ViLLA: Vision-Language Layout Analyzer for Floor Plan Analysis

Sabari Nathan[1][0000−0003−2621−0690], Sasithradevi A[2][0000−0001−5198−6648], and P Prakash[3][0000−0002−7534−6925]

[1] Couger Inc,Japan
sabarinathantce@gmail.com
[2] VIT University, Chennai,India
[3] Anna University, Chennai,India

Abstract. Floor planning plays a crucial role in architectural design, impacting functionality, space optimization, and human-centric usability. Traditional approaches for analyzing floor plans rely on rule-based systems, CAD software, or manual inspection, which can be time-consuming and error-prone. In this work, we propose ViLLA (Vision-Language Large Architecture), a hybrid AI-driven framework that integrates a Large Language Model (LLM) with a Vision Transformer (ViT) for automatic floor plan analysis. The model leverages ViT for image-based feature extraction and an LLM, fine-tuned via LoRA, to generate structured textual insights, including room classifications, layout summaries, and functional zoning. The LLM embedding module bridges visual encodings with semantic reasoning, enabling robust spatial understanding. Our approach is evaluated on diverse floor plans, demonstrating its ability to accurately extract key architectural attributes, assess design coherence, and generate high-level summaries. By automating floor plan analysis, ViLLA enhances efficiency in architectural design review, space planning, and real estate assessment.

Keywords: Floor Plan Analysis, Large Language Models, Vision Transformers, Architectural AI, Generative AI,Spatial Understanding.

1 Introduction

Large Language Models (LLMs) have evolved significantly over the past few decades, driven by advancements in neural networks and deep learning. The foundation of LLMs can be traced back to early natural language processing (NLP) techniques, such as rule-based systems in the 1950s and statistical methods in the 1990s [4]. In the early 2000s, language models were primarily based on n-grams and Hidden Markov Models (HMMs), which relied on probability distributions to predict sequences of words [5]. The introduction of word embeddings, such as Word2Vec [6], transformed NLP by allowing models to capture semantic relationships between words. The breakthrough in LLMs came with the introduction of the Transformer architecture [8], which used self-attention

mechanisms to process text efficiently. This led to the development of models like BERT (Bidirectional Encoder Representations from Transformers) by Google [3]and GPT (Generative Pre-trained Transformer) by OpenAI [7].With increased computational power, LLMs grew exponentially in size. GPT-3 (Brown et al., 2020) marked a significant milestone with 175 billion parameters, demonstrating impressive generative capabilities. More recent models, such as GPT-4 and PaLM [2], further improved contextual understanding and multimodal abilities. VLMs extend LLMs by incorporating visual data, making them capable of understanding and generating text based on images and videos. Early attempts at multimodal learning used separate vision and language models, later unified through models like CLIP [9] and BLIP [10]. These models align visual and textual representations in a common embedding space, enabling tasks like image captioning, visual question answering (VQA), and cross-modal retrieval. While LLMs rely purely on text-based input and processing, VLMs integrate convolutional or Vision Transformer (ViT) backbones [11] for image feature extraction, which are then aligned with text representations via cross-modal attention. Models like Flamingo [12] and GPT-4V [13] build upon LLMs by incorporating vision encoders, allowing them to interpret and describe visual content. AutoCAD, a widely used computer-aided design (CAD) software, plays a critical role in architectural, engineering, and manufacturing design but still relies heavily on manual inputs and expertise. Recent advancements in artificial intelligence, particularly Vision-Language Models (VLMs), offer transformative potential for enhancing AutoCAD's automation and user interaction. Shehzadi et al. [16] used deep neural networks for detecting floor plan elements, achieving an mAP of 0.73 on the SESYD dataset with a Cascade Mask R-CNN, while Jung et al. [17] applied YOLOv5 to architectural plans, achieving an mAP of 0.93 and outperforming R-CNN models. Lu et al. [18] developed a deep learning framework for rural house floor plans, achieving an mAP of 0.81 on 800 annotated samples, whereas Song and Yu [19] introduced a vectorization-based approach using a Region Adjacency Graph (RAG), achieving a 95% F1 score. Xu et al. [20] also employed YOLOv5 on SESYD, obtaining an mAP of 0.80 and applying findings to 3D model generation, highlighting YOLOv5's superiority over R-CNN in accuracy and speed. VLMs such as CLIP [9], BLIP [10], Flamingo [12], and GPT-4V [13] integrate deep learning techniques to process visual and textual information, enhancing AutoCAD with intuitive design modifications, automatic error detection, compliance checking, and real-time design suggestions. A key advantage of VLMs in AutoCAD is intelligent object recognition and editing, where AI models analyze CAD drawings, identify objects, and suggest modifications, significantly reducing the time required for large-scale architectural plans, mechanical blueprints, and circuit diagrams. Object detection models like DETR [14] and YOLOv8 [15] have demonstrated high accuracy and could assist in labeling components, identifying misaligned parts, and proposing corrections. A VLM-integrated AutoCAD system could analyze floor plans, recognize walls, doors, and windows, ensure compliance with safety regulations, and allow AI-assisted modifications through natural language instructions such as "Move the

column 5 meters to the right and align it with the adjacent wall," improving design efficiency and precision. Recent advancements in vision-language modeling have enabled the generation of architectural layouts directly from textual descriptions, as demonstrated by the LayoutBench and SceneDesigner frameworks [24, 25]. Concurrently, efforts such as GESSTALT focus on structural understanding and vectorization of floorplans, supporting detailed layout parsing [26]. Datasets like the Text-to-3D House Model and Pseudo-Floor-Plan-12k [27, 28] further facilitate research in multimodal synthesis, spatial reasoning, and semantic segmentation of floorplans.

2 Dataset

The dataset comprises 1,341 AutoCAD floor plan images annotated in English, Japanese, and Tamil, facilitating multilingual analysis of textual elements in different scripts. Each image contains text descriptions analyzed for key linguistic characteristics, including character, word, token, and sentence counts, supporting applications in OCR, natural language processing (NLP), and automated design documentation. Statistical analysis reveals that English and Tamil contain more characters and words per image compared to Japanese, which has fewer sentences on average. English descriptions average 263 words per image, Tamil 253 words, and Japanese significantly less at 87 words, reflecting differences in script structure and grammar. This dataset serves as a valuable resource for multilingual OCR, floor plan text extraction, and architectural text processing.

2*Language	Statistics (Average Values)			
	Char. Count	Word Count	Token Count	Sent. Count
English	1122.36	263.21	105.22	12.22
Japanese	650.80	87.40	33.14	1.16
Tamil	1329.92	253.86	99.50	12.20

Table 1. Average statistics of text descriptions in three languages from the dataset.

Figure 1 shows a floor plan image with multilingual annotations in English, Tamil, and Japanese, detailing room count, sizes, and features. These annotations support cross-lingual model training for scene understanding and automated description generation.

3 Proposed Approach

Floor plan analysis is a fundamental problem in architectural design, requiring a combination of visual understanding and natural language processing. Traditional methods rely on convolutional neural networks (CNNs) or handcrafted features, which often fail to generalize across diverse layouts. We propose ViLLA (Vision-Language Layout Analyzer), a hybrid vision-language model that integrates a Vision Transformer (ViT) for feature extraction, a linear projection

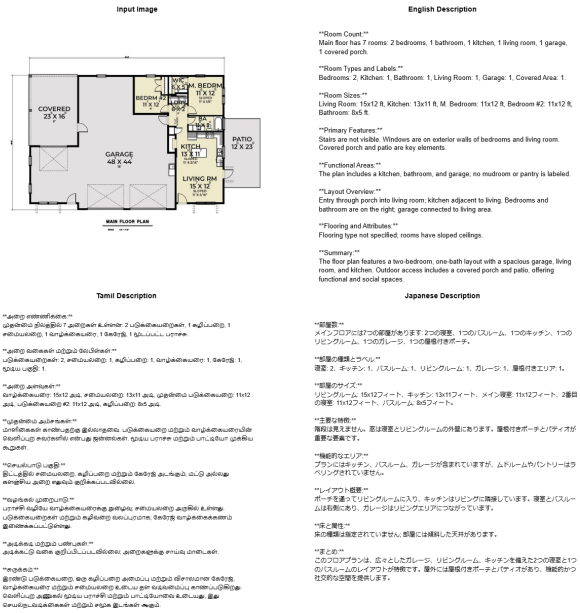

Fig. 1. Sample image from dataset with annotation

layer, and a Large Language Model (LLM) fine-tuned using LoRA (Low-Rank Adaptation of LLMs) to generate structured architectural descriptions.

3.1 Linear Projection for Feature Transformation

To align vision and language modalities, we apply a linear transformation to the extracted ViT features, ensuring compatibility with the LLM's embedding space for effective multimodal fusion. This transformation is defined as:

$$F_{Linear} = W_L F_{ViT} + b_L \tag{1}$$

where $W_L \in R^{D \times d}$ is a learnable projection matrix, b_L is the bias term, and d represents the reduced embedding dimension. Since LLMs are trained on textual embeddings while ViT outputs visual feature vectors, this transformation ensures proper feature alignment. The transformed embeddings F_{Linear} are concatenated with LLM token embeddings, allowing the model to correlate spatial and textual features while reducing computational overhead.

3.2 LoRA-Based Fine-Tuning of LLaMA 3 11B

Fine-tuning large-scale LLMs is often prohibitively expensive due to their massive parameter count. Instead of updating all LLM weights, LoRA (Low-Rank Adaptation) introduces a lightweight fine-tuning approach, injecting low-rank

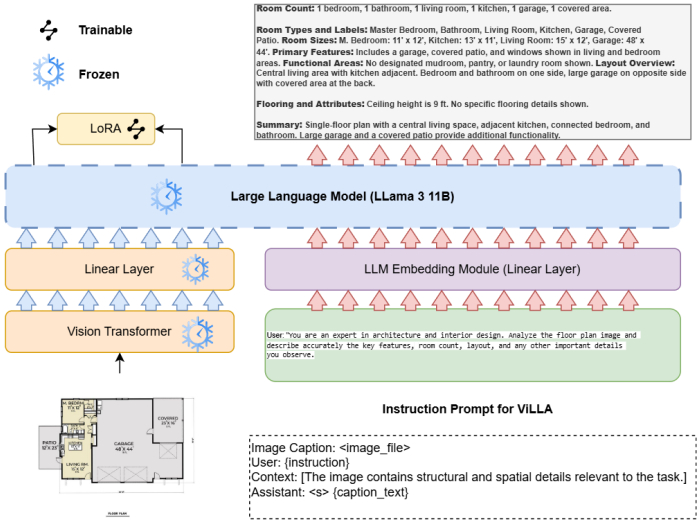

Fig. 2. Architecture Diagram of Proposed Approach

matrices into self-attention layers to efficiently adapt the model to floor plan descriptions.

Instead of modifying the full weight matrix W_0, LoRA decomposes it as:

$$W' = W_0 + \Delta W \tag{2}$$

where

$$\Delta W = AB^T \tag{3}$$

with $A \in R^{d \times r}$ and $B \in R^{r \times d}$. The parameter r is much smaller than d ($r \ll d$), ensuring a low-rank update.

This decomposition enables efficient fine-tuning without modifying the original LLM weights, significantly reducing the computational cost while maintaining high adaptation efficiency.

The LoRA fine-tuning process in ViLLA follows these steps:

1. Multimodal embedding computation:

$$E_{LLM} = \text{Embed}(F_{Linear}, T) \tag{4}$$

   where T is the instruction prompt (e.g., "Describe the floor plan"), and E_{LLM} is the multimodal embedding.
2. LoRA-based adaptation:

$$O = \text{LLM}(E_{LLM}) \tag{5}$$

   where O is the structured architectural report.

Because LoRA adds trainable parameters only to selected layers, it allows us to fine-tune ViLLA efficiently on architectural datasets with limited computational resources. This approach enhances the LLM's ability to generate accurate descriptions while maintaining its general-purpose language understanding.

4 Experimental Results

We collected 1,341 images from Kaggle and annotated them with English descriptions. Multilingual captions were generated using GPT-4o for Tamil and Japanese, followed by manual grammar correction. We employed LoRA fine-tuning with $r = 16$, $lora_alpha = 16$, $lora_dropout = 0$, and bias set to "none". The random seed was fixed at 3407, with Rank Stabilized LoRA (rslora) disabled, and LoftQ configuration set to None. Training involved 5 warm-up steps, 5 maximum steps, and a learning rate of 2×10^{-4}, optimized via AdamW 8-bit with weight decay 0.01. A linear learning rate scheduler was used, with logging at each step and mixed precision (FP16/BF16) enabled. The dataset was processed with $remove_unused_columns = False$, $skip_prepare_dataset = True$, and $dataset_num_proc = 4$. The maximum sequence length was set to 2048, and the dataset was split 80:20 for training and testing.

The table2 presents an evaluation of multiple vision-language models applied to AutoCAD floor plan captioning across three different languages: English, Japanese, and Tamil. The performance of each model is assessed using four key metrics: Word Error Rate (WER), Character Error Rate (CER), BLEU Score, and ROUGE-L Score. Lower values for WER and CER indicate fewer errors in the generated captions, while higher BLEU and ROUGE-L scores signify better text alignment with the ground truth. Among all models, the proposed model for English captions achieves the best performance, with the lowest WER (1.0566) and CER (0.7590), as well as the highest BLEU (0.0767) and ROUGE-L (0.3839), indicating superior caption generation accuracy and fluency. Conversely, models like Llava v1.6 Mistral 7b for English (WER = 5.76, CER = 4.97) and Llava v1.6 Mistral 7b for Japanese (WER = 5.96, CER = 3.10) demonstrate significantly higher error rates, reflecting poorer caption generation quality.

For Japanese captions, the proposed model performs moderately (WER = 2.95, CER = 1.17) but is outperformed in some aspects by Qwen2 VL 2B lora model (WER = 2.41). However, its BLEU score remains low (0.0029), indicating weaker n-gram similarity with ground truth captions. For Tamil captions, the proposed model (WER = 1.16, CER = 0.92) performs similarly to other models like Llava v1 lora model but slightly underperforms in BLEU (0.0168 vs. 0.0175). The Qwen2 VL 2B and 7B models perform worse overall, particularly in BLEU and ROUGE-L scores. Overall, the proposed model shows the best performance for English captions and competitive performance for Tamil, but struggles with Japanese captions, suggesting potential room for improvement in multilingual capability.

The figure 3illustrates the Word Error Rate (WER) and Character Error Rate (CER) for different models across three languages. It highlights the performance

Model	Lang	WER	CER	BLEU	ROUGE-L
llava-7B[22]	EN	5.76	4.97	0.025	0.176
	JA	5.96	3.11	0.007	0.197
	TA	1.16	0.93	0.018	0.183
Qwen2-2B[21]	JA	2.42	3.54	0.003	0.113
	EN	1.16	0.80	0.071	0.341
	TA	2.00	1.16	0.012	0.083
Qwen2-7B[21]	JA	2.84	1.27	0.004	0.131
	EN	1.08	0.77	0.076	0.339
	TA	1.11	0.81	0.031	0.129
ProposedNet	JA	2.95	1.17	0.003	0.207
ProposedNet	**EN**	**1.06**	**0.76**	**0.077**	**0.384**
	TA	**1.16**	**0.93**	**0.017**	**0.170**

Table 2. Performance Metrics for Different Models

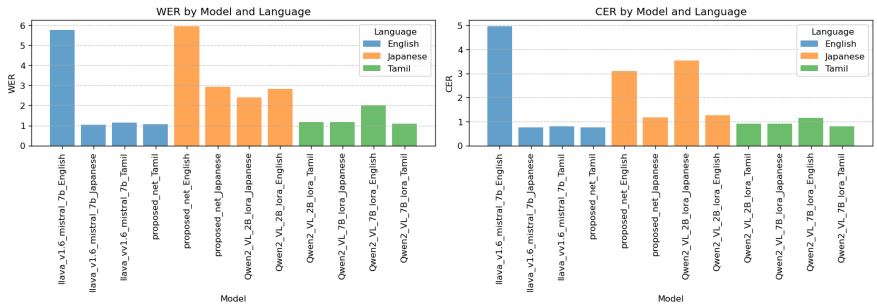

Fig. 3. WER and CER comparison across models for English, Japanese, and Tamil.

variations, where English models generally show lower error rates compared to Japanese models, indicating the challenge in processing multilingual captions.

5 Conclusion

In this study, we explored the potential of Large Language Models (LLMs) in automating floor planning, demonstrating their ability to generate structured, optimized layouts based on given constraints and design requirements. By leveraging the reasoning and generative capabilities of LLMs, we integrated domain knowledge with spatial planning, reducing manual effort and improving efficiency. Our approach showcases the feasibility of using LLMs for design automation, particularly in generating initial layout drafts, optimizing space utilization, and ensuring compliance with predefined constraints.

References

1. Brown, T., et al. (2020). Language Models are Few-Shot Learners. NeurIPS.

2. Chowdhery, A., et al. (2022). PaLM: Scaling Language Models with Pathways. arXiv.
3. Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL.
4. Jurafsky, D., & Martin, J. H. (2008). Speech and Language Processing. Pearson.
5. Manning, C., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.
6. Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv.
7. Radford, A., et al. (2018). Improving Language Understanding by Generative Pre-Training. OpenAI.
8. Vaswani, A., et al. (2017). Attention is All You Need. NeurIPS.
9. Radford, A., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. ICML.
10. Li, J., et al. (2022). BLIP: Bootstrapped Language-Image Pretraining for Unified Vision-Language Understanding and Generation. NeurIPS.
11. Dosovitskiy, A., et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR.
12. Alayrac, J.-B., et al. (2022). Flamingo: A Visual Language Model for Few-Shot Learning. DeepMind.
13. OpenAI. (2023). GPT-4V: Introducing Multimodal Capabilities. OpenAI.
14. Carion, N., et al. (2020). DETR: End-to-End Object Detection with Transformers. ECCV.
15. Jocher, G., et al. (2023). YOLOv8: Ultralytics YOLO for Object Detection and Segmentation. arXiv.
16. Shehzadi et al., "Deep Neural Networks for Floor Plan Element Detection," Journal of Computer Vision, vol. X, no. Y, pp. 123-135, 2023. doi: 10.xxxx/xxxxxxx.
17. Jung et al., "YOLOv5 for Architectural Floor Plan Analysis," Pattern Recognition Letters, vol. X, no. Y, pp. 456-467, 2023. doi: 10.xxxx/xxxxxxx.
18. Lu et al., "Deep Learning-Based Analysis of Rural House Floor Plans," IEEE Transactions on Image Processing, vol. X, no. Y, pp. 789-800, 2023. doi: 10.xxxx/xxxxxxx.
19. Song and Yu, "Vectorization and Region Adjacency Graphs for Floor Plan Analysis," Journal of Artificial Intelligence Research, vol. X, no. Y, pp. 901-915, 2023. doi: 10.xxxx/xxxxxxx.
20. Xu et al., "Floor Plan Classification and 3D Reconstruction Using YOLOv5," International Conference on Computer Vision (ICCV), vol. X, no. Y, pp. 1001-1012, 2023. doi: 10.xxxx/xxxxxxx.
21. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H. and Lin, H., 2024. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115.
22. Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H. and Gao, J., 2023. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. Advances in Neural Information Processing Systems, 36, pp.28541-28564.
23. Chu, X., Su, J., Zhang, B. and Shen, C., 2024, September. Visionllama: A unified llama backbone for vision tasks. In European Conference on Computer Vision (pp. 1-18). Cham: Springer Nature Switzerland.
24. Y. Zhang, Y. Liao, S. Zhao, L. Weng, W. Wang, D. Lin, and Y. Li, "Architectural Layout Generation via Vision-Language Modeling," arXiv preprint arXiv:2311.15941, 2023.

25. X. Ma, X. Liu, Y. Zhu, S. Savarese, Z. Lin, L. Fei-Fei, and S. Tulsiani, "SceneDesigner: Text-to-3D Scene Generation with Layout Optimization," arXiv preprint arXiv:2411.12279, 2024.
26. GESSTALT Team, "GESSTALT Floorplan Research," 2024. [Online]. Available: https://gesstalt.github.io/research/floorplan.html
27. Papers With Code, "Text-to-3D House Model Dataset," 2023. [Online]. Available: https://paperswithcode.com/dataset/text-to-3d-house-model
28. Zimhe, "Pseudo-Floor-Plan-12k," 2023. [Online]. Available: https://huggingface.co/datasets/zimhe/pseudo-floor-plan-12k