Deep Learning-Based Segmentation and 3D Reconstruction for Heterogeneous Mixed-Use Building Layouts

Andrea Carrara, Lip Kun Tee, Stavros Nousias, André Borrmann
Technical University of Munich, Germany
andrea.carrara@tum.de

1. Introduction

The ongoing development of Building Information Modeling (BIM) techniques has attracted increasing attention in architecture and engineering because of its many benefits for design and project management, streamlining paper-based processes to cut down on manual data entry, reducing both effort and the risk of errors (Borrmann et al. 2018). Even with these developments, many technical drawings of existing buildings remain preserved in paper archives, and digitizing those drawings is lengthy and costly. The conversion of 2D architectural building layouts into 3D representations has been a focal point in computer vision and pattern recognition (Vidanapathirana et al., 2021), (Liu, Wu, and Furukawa 2018), (Park and Kim, 2021).

However, existing research (LV et al. 2021), (WU et al. 2020), (LIU et al. 2020) has primarily concentrated on residential-scale floor plans, neglecting the intricacies of mixed-use building layouts found in transit hubs, educational buildings, shopping malls, museums, and hospitals.

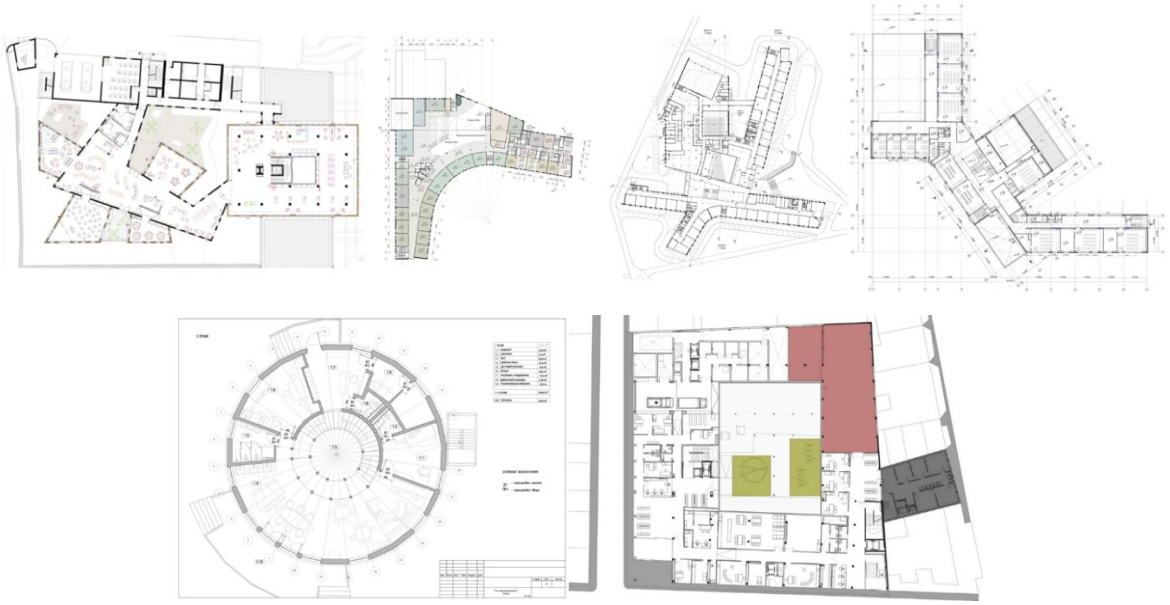

Figure 1: Example of mixed-use building layout in the analysis

Unlike residential buildings, mixed-use buildings combine multiple functionalities into one structure, catering to large crowds of occupants. This unique characteristic presents a less-explored domain. Unlike residential buildings, mixed-use buildings serve diverse functions, leading to intricate floor plan layouts. Residential structures are typically designed for a single purpose, resulting in straightforward floor plans optimized for living areas, kitchens, and bathrooms. However, mixed-use buildings designed for multiple functions exhibit greater complexity. The increased floor area, incorporation of vertical transportation spaces like elevators and staircases, and non-standard wall forms and door/window sizes make accurate

wall, door, and window detection in 2D plans challenging and hinder the precision of vectorization and 3D reconstruction.

This paper proposes a novel data-driven approach for vectorization and 3D reconstruction of mixed-use building layouts. The information from the drawings is extracted from two neural networks: the first detects the pixels depicting walls following a semantic segmentation approach, and the second detects the openings in the image using object detection. The information is post-processed and combined to obtain a 3D model.

2. Related Work

Various methods exist for converting 2D floor plan images into 3D models, highlighting diverse approaches. Graph Neural Networks (GNNs) and graph-based representations have effectively handled intricate floor plans with unconventional objects (Simonsen et al. 2021). However, they require the layout to have vector representations provided through SVG, PDF, CAD formats, etc. The 3DPlanNet method utilizes ensemble methods, combining data-based models and rule-based heuristics, significantly reducing the required dataset size while achieving high accuracy in generating wall objects from 2D residential drawings that follow the Manhattan-convention of perpendicularity of the walls (Park and Kim 2021). Other research defines the premise of the network on the Manhattan assumption to achieve high accuracy on wall vectorization and reconstruction from layouts (Liu et al. 2017), (Kim, Park, and Yu 2018), (Kalervo et al. 2019). In raster-to-vector, they introduce a methodology that transforms a floorplan image using two intermediary representation layers. A neural architecture initially transforms a floorplan image into a junction layer, where information is encoded as a collection of perpendicular junctions (i.e., points with fixed connection typologies). Integer programming combines junctions into a collection of essential elements (lines or boxes) while guaranteeing a topologically and geometrically coherent result for designs that follow the Manhattan assumption. Kalervo et al. employ in CubiCasa5k the neural network architecture from Raster-to-Vector, enhancing the results by applying the multi-task uncertainty loss function. The gap in the existing research is that the considered drawings are predominantly residential floor plans with straightforward perpendicular layouts. Non-rectilinear and complex commercial building designs have not been adequately addressed.

3. Methodology

The overall workflow of the proposed method, as shown in Figure 2, involves three steps: the segmentation step, the object detection step, and the mesh generation step. It generates a 3D model as an output corresponding to the 2D-floor plan image as the input in an end-to-end manner. The 3D model comprises geometric data that represents the walls and openings (doors and windows) of the mixed-use building.

Mixed-use floor plans often feature curving walls and a more comprehensive range of layouts than traditional residential floor plans. This makes it more challenging to locate the wall using a heuristic method. Consequently, the decision was made to employ the deep learning methodology to extract the walls from the image. The efficacy of the semantic segmentation approach lies in its ability to accurately identify walls while effectively excluding other elements within the image, including windows, doors, furniture, text, and miscellaneous items.

To consider the different floor design sizes, we partition them into many patches of standard size before inference. The segmentation model is initialized with an input size of 192 x 192

pixels. Subsequently, the floor plan image is scaled to the nearest integer multiple of 192 in width and height dimensions.

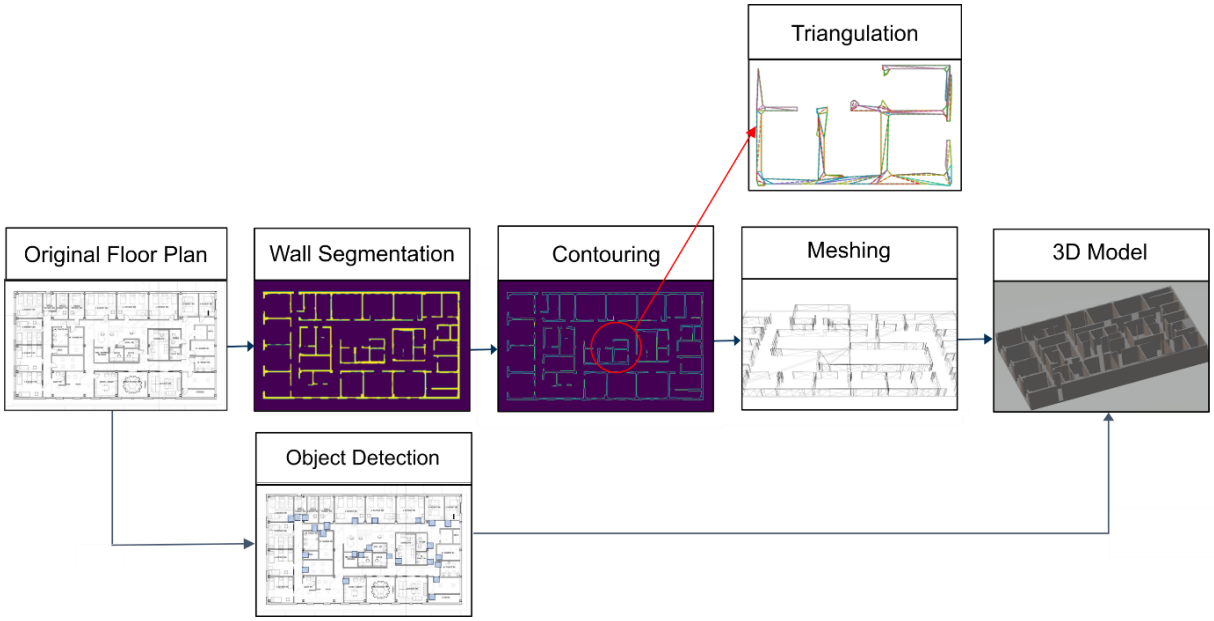

Figure 2: Proposed workflow from floor plan to 3D model

Figure 3 demonstrates the precise division of the image into several smaller patches of 192 × 192 pixels, which precisely matches the input geometry of the AI model. We opt to enlarge the image to prevent the potential of reducing any dimension to 0 pixels. Subsequently, the segmentation model conducts inference on these small picture patches to eliminate any visual elements other than the walls, generating a wall mask image for each patch.

The process involves merging the wall mask images into a unified image, following the same sequence as the original input image. The wall mask picture produced will be utilized to generate the results of the combined wall mask, resulting in the reversion of their positions to their original positions before inference. Finally, the wall mask image obtained is subsequently resized to match the dimensions of the initial floor plan image.

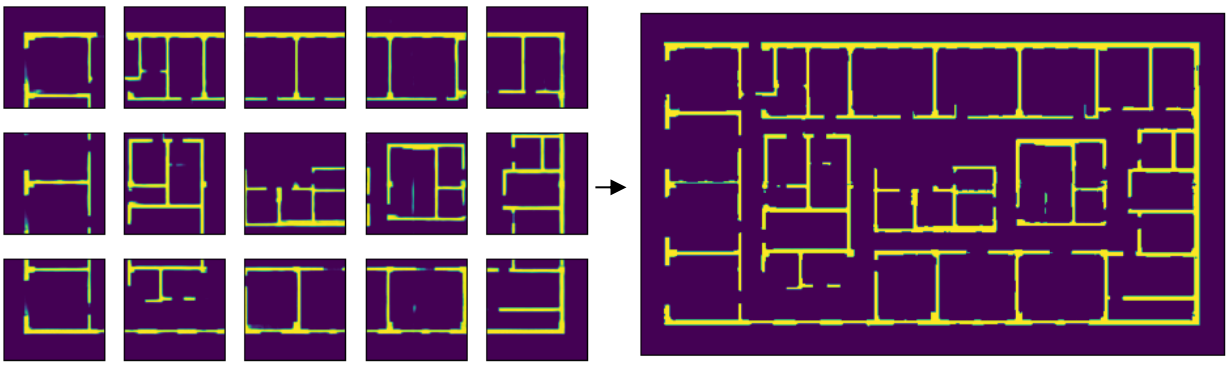

Figure 3: Prediction on fixed-dimension patches and recombination to a single image

The segmentation model we use is a U-Net model. The model's encoder has been modified to a feature extractor backbone to improve accuracy. In our case, we use the EfficientNetB3 as the backbone of the U-Net model.

The data analyzed in our research covers various structures, including large buildings like malls, hospitals, and airports. These structures are linked to certain symbols that communicate

unique meanings pertinent to their respective functions. Capturing every element in these floor layouts is difficult due to their intricate and diverse nature and inefficient using semantic segmentation. Consequently, we use object detection to identify other elements apart from walls, identifying and analyzing openings comprising windows and doors.

Figure 4: The three opening elements to train the object detection model (on the left) and an example of opening detection (on the right)

We process the semantic segmentation results depicting the walls and the openings detected through the object detection module to create the final 3D mesh. The wall mask is necessary to extract information like its location and thickness. The wall mask comprises an array generated by the neural network that contains continuous values within the range of 0.0 to 1.0, binarized with 0.5 thresholds to the nearest integer. The binary mask is subsequently utilized as the foundation for generating the wall mesh.

We apply contouring on the binary wall mask to determine the position of the walls. This process allows us to retrieve the outlines of the walls. The contours refer to the delineation of the walls, accompanied by positional data represented by vertices. It is necessary to connect these vertices to derive the geometry of the wall. The convexity of the walls is not a requirement. Hence, a simplistic Delaunay's triangulation (Ito 2015) approach would be erroneous. For our specific situation, we employ a triangulation method known as ear-clipping (Mei, Tipper, Xu 2013). Triangulation connects the vertices to create triangles, also known as faces, that constitute the 2D geometric structure of the wall. The methodology enables the acquisition of wall geometry while ensuring that the area is neither underestimated nor overestimated.

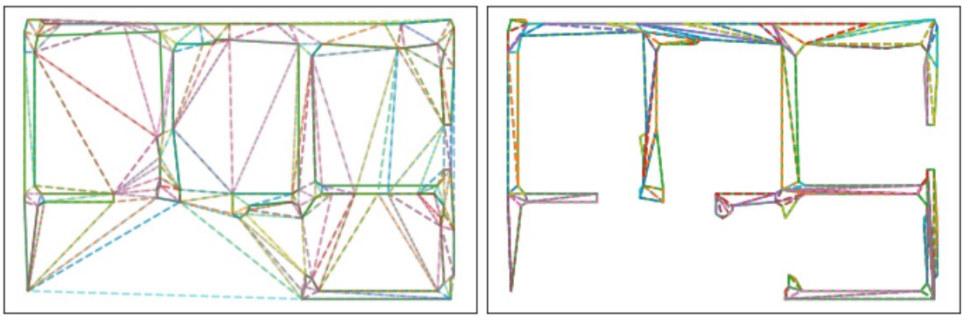

Figure 5: Comparison of different triangulation methods

Figure 4 demonstrates that the ear-clipping technique can accurately capture the contour of the wall, even when the geometry is concave. Subsequently, the wall mesh is extruded along an additional dimension, specifically the z-axis, with a predefined height of three meters. The extrusion method involves replicating a 2D wall mesh at a specific height. Subsequently, the two meshes are interconnected by numerous meshes parallel to the z-axis.

The door meshes can be generated using the bounding box information obtained from the object detection model. A predetermined default door mesh is available, with two basic rectangles

parallel to the z-axis; we define two meters in height by default. The mesh is created, positioned, and aligned based on the data obtained from the identified bounding box. Given the existing wall geometry and vertices obtained in the preceding phase, the computation of the door's position and angle can be readily performed by employing the relationships depicted in Figure 6. The orientation (angle) of the door can be effectively computed by establishing the coordinates of the red and green dots that constitute the wall gap's midpoint and the detected opening's center point. Each computed red dot is assigned a door mesh. Subsequently, the mesh undergoes rotation by the calculated angle along the z-axis concerning the x coordinates of the wall gap's midpoint.

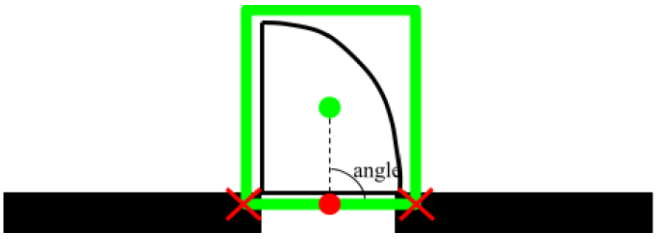

Figure 6: The red crosses indicate the collision points between the bounding box (green box) and the adjacent walls (solid black lines); the red dot is computed as the midpoint of these two crosses. The green dot is the midpoint of the whole bounding box.

We create a geometrical 3D model using the OBJ format. When dealing with geometrical representations, examining two main attributes, the vertex, and the face, is essential. A line break serves the purpose of separating each attribute. In the context of a 3D model, each attribute typically has three values. A vertex is represented as "v" and has three coordinates. In contrast, a face is represented as "f" with three indices representing the vertex (in the counterclockwise direction), forming a triangle. A facial structure may also possess four vertex indices constituting a quadrilateral. Nevertheless, our mesh is comprised solely of triangles obtained through triangulation. The output file is subsequently augmented with the vertex and face data of the wall and door meshes.

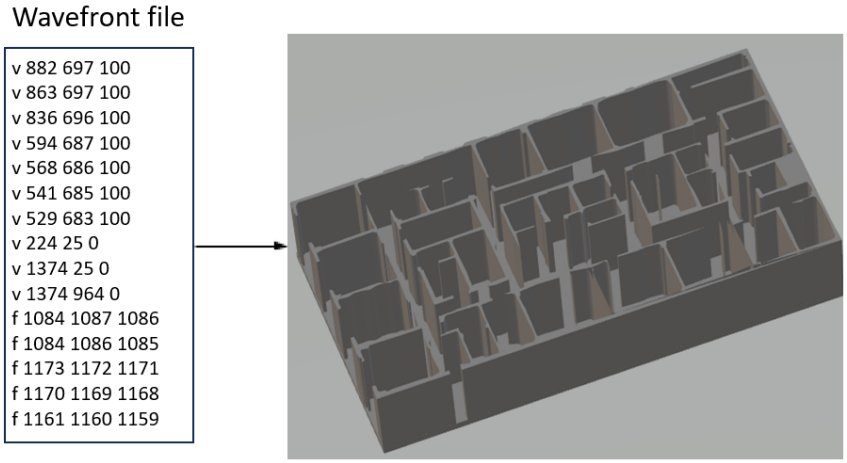

Figure 7: Render of the 3D model from the wavefront file

4. Experiment and Results

Our dataset comprises 100 mixed-use layouts of large-scale buildings such as hospitals, offices, airports, and malls retrieved from the internet. The style of the drawings is heterogeneous for

most of the retrieved layouts. Each layout has been labeled for two tasks: the wall mask for semantic segmentation and bounding boxes for opening detection. The distinction is dictated by the domain knowledge of the detected elements: openings maintain a similar representation following a determined ratio, while walls can be highly irregular and different. We use data-driven approaches to carry out those tasks.

To create the dataset for the segmentation task, we generate a wall mask image for every floor plan image used for the training dataset. The process involves removing any irrelevant or additional structural information on the floor plan, leaving only the walls.

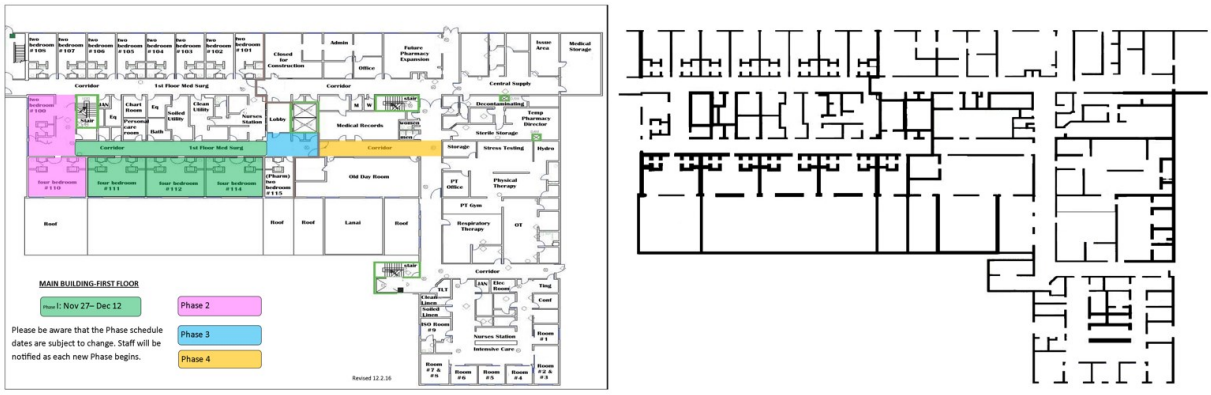

Figure 8: Example of wall mask for the semantic segmentation task

Furthermore, aside from obtaining a training dataset for the item detection task, we have employed the Synthetic Floor Plan Images (SFPI) dataset (Mishra et al 2021), which is readily available to the public. Nevertheless, the model trained using this dataset requires improved recall accuracy when tested on our specific floor plan images. This difficulty arises partly due to the restricted variety of shapes or patterns in the items in the synthetic data, which causes the model to overfit the synthetically generated objects.
The architectural elements in our data vary significantly from one to another. Therefore, it is necessary to create a new dataset for object detection. The annotation was done on three object classes (windows, single- and double-hinged doors). There are 7764 annotation instances on the floor plan images. Due to the low training data, we apply some image augmentation, such as random flipping, noise, and blur, to increase the number of instances.

A loss function analysis determines the best loss function to train the semantic segmentation model. Different loss functions are adequate for the segmentation tasks; however, each has different advantages, affecting the model's accuracy and training efficiency (Jadon 2020). We experimentally determine the best fitting for mixed-used buildings. The metrics used for the analysis consist of Intersection over union (IoU), sensitivity, and specificity. Intersection over Union (IoU) quantifies the degree of overlap between the anticipated and ground truth regions by dividing it by their combined area. Sensitivity, also called recall, measures the ratio of accurately detected actual positive events by the model. Specificity measures the accuracy of the model in correctly identifying negative instances.

As shown in Table 1, the log-cosh loss function works the best on the training data for our segmentation model. The log-cosh loss function is a function for regression problems designed to solve the limitations of mean squared and absolute errors. The loss is obtained by applying the natural logarithm to the hyperbolic cosine function to the difference between the actual and predicted values.

Loss Function	IoU	Sensitivity	Specificity
BCE Loss	0.5790	0.7824	0.9786
Focal Loss	0.5162	0.8832	0.9820
Dice Loss	0.5835	0.6351	0.9887
Tversky Loss	0.6383	0.7652	0.9916
Log-Cosh Loss	**0.6700**	**0.9275**	**0.9972**
Combo Loss	0.5969	0.6913	0.9952

Table 1: The accuracy of the segmentation model in 3 different metrics (IoU, sensitivity, and specificity) for different loss functions after training on floor plan image data.

The object detection model (YOLOv5) is fine-tuned using 90% of our annotated floor plan image dataset. The remaining 10% of the data is used as test data to evaluate the model performance on unseen images. Figure 9 shows the confusion matrix of the training results, where the horizontal axis refers to the annotation labels and the vertical axis refers to the prediction made by the model. We can observe that the model predicts the single-hinged doors correctly 54% of the time (true positive). However, it misses the prediction 42% of the time (false negative). Despite the minimal training data, the model performs sufficiently well enough for the mesh generation step.

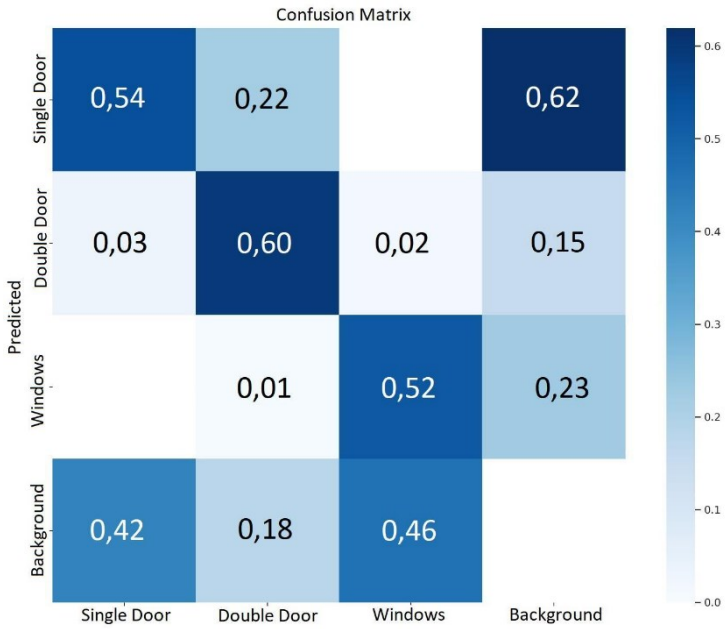

Figure 9: The confusion matrix of the training result of the object detection model for different classes. The number within the box denotes the prediction probability.

To assess the performance and accuracy of the approach, we overlay the original floor plan image onto the generated 3D model as the base. One way to achieve this is by adding a material (MTL) file to the top of the generated wavefront OBJ file.

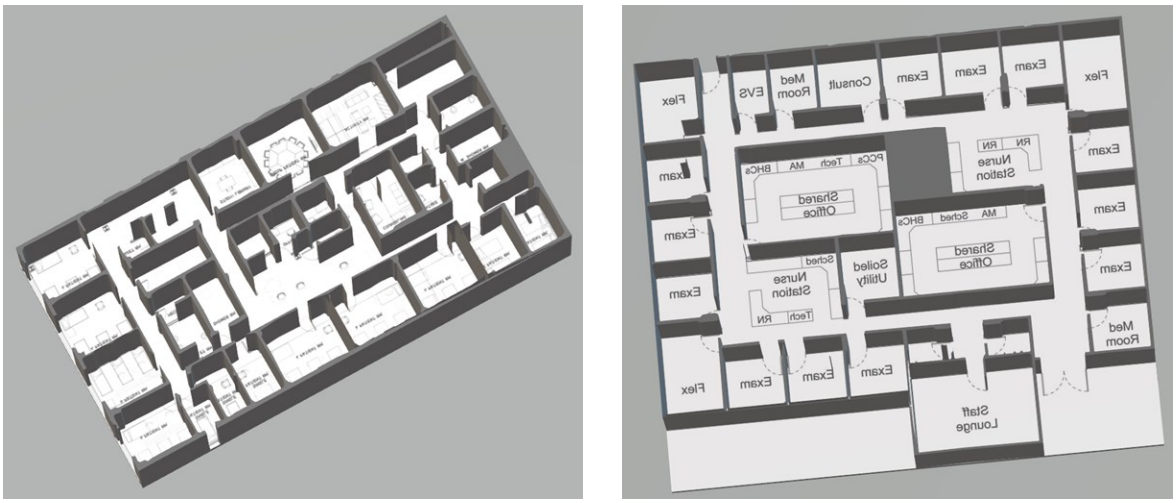

Figure 10: Example of mixed-use building layout in the analysis

We compare the results using the baseline method, cubicasa5k. Since CubiCasa5k's network uses a post-processing approach that assumes the Manhattan convention of wall perpendicularity does not hold in our dataset, we compare the raw segmentation results. The results from the CubiCasa5k network show that it can detect room boundaries when the wall follows a straight line, and the perpendicularity holds. In contrast, our method reliably recognizes the same cases while delineating the curved walls.

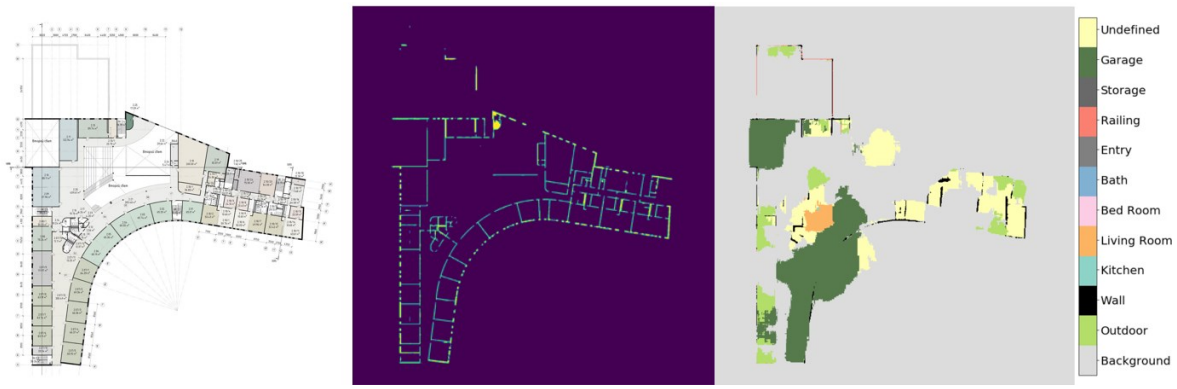

Figure 11: The mixed-use building in the test set and the prediction of the proposed model (in the middle) compared to the prediction from CubiCasa5k (in the right)

We provide qualitative results of 3D reconstruction from the test set of the dataset. The method can reconstruct the mixed-use building's overall geometry, capturing the complexities of curved heterogeneous walls.

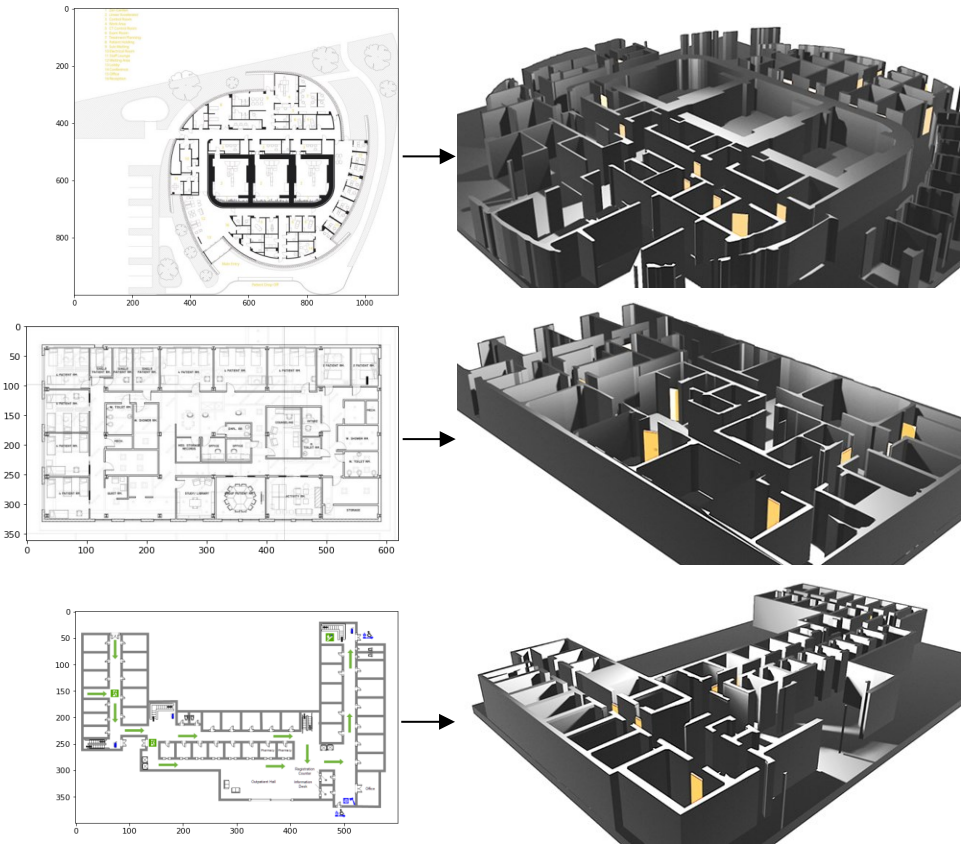

Figure 12: The 3D models are imported into Blender for post-processing. The walls and doors are applied with black and brown materials, respectively.

5. Conclusion

This research presents a novel approach to converting and reconstructing mixed-use building layouts into vectors and 3D models, outperforming previous methods on mixed-use building layouts. Two neural networks are used on images. The first network identifies the pixels that represent walls using a semantic segmentation method, while the second network identifies the openings in the image using object detection. The semantic segmentation mask is contoured and transformed into a mesh, then merged with the openings to generate a three-dimensional model. The proposed method is compared with a baseline showing the advances of the architecture in dealing with complicated layouts comprising increased floor area, incorporation of vertical transportation spaces like elevators and staircases, and non-standard wall forms and door/window sizes.

Some limitations of the current method are the limited capacity to precisely detect openings, explained by the difference in opening sizes in the training set and the low resolution of the images. The current methodology deals with the mask of the segmentation, recreating a contour that follows the prediction of the network. The consequence of this is that the walls are not smoothly represented but they are slightly irregular from the pixel-level prediction of the deep learning model, a next step is to have an additional post-processing step to smooth the prediction and vectorize it. Currently, the approach relies only on geometrical information without considering domain knowledge; this post-processing approach has been discarded because the mixed-use building use case makes it hard to generalize these kinds of buildings.

6. Acknowledgment

The research presented in this paper has received funding from the Bayerisches Verbundforschungsprogramm (BayVFP) des Freistaates Bayern - Förderlinie "Digitalisierung", grant number DIK0336/01 & 02.

References

Simonsen, C. P., Thiesson, F. M., Philipsen, M. P., Moeslund, T. B. (2021). "Generalizing Floor Plans Using Graph Neural Networks." In: Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, pp. 654–658. DOI: 10.1109/ICIP42928.2021.9506514.DOI: 10.1109/ICIP42928.2021.9506514.

Ronneberger, O., Fischer, P., & Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation." In: Proceedings of Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Part III (pp. 18). Springer International Publishing.

Park, S., Kim, H. (2021). "3DPlanNet: Generating 3D Models from 2D Floor Plan Images Using Ensemble Methods." Electronics, 10, 2729. https://doi.org/10.3390/electronics10222729.

Tan, M., & Le, Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks." In: Proceedings of the International Conference on Machine Learning (ICML). PMLR.

Liu, C., Wu, J., Kohli, P. and Furukawa, Y. (2017). Raster-to-Vector: Revisiting Floorplan Transformation. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.241.

Borrmann, A., König, M., Koch, C. and Beetz, J. (2018). Building Information Modeling: Why? What? How? Building Information Modeling, [online] pp.1–24. doi:https://doi.org/10.1007/978-3-319-92862-3_1.

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. arXiv:1904.01920 [cs]. [online] Available at: https://arxiv.org/abs/1904.01920 [Accessed Dec. 2022].

Kim, S., Park, S. and Yu, K. (2018). Application of Style Transfer in the Vectorization Process of Floorplans (Short Paper). [online] Dagstuhl Research Online Publication Server. doi:10.4230/LIPIcs.GISCIENCE.2018.39 .

Eberly, D. (2008). "Triangulation by ear clipping." Geometric Tools, 2002-2005.

Vidanapathirana, M., Wu, Q., Furukawa, Y., Chang, A.X., Savva, M. (2021). Plan2Scene: Converting Floorplans to 3D Scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 10733–10742.

Liu, C., Wu, J., Furukawa, Y. (2018). FloorNet: A Unified Framework for Floorplan Reconstruction from 3D Scans. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 201–217.

Lv, X., Zhao, S., Yu, X., & Zhao, B. (2021). Residential Floor Plan Recognition and Reconstruction. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 16712–16721. DOI: 10.1109/CVPR46437.2021.01644.

Wu, W., Fu, X. M., Tang, R., Wang, Y., Qi, Y. H., & Liu, L. (2019). Data-driven Interior Plan Generation for Residential Buildings. ACM Transactions on Graphics, 38(6). DOI: 10.1145/3355089.3356556.

Liu, Y., Lai, Y., Chen, J., Liang, L., & Deng, Q. (2020). SCUT-AutoALP: A Diverse Benchmark Dataset for Automatic Architectural Layout Parsing. IEICE Transactions on Information and Systems, E103D(12), 2725–2729. DOI: 10.1587/transinf.2020EDL8076.

Ito, Y. (2015). Delaunay Triangulation. In: Engquist, B. (eds) Encyclopedia of Applied and Computational Mathematics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-70529-1_314

Mei, G., Tipper, J. C., & Xu, N. (2013). Ear-clipping based algorithms of generating high-quality polygon triangulation. In Proceedings of the 2012 International Conference on Information Technology and Software Engineering: Software Engineering & Digital Media Technology (pp. 979-988). Springer Berlin Heidelberg.

Jadon, S. (2020, October). A survey of loss functions for semantic segmentation. In 2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB) (pp. 1-7). IEEE.

Mishra, S., Hashmi, K. A., Pagani, A., Liwicki, M., Stricker, D., & Afzal, M. Z. (2021). Towards robust object detection in floor plan images: A data augmentation approach. Applied Sciences, 11(23), 11174.