Wall and Room Detection in Architectural Blueprints Using Mask R-CNN

Vinat Goyal

September 2024

1 Introduction

Room and wall detection from architectural blueprints is an important task in many fields, including construction, real estate, and smart building design. Accurate detection of walls and rooms helps streamline construction planning, automate building assessments, and enable efficient use of smart systems. However, detecting walls and rooms from floor plans presents a number of challenges, such as varying architectural styles, complex layouts, and overlapping structural elements.

The advent of machine learning and computer vision techniques have significantly improved the automation of these tasks. Among these, convolutional neural networks (CNNs), particularly Mask R-CNN, have proven to be highly effective for instance segmentation tasks. Mask R-CNN can accurately detect and segment walls and rooms in floor plans, making it a powerful tool for analyzing architectural blueprints.

In this project, we conducted a literature survey to understand existing methods for room and wall detection, explored openly available datasets, and ultimately trained a Mask R-CNN[2] model on the CubiCasa5k dataset. This report provides an overview of the problem, methods, and results obtained during the project.

2 Literature Survey

Room and wall detection from floor plans has been approached using various methods over the years. Traditional methods often rely on edge detection techniques such as Canny Edge Detection and Hough Transform, which focus on identifying straight lines to detect walls. However, these methods struggle with complex and noisy floor plans, where walls may not always be represented as straight lines[5].

To address these challenges, deep learning models, particularly convolutional neural networks (CNNs), have become increasingly popular. Among these, Mask R-CNN is widely used for instance segmentation tasks, making it highly suitable for wall and room detection. Mask R-CNN builds on Faster R-CNN by adding a branch for predicting segmentation masks, allowing it to both detect and segment objects in an image. In the context of architectural blueprints, this enables the model to detect walls and rooms with high accuracy.

Other advanced methods have explored the use of super-resolution to enhance low-quality floor plan images before performing detection. For instance, techniques like EDSR and FSRCNN have been used to upscale images, resulting in improved detection of walls and rooms in complex layouts[4].

Additionally, researchers have investigated the use of graph neural networks (GNNs) to detect walls and room boundaries. GNNs represent architectural elements, such as walls and junctions, as nodes and edges in a graph, making it easier to detect complex structures in floor plans[1].

It is clear that floor plan research offers a vast area of exploration with a lot of ongoing research. However, there is still a lack of a large-scale open-source dataset. From what I have found, the CubiCasa5k[3] dataset is currently the largest publicly available dataset for floor plan research. It offers detailed annotations for walls, rooms, and other architectural elements. Because of its scale and quality, I chose the CubiCasa5k dataset for this project to train and evaluate the Mask R-CNN model for room and wall detection.

3 Proposed Solution

The problem statement requires that both walls and rooms be detected and annotated separately in architectural blueprints. Given the need for distinct, non-overlapping annotations of individual elements, it is convenient to approach the problem as an instance segmentation task.

This section details the choice of model, the dataset used, and the environment employed for training.

3.1 Model Selection: Mask R-CNN

Mask R-CNN[2] was chosen for this task as it is the go-to model for instance segmentation. It not only detects objects by generating bounding boxes but also provides pixel-level segmentation masks for each detected instance, making it ideal for distinguishing both walls and rooms in architectural blueprints.

Additionally, since architectural floor plans suffer from a scarcity of large annotated datasets, leveraging pre-trained models such as ResNet-50 (pre-trained on ImageNet) is highly beneficial for transfer learning. By using these pre-trained weights, the model can start with a strong understanding of basic features, speeding up convergence and improving performance, even with limited training data.

3.2 Dataset

As previously mentioned, the CubiCasa5k dataset was used for this task. The dataset provides annotations for walls, rooms, and icons. However, in this study, we focus solely on walls and rooms, disregarding the icon annotations. The dataset includes 12 room-related classes: 'Background,' 'Outdoor,' 'Wall,' 'Kitchen,' 'Living Room,' 'Bedroom,' 'Bath,' 'Entry,' 'Railing,' 'Storage,' 'Garage,' and 'Undefined.' We have excluded indices 0 and 1, which correspond to the background and outdoor classes, respectively.

The dataset contains 4,200 training images organized into three folders: 'colorful,' 'high-quality,' and 'high_quality_architectural.' Due to hardware constraints, we opted to train the model using only the high-quality subset, which consists of 992 images, with 600 used for training, 200 for validation, and 192 for testing

3.3 Training Environment

The model was trained using the **Stochastic Gradient Descent (SGD)** optimization algorithm. The specific hyperparameters used in the training process are outlined below:

- **Learning Rate**: 0.005
- **Momentum**: 0.9
- **Weight Decay**: A regularization technique to prevent overfitting was applied, with a weight decay value of 0.0005.
- **Scheduler**: The learning rate was scheduled to decay by a factor of 0.1 after every 3 epochs using a step-based learning rate scheduler. This helps ensure that the learning rate decreases over time to fine-tune the model during the latter stages of training.

The training was conducted for 10 epochs.

4 Results and Discussion

4.1 Quantitative Results

The results were obtained by applying non-maximum suppression with an IoU threshold of 0.25 and confidence score of 0.25, followed by calculating True Positives (TPs), False Positives (FPs), and False Negatives (FNs) using an IoU threshold of 0.5. A lower IoU threshold was chosen for non-maximum suppression based on the rationale that the bounding boxes for rooms and walls might overlap at the edges. The results are presented in table 1,

Class	TPs	FPs	FNs	Precision	Recall
Wall	3057	1044	1512	0.7454	0.6691
Rooms	1106	663	815	0.6252	0.5

Table 1: Detection Results for Wall and Rooms Classes

The results are satisfactory, especially considering that the full dataset was not utilized and this was the first experiment. With a precision of 0.7454 and recall of 0.6691 for the wall class, the performance is above average. The rooms class consists of 9 sub-classes, and since only 600 images were used for training, it is possible that not all sub-classes were adequately represented. Despite this limitation, the model has produced promising results.

There remains scope for improvement, and with the inclusion of more data and further experiments, a precision and recall of over 0.85 for both the walls and rooms classes is achievable.

4.2 Qualitative Results

In addition to the quantitative metrics, we provide visualizations that showcase the model's predictions on a sample image from the dataset. These visual examples help illustrate how well the model identifies walls and rooms, highlighting both its strengths and areas where improvements are still possible. Figure 1 shows the sample image. Figure 2 shows the visualised model predictions on the sample image.

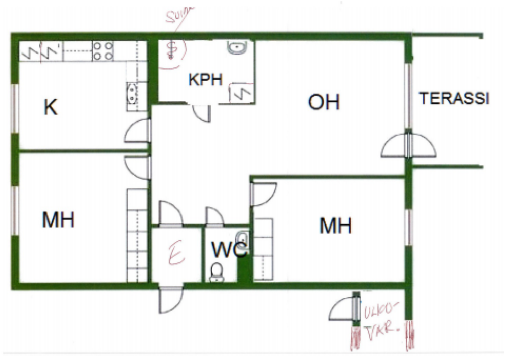
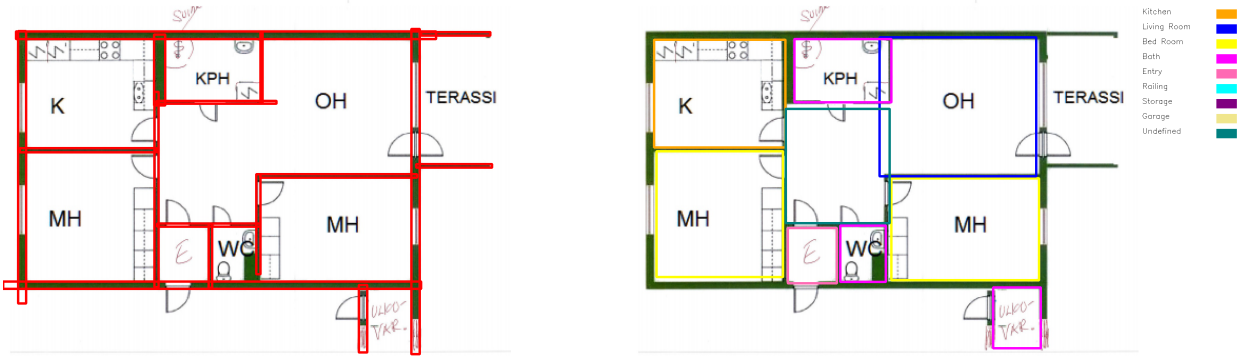


Figure 1: Sample Image



(a) Input image annotated with Predicted Walls

(b) Input image annotated with predicted rooms

Figure 2: Visualisation results of the model

The visualizations clearly demonstrate that the model performs well in predicting both the walls and different rooms. However, there are areas for improvement, such as overlapping room bounding boxes and occasional mispredictions in the wall class..

4.3 Training Discussion

Figure 3 shows the training and validation loss over the course of 10 epochs. The training loss drops rapidly in the first few epochs. The validation loss initially drops but then plateaus and stays relatively flat after a few epochs (around 3 to 5 epochs). There is a noticeable gap between the training loss (which keeps improving) and the validation loss. This difference suggests that the model may be overfitting, meaning it is learning the training data well but struggling to generalize to the unseen validation data. Possible solutions include using more data, batch normalization, data augmentation, and hyper-parameter tuning.

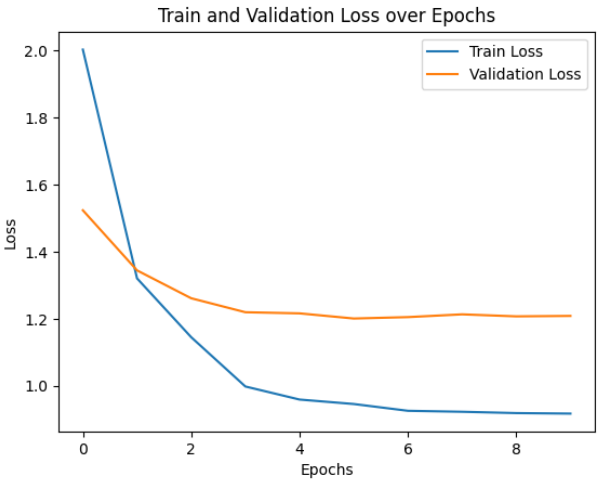


Figure 3: Train and Validation Loss over Epochs

5 Future Work

This project served as an initial exploration into the field of floor plan analysis. We reviewed relevant research papers, explored openly available datasets, and trained a Mask R-CNN model on a subset of the CubiCasa5k dataset, running for 10 epochs. There is significant room for improvement. Some key areas for future enhancement are discussed below:

5.1 Exploring Alternative Architectures

Exploring newer architectures such as YOLOv11 could offer significant advantages over the current Mask R-CNN model. YOLOv11 and other modern models are designed for faster inference and often provide competitive or superior performance with fewer computational resources. By leveraging these recent advancements, the model could achieve quicker detection of walls and rooms while maintaining or improving accuracy. This would be particularly beneficial when scaling to larger datasets or deploying in real-time applications.

5.2 Separate Detectors for Walls and Rooms

The trained model has underperformed on the room class. In this project, we used a single model for detecting both walls and rooms. Given the clear relationship between walls and rooms—where wall detections can guide room detections—a more effective strategy would be to use two separate detectors. The first detector would handle wall detection, and its outputs would serve as inputs to a second detector which would perform room detection. This approach could leverage the spatial relationships between walls and rooms, leading to improved room detection accuracy.

5.3 Utilizing the Full Dataset

In the current project, we only used 400 images out of the 4,200 available in the CubiCasa5k training set. To enhance the model's ability to generalize and reduce overfitting, future work should focus

on leveraging the entire dataset for training. Using all available data will allow the model to learn more diverse architectural patterns.

5.4 Multi-Task Learning

Currently, we have trained a single-task (Mask R-CNN) model solely for room and wall detection. A promising future direction would be to explore multi-task learning approaches as used by [5]. This would allow the model to learn from shared representations and further improve its accuracy and robustness by integrating multiple related tasks.

5.5 Hyperparameter Tuning

In this project, we conducted only one experiment without thorough hyperparameter optimization. To achieve better results, it is important to fine-tune key hyperparameters such as the learning rate, batch size, and weight decay. Packages like ML Flow could be used to keep a track of different experiments.

5.6 Training on Region-Specific Floor Plans

It would be beneficial to collect and train the model on floor plans from the area where it will be deployed since architectural styles vary by region. This ensures the model becomes familiar with local styles, reduces bias from unrelated architectural designs, and improves its accuracy and adaptability for the intended environment.

References

[1] Mingxiang Chen and Cihui Pan. Parsing line segments of floor plan images using graph neural networks, 2023.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[3] Ahti Kalervo, Juha Ylioinas, Markus Häikiö, Antti Karhu, and Juho Kannala. Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis, 2019.

[4] Dev Khare, N S Kamal, Barathi Ganesh HB, V Sowmya, and V V Sajith Variyar. Enhanced object detection in floor-plan through super resolution, 2021.

[5] Zhiliang Zeng, Xianzhi Li, Ying Kin Yu, and Chi-Wing Fu. Deep floor plan recognition using a multi-task network with room-boundary-guided attention, 2019.