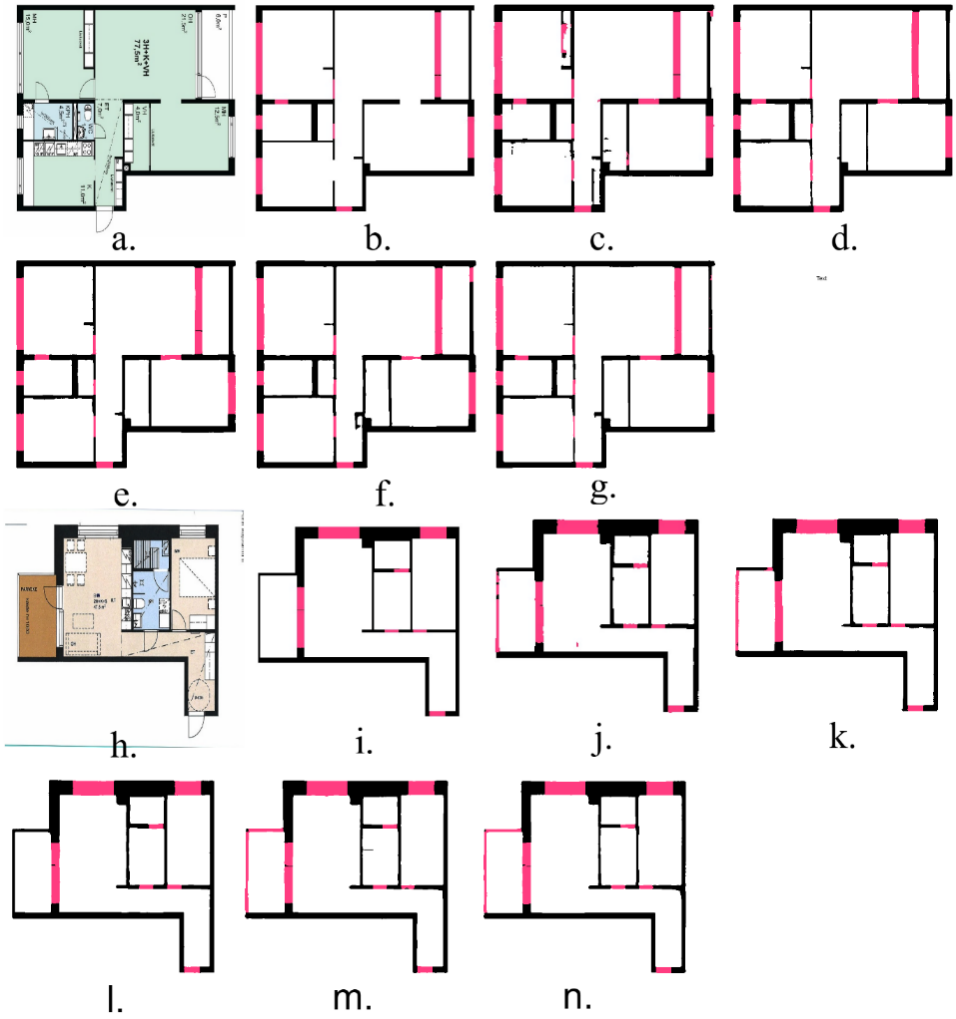


Figure 1. Comparison of boundary prediction results across different models

Upon replacing VGG16 with ResNet34 and ConvNeXt-tiny (the versions without the edge loss function), the overall segmentation quality is improved. The network successfully predicts complete, long wall segments with fewer fragmentations. However, because it still solely relies on the Cross-Entropy loss function, the phenomenon of jagged "extra pixels" and boundaries with uneven thickness persists at certain wall corners or door boundaries.

The decisive difference becomes clearly apparent when comparing the standard models with the versions integrating the Edge-aware Loss (indicated by the *wkernel* suffix). By observing the outputs of ResNet34 and particularly ConvNeXt-tiny when combined with the Edge-aware Loss, it is evident that the loss function based on the Sobel operator acts as a highly effective spatial regularizer. The phenomena of granular noise and jagged wall edges are almost completely eliminated. The model is

forced to interpolate sharp, straight boundaries that perfectly adhere to the topological structure of the ground truth label.

5. Conclusions

This research presents an improved method for the automatic segmentation and recognition of architectural floor plans from complex technical drawings. By inheriting and developing a multi-task network architecture, this study focused on addressing two core challenges: the geometric rigidity of boundary lines and the efficiency of the feature extraction network.

The main contributions and conclusions from this research include:

- Effectiveness of the Edge-aware Loss: The integration of the Sobel operator into the objective function serves as a powerful spatial regularizer. Empirical results show that this loss function helps eliminate local noise, rectify jagged boundaries, and preserve the strict geometric properties of technical drawings.

- The study proves that replacing traditional CNN networks with the ConvNeXt-tiny architecture brings outstanding feature extraction capabilities. The combination of ConvNeXt-tiny and Edge-aware Loss enabled the model to achieve superior performance on the benchmark CubiCasa5K dataset.

Although the proposed method achieves improved segmentation accuracy and better boundary preservation, several limitations remain. The experiments are mainly conducted on the CubiCasa5K dataset, so the robustness of the method on more diverse real-world drawings, such as scanned, noisy, or low-resolution floor plans, should be further investigated. In addition, the current model produces pixel-level segmentation maps, which still require post-processing and vectorization before being directly used in CAD/BIM systems.

Future work will focus on extending the evaluation to larger and more diverse datasets, integrating the proposed segmentation model with graph-based vectorization algorithms, and developing a more complete 2D-to-BIM reconstruction pipeline. Moreover, adaptive weighting of the Edge-aware Loss and the combination of boundary-aware segmentation with graph or transformer-based models may further improve both geometric consistency and semantic recognition performance.

Declaration of generative AI and AI-assisted technologies

The author(s) declare that no generative AI or AI-assisted technologies were used in the preparation of this manuscript.

References

[1] Sacks, R., Eastman, C., Lee, G., Teicholz, P. (2018). *BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers*. 3rd edition, John Wiley & Sons, Inc., Hoboken, NJ.

[2] Nguyen, T. B., Tran, A. B., Phan, H. T., Do, Q. H., Nguyen, Q. T. (2021). Exploitation of digital data from building information models in virtual reality technology. In *Proceedings of the 6th International Conference on Geotechnics, Civil Engineering and Structures (CIGOS)*, Springer Nature Singapore, 1833–1840.

[3] Dosch, P., Tombre, K., Ah-Soon, C., Masini, G. (2000). A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition (IJDAR)*, 3(2):102–116.

[4] O'Shea, K., Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

[5] Kalervo, A., Ylioinas, J., Häikiö, M., Karhu, A., Kannala, J. (2019). CubiCasa5K: A dataset and an improved multi-task model for floorplan image analysis. In *Proceedings of the 21st Scandinavian Conference on Image Analysis (SCIA)*, 28–40.

[6] Zeng, Z., Li, X., Yu, Y. K., Fu, C. W. (2019). Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9095–9103.

[7] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
[8] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., Xie, S. (2022). A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11976–11986.
[9] Macé, S., Locteau, H., Valveny, E., Tabbone, S. (2010). A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (DAS)*, 167–174.
[10] Becker, S., Peter, M., Fritsch, D. (2015). Grammar-supported 3D indoor reconstruction from point clouds for "as-built" BIM. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W4:17–24.
[11] Xu, Z., Jha, N., Mehadi, S., Mandal, M. (2025). Automatic floor plan analysis: Datasets, methods, and applications (2000–2025). *Automation in Construction*, 178:106378.
[12] Liu, C., Wu, J., Kohli, P., Furukawa, Y. (2017). Raster-to-vector: Revisiting floorplan transformation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2214–2222.
[13] Lv, X., Zhao, S., Yu, X., Zhao, B. (2021). Residential floor plan recognition and reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16712–16721.
[14] Chen, M., Pan, C. (2023). Parsing line segments of floor plan images using graph neural networks. *arXiv preprint arXiv:2303.03851*.
[15] Hu, S., Wu, W., Su, R., Hou, W., Zheng, L., Xu, B. (2024). Raster-to-graph: Floorplan recognition via autoregressive graph prediction with an attention transformer. *Computer Graphics Forum*, 43(2):e15007.
[16] Xu, Z., Yang, C., Alheejawi, S., Jha, N., Mehadi, S., Mandal, M. (2024). Automatic floor plan analysis using a boundary attention-based deep network. *International Journal on Document Analysis and Recognition (IJDAR)*, 28(1):19–30.
[17] Kratochvila, L., de Jong, G., Arkesteijn, M., Bilik, S., Zemcik, T., Horak, K., Rellermeyer, J. S. (2024). Multi-unit floor plan recognition and reconstruction using improved semantic segmentation of raster-wise floor plans. *arXiv preprint arXiv:2408.01526*.
[18] De Nardin, A., Zottin, S., Toma, A., Piciarelli, C., Foresti, G. L. (2025). A multibranch and multiattention framework for floor plan segmentation. *Computer-Aided Civil and Infrastructure Engineering*, 40(23): 3674–3687.
[19] Kirillov, A., Wu, Y., He, K., Girshick, R. (2020). PointRend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9799–9808.
[20] Xie, H., Ma, X., Mei, Q., 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.
[21] Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ben Ayed, I. (2021). Boundary loss for highly unbalanced segmentation. *Medical Image Analysis*, 67:101851.
[22] Hu, R., Huang, Z., Tang, Y., Van Kaick, O., Zhang, H., Huang, H. (2020). Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (TOG)*, 39(4):118:1–118:14.
[23] Nauata, N., Hosseini, S., Chang, K.-H., Chu, H., Cheng, C.-Y., Furukawa, Y. (2021). House-GAN++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13627–13636.
[24] Shabani, M. A., Hosseini, S., Furukawa, Y. (2023). HouseDiffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5466–5475.
[25] Kuhn, E. (2023). Adapting HouseDiffusion for conditional floor plan generation on modified Swiss dwellings dataset. *arXiv preprint arXiv:2312.03938*.
[26] Wang, S., Pajarola, R. (2025). Eliminating rasterization: Direct vector floor plan generation with DiffPlanner. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):7906–7922.
[27] Lu, Z., Wang, T., Guo, J., Meng, W., Xiao, J., Zhang, W., Zhang, X. (2021). Data-driven floor plan understanding in rural residential buildings via deep recognition. *Information Sciences*, 567:58–74.