Future Designer - generative AI meets interior design

Filip Nowicki[0009−0002−8415−8771] (✉), Arkadiusz Charliński[0009−0008−9179−7348] and Andrzej Wójtowicz[0000−0003−1385−6572]

Adam Mickiewicz University, Faculty of Mathematics and Computer Science, Poznań, Poland {filnow3@st.amu.edu.pl}

Abstract. Interior design visualization plays a crucial role in property evaluation, yet traditional methods, such as professional consultations and 3D modeling, are often time-consuming and costly. This paper introduces an AI-driven system that enables room customization in real time through automated image generation and intelligent furniture search. Our solution democratizes design visualization by reducing dependence on professional services, empowering users to instantly tailor spaces to their preferences during property comparisons. This approach enhances decision-making efficiency while offering scalable personalization, redefining how individuals engage with potential living environments. Importantly, the implementation relies entirely on open-source models and is optimized for consumer-class GPU execution. The system architecture also supports facile scaling, making it suitable for companies seeking to integrate a customizable interior design feature, potentially linked to their specific furniture catalogues.

Keywords: diffusion · interior design · vision language model

1 Introduction

Visualizing potential living spaces is crucial when choosing or redesigning a home. AI-powered commercial tools like Interior AI[1], Collov[2], and ApplyDesign[3] simplify this with virtual staging and style application. However, these often proprietary, subscription-based services can limit accessibility, customization, and integration with specific furniture catalogues. To address these gaps, we introduce a novel AI system built on open-source foundations. Our solution offers near real-time room customization, integrating automated image generation with furniture search. The key contribution is an accessible, flexible alternative; by using open-source models optimized for consumer GPUs[4], we lower the entry barrier compared to traditional methods and commercial AI platforms.

[1] https://interiorai.com/
[2] https://collov.ai/
[3] https://www.applydesign.io/
[4] Refers to GPUs with at least 24GB of Video RAM (VRAM), e.g., NVIDIA GeForce RTX 4090 or RTX 3090.

2 Furniture Search

A critical limitation of many AI interior design tools is the gap between visualizations and tangible products, hindering practical implementation. Our system's core innovation bridges this by enabling users to search for similar, potentially purchasable furniture based on items selected within the visualization.

2.1 Furniture Localization and Isolation

The initial stage of the furniture search pipeline involves accurately identifying and isolating target furniture items within the input image. This is accomplished through a two-step process:

Object Detection: We employ Grounding-DINO [6] to object detection, specifically localizing instances belonging to predefined furniture categories relevant to interior design (e.g., bed, table, sofa, chair). This step yields bounding boxes around detected furniture items.

Semantic Segmentation: For each detected bounding box, we utilize the Segment Anything Model 2 [7] to generate a precise instance segmentation mask of furniture. The generated segmentation mask is inverted to make background white and isolate only the furniture.

2.2 Structured Attribute Extraction via Vision-Language Model

Following isolation, the system derives a structured semantic description of the furniture item. For this task, we use the Qwen-2-VL 2B model [8], fine-tuned with a custom LoRA [9]. This fine-tuning was performed on our synthetically generated dataset[5] comprising diverse furniture images paired with structured descriptions.

The fine-tuned model is prompted to generate a caption for the isolated furniture image, constrained to a specific JSON format. This format enforces the extraction of key descriptive attributes: type, style, color, material, details, and room type. Our selection of the Qwen-VL model and the LoRA fine-tuning approach was based on comparative experiments involving several VLMs in the 2B parameter range. The fine-tuned Qwen model demonstrated superior performance in consistently generating valid JSON outputs and achieved higher CLIP-Score metrics [10].

Notably, the LoRA fine-tuning proved essential for constraining the verbosity of the details field, which was challenging to control via prompt engineering alone in baseline models. Furthermore, this approach facilitated the normalization of attribute values, such as mapping diverse color descriptions (e.g., "jet black," "charcoal," "ebony") to a standardized term (e.g., "black"), thereby improving consistency for user interaction.

[5] https://huggingface.co/datasets/filnow/furniture-synthetic-dataset

2.3 Attribute-Driven Search and Integration

Once generated, the structured JSON caption enables the following interactive functionalities:

- **Attribute Refinement**: The UI allows for direct refinement of attributes (e.g., changing color). These modifications then guide a diffusion model-based inpainting pipeline to regenerate the furniture item accordingly.
- **Similarity Search**: Based on the current attributes, a similarity search can be initiated. This triggers a semantic query within a vector database populated with text embeddings derived from the JSON captions to retrieve comparable items.

Retrieved items can be swapped in to replace the original furniture, or added to empty space.

This closed-loop system, connecting visual identification, structured semantic description, attribute-based refinement, and text-based vector search, forms the foundation for practical application. For enterprise use, the vector database can be populated with embeddings corresponding to a real-world furniture catalogue. Each entry can link its JSON description to actual product images, details, and purchasing URLs. This transforms the system from a purely visualization tool into an interactive visual search engine, enabling users to discover and potentially purchase real items that closely match their visualized preferences directly from the interface.

3 Image Generation

Our system at its core is using the Stable Diffusion XL [1] model, fine-tuned for realistic images called RealVisXL V5.0[6], renowned for producing photorealistic interior scenes efficiently, even on standard consumer hardware. To maintain structural integrity and spatial coherence during image manipulation, we integrate a unified ControlNet [2]. This ensures that the fundamental layout of a room remains intact while allowing for stylistic and content modifications. For near real-time user interaction, we employ Trajectory Consistency Distillation [3], which accelerates the diffusion sampling process.

The system supports a versatile suite of distinct image generation pipelines tailored to specific user interactions:

1. **Furniture Modification:** Utilizes inpainting guided by ControlNet to alter attributes (e.g., color, material) of existing furniture items.
2. **Furniture Addition:** Employs outpainting techniques to seamlessly introduce new furniture elements into designated empty spaces.
3. **Style-Consistent Furniture Replacement:** Leverages an Image Prompt Adapter [4] to replace selected furniture with similar one.

[6] https://huggingface.co/SG161222/RealVisXL-V5.0

4. **Object Removal:** Implements a tile-based ControlNet strategy for cleanly removing unwanted furniture or objects from the scene.
5. **Global Style Transformation:** Enables comprehensive changes to the room's overall aesthetic (e.g., transforming to 'Scandinavian' or 'Modern' style) using a depth-conditioned ControlNet. The requisite depth map is made using DepthAnything V2 model [5].

4 Conclusion

In conclusion, we presented an open-source AI system that leverages deep learning on consumer GPUs for real-time interior design visualization and customization. **For individuals** Like homebuyers and tenants, it democratizes design by providing an accessible, cost-effective tool for instant personalization that connects virtual ideas to tangible item characteristics. **For businesses** Such as retailers or real estate platforms, its open architecture allows flexible integration and customization, enabling engaging visual experiences potentially linked directly to their product catalogs. Project source code and documentation are available on GitHub[7] and an introductory video can be viewed at https://youtu.be/NOlGHFNzzrM

5 Appendix

To extract furniture attributes, the vision-language model is prompted using a multi-turn conversational format that ensures structured and consistent outputs in JSON. Firstly, the system prompt sets a clear expectation by emphasizing the following schema:

```
You are a furniture expert. Analyze images and provide
descriptions in this exact JSON format:
{"type": "<must be one of: bed, chair, table, sofa>",
"style": "<describe overall style>", "color": "<describe
main color>", "material": "<describe primary material>",
"shape": "<describe general shape>", "details": "<describe
one decorative feature>", "room_type": "<specify room
type>", "price_range": "<specify price range>"}
Focus on maintaining this exact structure while providing
relevant descriptions.
```

To confirm comprehension, we used the assistant prompt:

```
I will analyze the image and respond with a valid JSON object
following the exact schema.
```

[7] https://github.com/future-d3signer/future-designer-api

After that, a multi-modal input, combining the image with a textual prompt, is added:

```
Describe this furniture piece in JSON format.
```

This carefully designed prompt sequence guides the model to generate consistent, high-quality attribute descriptions in JSON (see Section 2.2).

The generation prompts for tasks such as furniture modification or addition (see Section 3) are systematically constructed. Initially, a descriptive prompt is formed based on the structured attributes extracted by the VLM. For instance, this might take the form:

```
A ${color} ${type}, made of ${material}, with a ${style} style,
featuring ${details}, suitable for a ${room_type}.
```
[8]

This base prompt is then augmented with an enhancement string, *"masterpiece, professional lighting, realistic materials, highly detailed"*, to improve image quality. Concurrently, a negative prompt, *"deformed, low quality, blurry, noise, grainy, duplicate, watermark, text, out of frame"*, is employed to mitigate common generation artifacts.

References

1. Podell et al. "Sdxl: Improving latent diffusion models for high-resolution image synthesis." The Twelfth International Conference on Learning Representations (2023).
2. Zhao et al. "Uni-controlnet: All-in-one control to text-to-image diffusion models." Advances in Neural Information Processing Systems 36 (2023): 11127-11150..0
3. Zheng et al. "Trajectory Consistency Distillation: Improved Latent Consistency Distillation by Semi-Linear Consistency Function with Trajectory Mapping." arXiv preprint arXiv:2402.19159 (2024).
4. Ye et al. "Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models." arXiv preprint arXiv:2308.06721 (2023).
5. Yang, Lihe, et al. "Depth anything v2." Advances in Neural Information Processing Systems 37 (2024): 21875-21911.
6. Liu et al. "Grounding dino: Marrying dino with grounded pre-training for open-set object detection." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024.
7. Ravi et al. "Sam 2: Segment anything in images and videos." The Thirteenth International Conference on Learning Representations (2024).
8. Wang et al. "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution." arXiv preprint arXiv:2409.12191 (2024).
9. Hu et al. "Lora: Low-rank adaptation of large language models." International Conference on Learning Representations (2022).
10. Hessel et al. "Clipscore: A reference-free evaluation metric for image captioning." Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021) 7514-7528.

[8] Placeholders like ${color} are dynamically replaced with the actual attribute values (e.g., "blue," "red") from the JSON caption.