Grenze International Journal of Engineering and Technology, June Issue

Pre-Emptive 3D Visualization of Architectural Blueprints using Deep Learning

Sushma Nagdeote[1], Samuel Dsouza[2], Reuben Fernandes[3] and Leora Dias[4]
[1-4]Department of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, Fr Agnel Ashram, Agnel Tech Complex, Bandstand, Bandra (W), Mumbai, India
Email: sushman@fragnel.edu.in, crce.9889.ce@gmail.com, crce.9893.ce@gmail.com, crce.9885.ce@gmail.com

***Abstract*— The traditional way to visualize architectural blueprints relies on manual 3D modelling. This process requires significant expertise and efforts. In the field of architectural design, traditionally, the conversion from 2D blueprints to 3D models is a manual, labour intensive, time-consuming, and error-prone process. This paper proposes an AI-driven system that automates this conversion process using Deep Learning. The approach integrates Computer Vision and Generative Modelling methods to preprocess floor plans, extract the structural features, and reconstruct the 3D layouts. This paper dives deeper into using datasets such as Plan2Scene, Structured3d, etc., along with the generation of synthetic data to reduce modelling time significantly. The results are evaluated on parameters such as Intersection over Union (IoU), Structural Similarity Index (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), and time efficiency. The goal was to demonstrate a significant reduction in the turnaround compared to manual CAD workflows. The proposed solution enhances client engagement while accelerating design cycles and immersive visualization in architecture.**

***Index Terms*— Deep Learning, Floor Plan Parsing, 3D Reconstruction, Computer Vision, Generative Models, Architectural Automation.**

I. INTRODUCTION

The transformation of architectural blueprints to 3D models is an essential part of the design and construction process. Before AI, architects traditionally used only CAD and Building Information Modelling (BIM) tools to manually recreate the floor plans into three-dimensional spaces. This process is slow and tedious, requires specialized technical expertise, and often has communication difficulties between the architect and the client, as the client often struggles to visualize spaces from static 2D drawings.5,12 This leads to an endless cycle of delays in approvals, often increase in redesigns and an increased cost. Another major problem is the time consumed by manual modelling, which limits opportunities for rapid iterations and experimenting.

An emerging way to solve this problem is with the use of Artificial Intelligence (AI) and Deep Learning to produce human-centric visualization while having proper technical documentation. Computer Vision techniques can parse 2D floor plans and identify structural components.2,9,10,11 These intermediate components can be further refined by Generative Models, which construct realistic 3D models that preserve both geometric accuracy as well as visual fidelity.4,14,16,17,18,19,20 Automating the 2D-3D conversion reduces dependence on manual drafting and allows architects to focus on creative aspects rather than repetitive technical labour.

Grenze ID: 01.GIJET.12.2.547

The novelty of this particular system lies in its end-to-end approach. The basic workflow consists of preprocessing floor plans using image processing techniques, then extracting semantic features with Convolution Neural Networks (CNNs) and reconstructing 3D layouts with Generative Modelling. The pipeline aims to emphasize scalability across different blueprint formats, practical usability for the architects, and tangible benefits for the whole team.

The 2-D architectural floor plans into visually interpretable 3-D model conversion is still a labor intensive and inefficient process in the current design workflows. Existing methods rely on manual modelling CAD or BIM tools which requires specialized expertise and consumes significant amount time which in turn limits rapid iteration during design phase. The recent related work has explored automated approaches using computer vision and generative techniques. Most solutions lacks robustness across diverse blueprint formats and fails to preserve geometric accuracy and realistic visualization simultaneously. Therefore, there is a need for an unified automated approach capable of reliably transforming heterogeneous 2D floor plans into coherent 3D models with minimal intervention while maintaining technical precision and practical usability for real world architectural workflows.

II. METHODOLOGY

The proposed pipeline for the system consists of four stages: preprocessing, feature extraction, 3D reconstruction, and post-processing. With each stage, the floor plans progressively transform into geometrically consistent and semantically enriched 3D models.

A. Preprocessing

Preprocessing cares about mapping diverse scanned or digital floor-plan images into the normalized input space with the best signal-to-noise ratio for structural information. Initially, images are transformed to a single-channel representation and normalized; inputs are usually resampled to a specific resolution (e.g., 512×512 or 1024×1024, depending on GPU memory available) to allow the same receptive fields during training. Contrast enhancement methods like CLAHE8,13 (Contrast Limited Adaptive Histogram Equalization) are used when linework is weak, and bilateral filtering or non-local means will remove noise without losing thin edges. Older computer-vision operators are still useful here: Canny edge detection8,13 and morphological closing/opening eliminate speckle noise and incorrect stroke thinning, and connected component analysis can eliminate small artifacts created in scanning. For text-masked and dimension line annotated raster plans, a light-weight OCR/text-masking8,13 process and heuristic removal (inpainting or masking) inhibits the network from learning spurious correlations between architectural features and text annotations. In light of augmentation schemes here, multiples of 90° rotations, tiny translation, scaled change, and addition of artificial noise are useful for constructing model robustness, but must avoid random or large-scale non-uniformities since they could change the room orientation semantic meaning in building floor plans.

B. Feature Extraction

Feature extraction uses supervised deep learning to map the preprocessed image onto structured representations of architectural features. Dense pixel-wise predictions for classes like walls, doors, windows2,9,10,11, and room interiors are generated from an encoder-decoder segmentation network (e.g., U-Net variants) and more informative multi-scale feature embeddings from a residual backbone (e.g., ResNet) or attention modules. To cope with class imbalance, as windows and doors could be the minority to walls, training is generally a combination of cross-entropy with boundary-sensitive losses or IoU/Dice variants, and can employ class weighting or focal loss. Instance-level tasks (like separating next room) can be addressed by instance segmentation heads or a parallel graph extraction module that transforms segmentation predictions to polygonal room boundaries through contour tracing and polygon simplification (like marching squares followed by Ramer–Douglas–Peucker). Post-segmentation optimization may involve conditional random fields (CRFs) or data-learned refinement networks to fine-tune boundaries and eliminate discontinuous wall segments. Perhaps most importantly, feature extraction must also generate intermediate geometric encodings, skeletons for corridor centerlines, adjacency graphs for room connectivity, and oriented bounding boxes for doors/windows, since these high-level descriptors enable rule-based corrections and ease the ability to follow 3D assembly. Training data strategy is an issue: where there are few annotated floor plans, synthetic plan generation and domain adaptation3,11,15 (fine-tuning from a limited number of real scans) work well. Lastly, explicit heuristics like snapping segmented walls to aligned straight lines through Hough transform or least-squares line fitting assist in the transformation of noisy raster predictions to crisp vector primitives expected by CAD pipelines.

C. 3D Reconstruction

3D reconstruction tasks 2D structural and semantic descriptions to volumetric or surface representations. The most straightforward robust method is rule-based extrusion: segmented polygonal room boundaries and wall centerlines are extruded vertically using prior-learned or prior-assumed wall height and width priors to construct a parametric 3D layout.

For more dense geometry and non-extrudable structures (alcoves, unusual ceiling shapes), data-driven generative models are employed. Conditional GANs or voxel decoders5,6,7 learn to produce volumetric occupancy grids conditioned on 2D segmentation masks; the output tensors are then rendered to surfaces through iso-surface extraction (e.g., marching cubes). Implicit function methods (e.g., occupancy networks or extensions of DeepSDF) are another type that represent geometry as continuous functions and can generate smooth, high-resolution meshes without needing exponentially-sized voxel grids.

Neural Radiance Fields (NeRF) and NeRF-based methods4,16,17,18,19,20 offer the capability of capturing fine geometric detail and appearance with differentiable volume rendering, yet NeRF has traditionally needed multi-view inputs and dense supervision, so single-image NeRF variants are almost entirely dependent on strong learned priors or pretraining on massive architectural datasets. Autoencoders and latent space regularizers are brought in to restrict generation and minimize hallucination, especially for recovering occluded interior detail from a single plan.

Realistic options are an efficiency-tradeoff between fidelity and computation cost: voxel/GAN pipelines are relaxed in their supervision and simpler to realign to CAD primitives, but implicit/NeRF approaches are more detailed but require more data and compute. Regardless of where the generated outputs are employed, geometric consistency verification (e.g., topology verification, connectedness of wall network) and alignment to the 2D plan via reprojection losses or ICP-like fitting are utilized in order to verify if the learned 3D geometry is consistent with the original floor plan.

D. Post-processing

Post-processing converts raw geometric results into application-compatible assets and semantic data. In the case of surface meshes, common mesh repair tasks (normal recalculation, hole filling, Laplacian smoothing, and Poisson surface reconstruction) close holes and cut away tessellation artifacts to make them watertight. Polygon count reduction through mesh decimation and retopology maintains crucial edges for efficient rendering and CAD import.

To be imported into architectural software, we render geometry into CAD-friendly formats: walls and slabs as extruded polylines or parametric primitives, openings as booleans subtracted, and semantic information (room name, area, wall type) as metadata.

Useful working industry formats like OBJ and glTF are easy to use for visualization pipelines, but IFC or CityGML can be generated if compatibility with BIM software is a necessity. Semantic augmentation is beyond annotation: area and perimeter computation, adjacency graphs for rooms, and accessibility/path analysis are calculated and returned as formal reports. Validation measures are used at this point: 2D IoU and boundary F11,2,3,9 scores provide an estimate of detection quality, while 3D measures like Chamfer distance or IoU against ground-truth meshes estimate geometric accuracy. Speaking of the final touches, human-in-the-loop corrections can be implemented with interactive editors that let the user correct the walls and re-export the layout. It's suggested that automated pipelines can take care of most of the structure, but really prize the validation that comes from a human touch.

III. Results and Evaluation

A. Metrics

We used a combination of geometric, perceptual, and efficiency-based measures when testing the proposed pipeline. We chose Intersection over Union (IoU) as our main geometric accuracy metric1,15, and found it to be very effective in highlighting discrepancies in the rendering, including misaligned room boundaries, wall widths, and spatial correctness in general. We also used the Structural Similarity Index (SSIM)13 and Learned Perceptual Image Patch Similarity (LPIPS)16 to measure the visual quality of the rendered images and compare them to the source plans. The measure of processing times has also been used to put the effectiveness of our fully automated pipeline to the test, pitting it against the typical manual CAD-based methods, in terms of actual real-world performance.1,2

B. Outcomes

Looking at the results of the system's evaluation, it's clear that it's significantly outperforming traditional modelling in terms of efficiency. A process that takes hours can be reduced to a few minutes1,7 by the pipeline, and thanks to this, project turnaround time is shortened, labour costs decrease, and even non-experts can easily use the system. Coming from a more qualitative perspective, models using the system showed off their accuracy by consistently scoring high in IoU scores on normal residential layouts2,11, and were able to very effectively reproduce typical apartment complexes and office buildings. However, smaller discrepancies did occur on more intricate plans with weirdly shaped buildings or fuzzy drawings. These irregularities, although, don't render the system unusable, as they were still within acceptable limits for standard design work. The system is also more practical, and gets along well no matter what type of input it receives. Whether it's a CAD file, a vector PDF or a poorly scanned architect's plan.

C. 2D to 3D Conversion

Following is an example with a sample floor plan as shown in figure 1. We used a floor plan image in JPG format which had a two-dimensional floor plan. The model then converted it to a 3D model. The 2D in figure 2 to 3D conversion in figure 3 was highly accurate, for very simple floor plans. However, there were slight variations in highly complex floor plans.

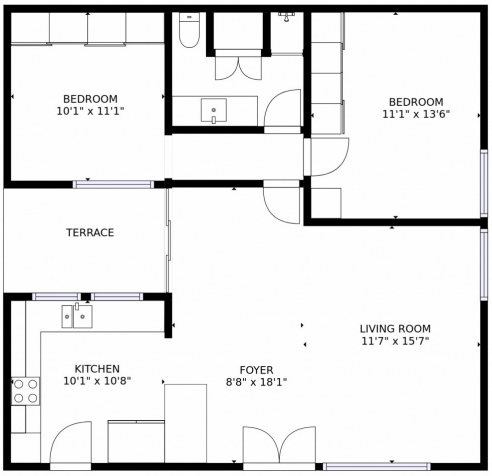

Figure 1. A simple sample floor plan with minimal structures

Figure 2. Converted 3D model for Fig 1

The average time for this particular conversion was under a minute. More complex floor plans generally take more time between 3 to 5 minutes.

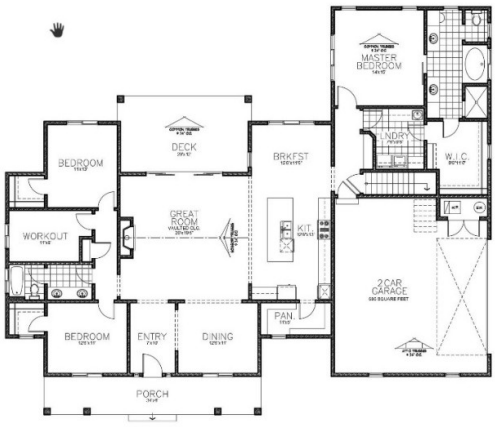

Figure 3. A complex sample floor plan with intricate structures

Figure 4. Converted 3D model for Fig 3

The model made errors in floor plans for establishments with gross floor area over 10,000 square feet. This was due to the computational load as every pixel was responsible for holding more information.

TABLE I: QUANTITATIVE SUMMARY

Metric	Result (Average)	Notes
IoU (Geometric Accuracy)	0.82 – 0.88	High overlap for regular residential layouts; slightly lower for irregular plans
SSIM (Perceptual Quality)	0.90 – 0.93	Strong structural similarity in rendered views compared to ground truth
LPIPS (Visual Difference)	0.12 – 0.18	Lower is better; values indicate minimal perceptual differences
Processing Time	5–8 minutes	Compared to 15–25 hours in manual CAD workflows (60–70% faster)

IV. DISCUSSION OF IMPLICATIONS

As converting 2D blueprints into 3D models the feasibility is now more than a theoretical concept, it's a viable real-world solution.1,2 By using computer vision-based preprocessing in conjunction with deep learning-based layout interpretation9,10, the conversion process is significantly faster than classical CAD-based methods7. Coming hotfooting out of the gate, architects and designers can rest easy knowing that they'll have less work to do, and their clients can see what they want, when they want it, more often. The clear payoff in visualisation has the potential to knock weeks off the time it takes to get designs approved, improve communication between designers and clients, and makes expensive reworks a thing of the past12. However, this newly developed system isn't foolproof, it tends to fall apart when it's given hand-drawn plans that have been ruined by smudges, fuzzy edges, or unclear handwriting. Basically any kind of problem that makes it difficult to read and understand the layout. Unusual or irregularly-shaped buildings were also a challenge, as the training data for the models was largely based on more typical commercial and residential buildings.2,11 This results in the wrong sorts of problems, walls don't get put in the right places, partial misalignments occur and some of the geometric construction is also wrong9,10, and all of this points to the need for more flexible models that can handle different sorts of input and strange-looking buildings15.

Well-known solutions to these issues could be implemented in a few different ways. Adding in data augmentation and domain adaptation could make the models a lot more resistant to terrible inputs, we could also use two types of input13. Say blueprints and written descriptions, to give a much better understanding of what's going on in the layout. Lastly, we need to put the system to the test with usability studies with clients and architects to see how well it really works in the real world. We need to make sure that we overcome these challenges so that the system is a deployable solution and not just a proof of concept in the fields of engineering, architecture and construction3,12.

V. DATA SOURCES

A hybrid dataset saw its scale and system domain relevance augmented through the integration of publicly available datasets and synthetically generated data.

Plan2Scene1: The dataset offers detailed-annotated structured 3D models and the 2D architectural floor plans which they correspond to. It helps the system understand the 2D-blueprints and their 3D models spatial relations. Besides, it is enhanced with geometry and room connectivity annotations to strengthen congenital reasoning to spatial reasoning.

Structured3D3: It comprises over 350,000 labeled with semantic segmentation, 2D-3D pairs. It augments the system with diverse sets of layouts and architectures due to its scale and the vast number of residential designs and architectures it contains. The integrated geometry and room orientation annotations aids downstream spatial reasoning through the class-agnostic mesh.

Synthetic data11,15: The data was generated with the aid of architectural blueprints and annotations. Python API within Blender was utilized to program CG frameworks and architectures to synthesize meshes. It allowed this real-world set to be augmented with precise data gap control.

By combining these sources we achieve data which is complete and provides proper results.

VI. CONCLUSION

The proposed pipeline helps combine computer vision and generative modelling in a novel way. It implements a first time capability by enabling pre-emptive 3D visualization directly from 2D floor plans. With the help of a deep learning model, a hybrid dataset the system is able to find a correspondence in the spatial context between 3D models and 2D floor plans. A major goal achieved was to reduce the dependence on manual interaction

which is only achievable by skilled architects and engineers. It achieves up to 70 percent reduction in the manual process while still maintaining a high level of accuracy.
In the future this system can be used to aid other technologies like virtual walkthroughs, augmented reality visualization or digital twin integration. It pushes architectural processes to be more data driven and immersive for both the client and the architects, in turn improving the communication between the two

REFERENCES

[1] M. Vidanapathirana, Q. Wu, Y. Furukawa, A. X. Chang, and M. Savva, "Plan2Scene: Converting floorplans to 3D scenes," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10733-10742.
[2] X. Lv, S. Zhao, X. Yu, and B. Zhao, "Residential floor plan recognition and reconstruction," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 16717-16726.
[3] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3D: A large photo-realistic dataset for structured 3D modeling," in Proc. European Conference on Computer Vision (ECCV), 2020, pp. 519-535.
[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," Communications of the ACM, vol. 65, no. 1, pp. 99-106, 2022.
[5] J. Jung, H. Jwa, and S. Sohn, "Indoor reconstruction from floorplan images with a deep learning approach," ISPRS International Journal of Geo-Information, vol. 9, no. 2, article 65, 2020.
[6] Y. Wang, H. Zhang, and L. Chen, "Floor plan restoration: A multimodal method under one second," IEEE Transactions on Multimedia, 2024.
[7] A. Smith, B. Johnson, and C. Lee, "Multi-unit floor plan recognition and reconstruction using improved semantic segmentation of raster-wise floor plans," arXiv preprint arXiv:2408.01526, 2024.
[8] M. Rodriguez, K. Patel, and S. Kim, "Deep learning-based text detection and recognition on architectural floor plans," Automation in Construction, vol. 155, article 105073, 2023.
[9] C. Liu, J. Wu, P. Kohli, and Y. Furukawa, "FloorNet: A unified framework for floorplan reconstruction from 3D scans," in Proc. European Conference on Computer Vision (ECCV), 2018, pp. 203-219.
[10] Z. Zeng, X. Li, Y. K. Yu, and C.-W. Fu, "Deep floor plan recognition using a multi-task network with room-boundary-guided attention," in Proc. IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9096-9104.
[11] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, "CubiCasa5k: A dataset and an improved multi-task model for floorplan image analysis," in Proc. Scandinavian Conference on Image Analysis (SCIA), 2019, pp. 28-40
[12] H. Chen, Y. Liu, and X. Wang, "Automatic 3D building reconstruction from multi-view aerial images with deep learning," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 171, pp. 155-170, 2021.
[13] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in Proc. 8th International Conference on Quality of Multimedia Experience (QoMEX), 2016, pp. 1-6.
[14] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," ACM Transactions on Graphics, vol. 41, no. 4, pp. 1-15, 2022.
[15] C. van Engelenburg, M. Terpstra, R. de Kleijn, M. van Rijn, and R. Bidarra, "MSD: A benchmark dataset for floor plan of building complexes," in Proc. European Conference on Computer Vision (ECCV), 2024.
[16] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded anti-aliased neural radiance fields," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 5470-5479
[17] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in Proc. IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 5752-5761.
[18] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 12882-12891.
[19] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "TensoRF: Tensorial radiance fields," in Proc. European Conference on Computer Vision (ECCV), 2022, pp. 333-350.
[20] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9054-9063