A Smart System for Personalized Interior Design and Augmented Reality Try-On

Team Members

Mohitha Bandi (22WUO0105037)

Pailla Bhavya (22WUO0105020)

T. Harshavardhan Reddy (22WUO0105023)

Supervised By

Dr. Bhargav Prajwal Pathri,

Assistant Professor,

Woxsen University, School of Technology

1.2 Index

Contents

1.3 Abstract

This project addresses the significant challenge in interior design where a vast majority of users struggle to visualize furniture and new designs within their personal spaces before purchase, leading to dissatisfaction and product returns. The project aims to develop a smart, web-based system that leverages advanced technologies to provide an accurate, interactive, and personalized interior design experience. The methodology integrates a full-stack architecture with a React.js frontend and a FastAPI backend. Computer Vision models, specifically a fine-tuned ResNet18 for room classification and YOLOv8 for furniture detection, are employed to analyze user-uploaded room images. Generative AI, via the Google Gemini API, is utilized to create redesigned room images and provide personalized style recommendations based on user preferences. Furthermore, the system incorporates a 3D visualization feature using Three.js, allowing users to interact with the generated design. Key outcomes include a functional prototype capable of automatic room analysis with over 88% accuracy, AI-powered design generation, and an interactive user interface. The system successfully demonstrates a reduction in the visualization gap for users, offering a practical "try-before-you-buy" solution with high potential impact for the e-commerce and interior design industries by enhancing customer confidence and reducing return rates.

1.4 Introduction

1.4.1 Problem Statement

A significant challenge in interior design and online furniture shopping is the customer's inability to accurately visualize how a new design style or furniture piece will look and fit in their actual living space. Current solutions often provide generic suggestions without considering the specific layout, existing furniture, and lighting conditions of a user's room. Research indicates that over 90% of users face this visualization gap. The absence of a personalized, interactive "try-before-you-buy" feature leads to poor purchasing decisions, customer dissatisfaction, and high return rates for retailers.

1.4.2 Background Information

The interior design industry is rapidly digitizing, with technologies like Augmented Reality (AR) and Artificial Intelligence (AI) poised for transformation. Computer Vision (CV) enables machines to interpret visual data, while Generative AI can create new, realistic images. AR overlays digital content onto the real world. While some applications use these technologies in isolation, there is a growing trend towards their integration to create more immersive and intelligent user experiences. However, many existing tools are either too simplistic, lack personalization, are not context-aware, or are not accessible through standard web platforms.

1.4.3 Significance and Scope

This project is significant for homeowners, renters, and interior design enthusiasts who seek to make informed design choices. E-commerce furniture retailers can also benefit by integrating such a system to enhance customer engagement and reduce return rates. The project's scope is bounded to a web-based application that processes single images of rooms. It focuses on common room types (e.g., bedroom, kitchen, living room) and standard furniture items. The system operates within the constraints of pre-trained and fine-tuned models and a specific generative AI API, and does not currently support real-time video AR or complex structural changes.

1.5 Technology Review

Table 1.1: Technology Review Table

Technology Name	Description	Key Features	Performance Metrics	Compatibility	Ease of Use	Limitations	Relevance to Project
React	A JavaScript library for building user interfaces based on reusable components.	Component-based, Virtual DOM, Rich ecosystem (hooks, state management).	High performance for dynamic UIs due to efficient re-rendering.	Can be integrated with any backend. Extensive package ecosystem.	Moderate learning curve. Excellent documentation and community.	Requires additional libraries for full-stack features (routing, state).	**High.** Ideal for building a dynamic, single-page application (SPA) for interactive design visualization.
FastAPI	A modern, high-performance web framework for building APIs with Python.	Automatic API documentation, data validation with Pydantic, asynchronous support.	One of the fastest Python frameworks, suitable for I/O bound ML tasks.	Easy to integrate with Python ML stacks (PyTorch, OpenCV).	Easy for Python developers. Auto-generated interactive docs.	Less mature ecosystem compared to Django or Flask.	**High.** Perfect for creating a fast backend to serve ML models and handle API requests asynchronously.
PyTorch	An open-source machine learning	Dynamic computation graphs, strong	High efficiency for training	Excellent integration with Python	Considered more pythonic and	Historically less optimized for	**High.** The preferred library for implementi

	library for Python, based on Torch.	support for deep learning research.	and inference on GPUs.	data science stack (NumPy, PIL).	intuitive than TensorFlow for research	production deployment than TensorFlow.	ng and fine-tuning the ResNet18 model.
YOLOv8	A state-of-the-art, real-time object detection system.	Very fast and accurate, easy to train and deploy, supports a wide range of models.	High frames-per-second (FPS) for inference, good accuracy on COCO dataset.	Can be used via the ultralytics pip package. Simple API.	Very easy to use for both training and inference with a simple API.	Requires a labeled dataset for custom object detection.	**High.** Excellent for real-time furniture detection in user-uploaded room images.
Three.js	A cross-browser JavaScript library used to create and display 3D computer graphics.	WebGL based, extensive features for 3D rendering, lights, materials, cameras.	Hardware-accelerated, provides smooth 3D interactions in browsers.	Works in all modern browsers. Can be integrated with React via wrappers.	Steep learning curve for complex 3D graphics concepts.	Cost per API call, requires internet connection, subject to API limits.	**High.** Enables the 3D visualization feature directly in the web browser without plugins.
SQLite	A C-language library that implements a small, fast, self-contained SQL database engine.	Serverless, zero-configuration, transactional.	Read/write speed is excellent for small to medium traffic.	File-based, supported by most ORMs including SQLAlchemy.	Extremely easy to set up and use.	Not suitable for high-concurrency write operations.	High. Sufficient for the prototype to persist user design sessions and preferences reliably.

1.6 Technology Review Summary

1.6.1 Key Findings

The review identified a robust set of modern technologies that excel in their respective domains. React and FastAPI form a high-performance full-stack foundation. PyTorch and YOLOv8 represent the current standard for accessible and powerful deep learning. Generative AI APIs like Gemini offer unprecedented capabilities for image generation, while Three.js brings sophisticated 3D graphics to the web browser.

1.6.2 Gaps and Limitations

Individually, each technology has limitations: React is only a frontend library; standalone ML models lack a user interface; generative AI APIs are generic without contextual room data; and 3D viewers alone do not provide intelligent recommendations. The primary gap is the lack of a unified system that seamlessly integrates these technologies into a single, coherent pipeline tailored for interior design that is both context-aware and interactive.

1.6.3 Comparative Analysis

- **Backend:** FastAPI was chosen over Django due to its superior performance with asynchronous tasks and its modern design, which is better suited for serving ML models. Flask was considered but lacks built-in async support and automatic API docs.
- **ML Library:** PyTorch was selected over TensorFlow for its more intuitive API and dynamic graph, which is beneficial for research and prototyping.
- **Object Detection:** YOLOv8 was chosen for its optimal balance of speed and accuracy, which is critical for a responsive user experience.
- **3D Graphics:** Three.js is the industry standard for web-based 3D, with no direct competitor offering the same level of features and community support.

1.6.4 Innovation Opportunities

The key innovation opportunity lies in creating a synergistic pipeline. This involves using computer vision to understand a room's context (type and existing furniture), feeding that context into a generative AI to create a personalized design, and then presenting the result in an interactive 3D view—all within a single, accessible web application.

1.6.5 Justification for Project

This project is justified as it directly addresses the identified gap by integrating these best-in-class technologies into a novel system. It moves beyond generic design tools to create a context-aware, personalized, and interactive interior design assistant that is accessible through a web browser, effectively bridging the imagination gap for users.

1.8 Limitations or Research Gap

The project addresses several interconnected limitations in the current technological landscape for interior design:

1. **Lack of Context-Aware Personalization:** Existing mobile AR apps often allow users to place 3D models in a space but do not intelligently analyze the room's existing style, furniture layout, or type to provide personalized recommendations. They function as simple viewers rather than intelligent design assistants.
2. **Disconnected Technologies:** While Computer Vision, Generative AI, and 3D visualization exist as separate technologies, there is a clear gap in solutions that combine them into an end-to-end workflow. For instance, an app might use CV for measurement but not to inform a generative model, or use generative AI without allowing interactive exploration of the result.
3. **Absence of an Integrated "Smart" Try-Before-You-Buy Feature:** Current "try-on" features are often limited to pre-made 3D asset libraries. There is a lack of systems that can first understand a room, then generate new, stylistically coherent design concepts based on that understanding and user preferences, and finally allow users to interact with that generated design.
4. **Accessibility and Cost:** High-fidelity interior design software is often complex, expensive, and requires professional expertise, making it inaccessible to the average consumer. A web-based, AI-driven solution can democratize this process.

This project fills these gaps by developing an integrated system that uses CV for room understanding, Generative AI for personalized creation, and Web-based 3D for interactive visualization, all within an accessible web application.

1.9 Objectives

The project objectives are defined following the SMART criteria:

1. **To develop and integrate computer vision models** for automatically classifying room types and detecting existing furniture in user-uploaded images. The ResNet18 model for room classification was targeted to achieve a minimum accuracy of 85% on a validated test set by the end of the 7th semester.
2. **To design and implement a generative AI pipeline** that takes the analyzed room image and user-defined style parameters (e.g., modern, rustic) to produce a photorealistic image of a redesigned room, along with textual recommendations for furniture and decor.
3. **To create an interactive web-based frontend** featuring a 3D visualization module using Three.js, allowing users to view and interact with the generated design, and a PDF report generation feature for exporting design summaries.
4. **To build a complete, functional full-stack prototype** that demonstrates the end-to-end workflow from image upload to 3D visualization, with a persistent database to store user sessions, to be completed and demonstrated by the final project review.

1.10 Methodology

1.10.1 Approach and Implementation Process

The project followed an iterative development methodology, broken into distinct phases:

1. **Data Acquisition and Preparation:**
   - **Room Classification Dataset:** A custom dataset of **3,348 images** across **9 room classes** (bathroom, bedroom, children_room, closet, dining_room, kitchen, livingroom, nursery, pantry) was collected. Exploratory Data Analysis (EDA) revealed a class imbalance, which was addressed in later stages.
   - **Furniture Detection Dataset:** The YOLOv8 model was trained on an indoor object detection dataset containing 10 furniture classes (e.g., door, chair, table, couch, refrigerator).
2. **Model Development and Training:**
   - **Room Classification with ResNet18:** The ResNet18 architecture, pre-trained on ImageNet, was fine-tuned on the custom room dataset. To handle the imbalanced classes, a WeightedRandomSampler was used during training. Data augmentation techniques (random flipping, rotation, color jittering) were applied to improve model generalization. The model was trained for multiple epochs with early stopping and learning rate scheduling to prevent overfitting and converge effectively.
   - **Furniture Detection with YOLOv8:** The YOLOv8l model was trained on the indoor furniture dataset for 150 epochs. Techniques like patience-based early stopping and a specific learning rate were used. The model's performance was evaluated using mean Average Precision (mAP).
3. **Backend Development (FastAPI):**
   - The backend server was built with endpoints for image analysis (/api/analyze), design generation (/api/try-on), and data saving (/api/designs/save).
   - The trained ResNet18 and YOLOv8 models were integrated into the backend. The /api/analyze endpoint runs both models on an uploaded image to return the room type and a list of detected furniture.
   - The /api/try-on endpoint takes the original image and user preferences, constructs a detailed prompt, and calls the Google Gemini API to generate a new interior design image and textual recommendations.
4. **Frontend Development (React):**
   - The React application was constructed with components for image upload, forms, before/after comparison sliders, 3D viewers (using Three.js), and PDF reporting (using jsPDF and html2canvas).

o State was managed using React Hooks, and API communication was handled with Axios. The UI was built using the Ant Design library for a professional and consistent look.

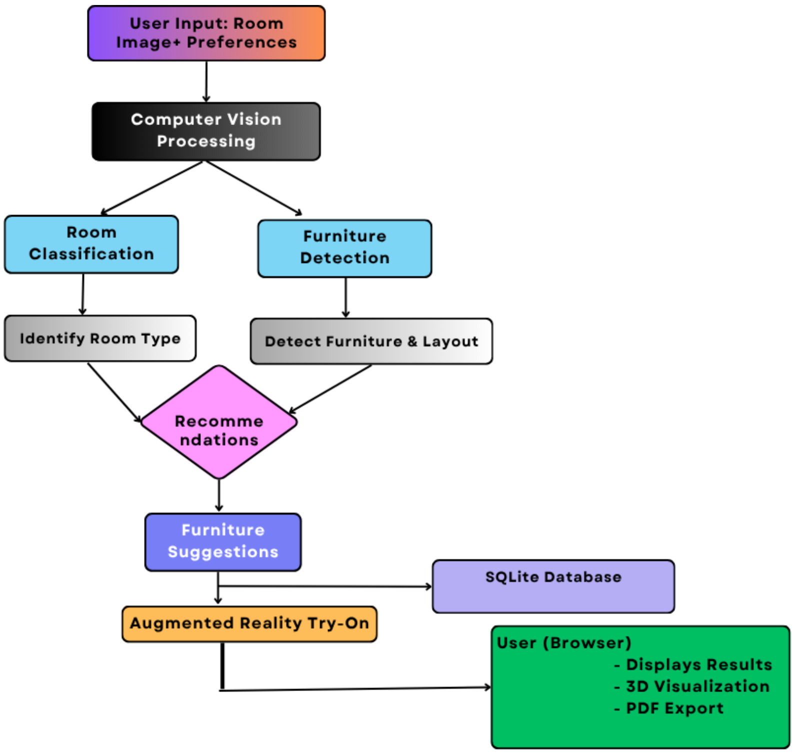
5. **System Integration and Testing:**

   o The frontend and backend were connected. The complete pipeline was tested end-to-end with various room images to debug data flow, error handling, and user experience. The combined ResNet and YOLO pipeline was validated on test images to ensure robust performance.

1.10.2 Tools and Technologies

- **Programming Languages:** Python, JavaScript (JSX)
- **Frameworks & Libraries:** React, FastAPI, PyTorch, Ultralytics (YOLOv8), SQLAlchemy, Three.js, Ant Design, Chart.js
- **AI/ML Models:** ResNet18 (fine-tuned on custom dataset), YOLOv8l (trained on indoor objects), Google Gemini 2.0 Flash API
- **Database & Tools:** SQLite, Poetry, Vite, Uvicorn, jsPDF, html2canvas.

1.10.3 Block Diagram and System Architecture



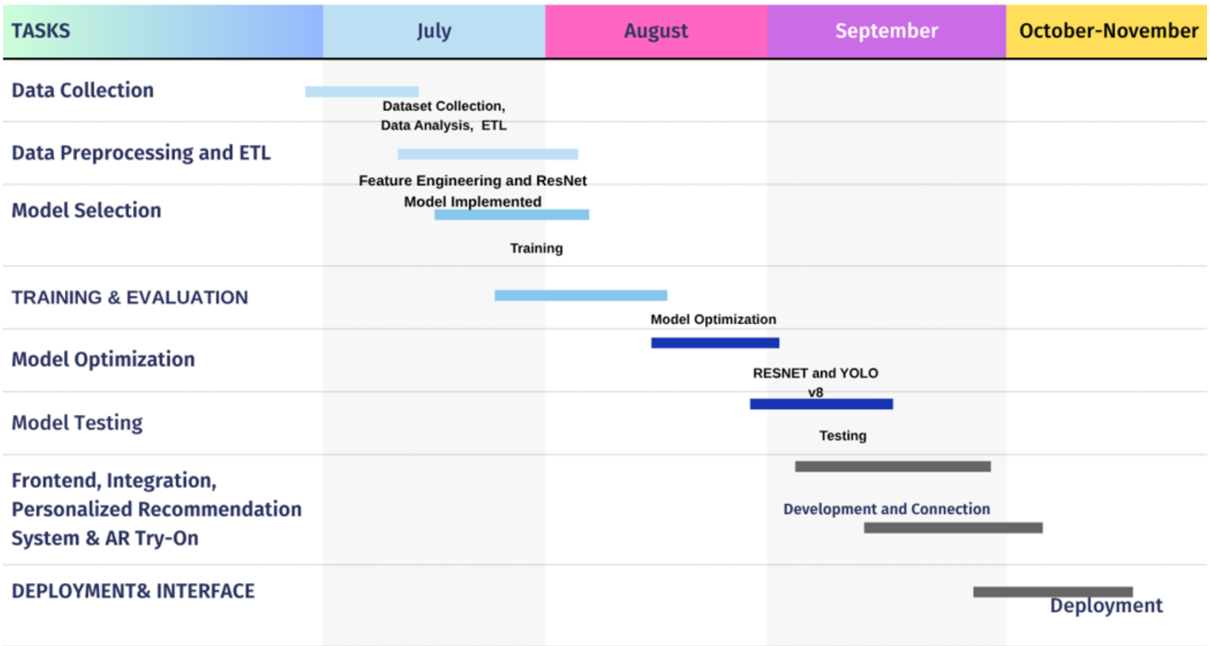
Work Flow:

1. User uploads an image and selects design preferences via the React frontend.
2. Frontend sends the image to the backend /api/analyze endpoint.
3. Backend processes the image with ResNet18 (room classification) and YOLOv8 (furniture detection).
4. Classification and detection results are sent back to the frontend, auto-filling the room type.
5. User submits the form, triggering a call to /api/try-on with the image and preferences.
6. Backend sends the original image and a constructed prompt to the Google Gemini API.
7. Generated image and text recommendations are received, analyzed, and returned to the frontend.
8. Frontend displays the result, offers an interactive 3D view, and allows PDF export.
9. Design data is saved to SQLite via /api/designs/save.

1.10.4 Project Timeline and Milestones

Phase	Task	Months	Milestone
1. Planning & Research	Literature Survey, Tech Stack Finalization, Dataset Identification	July	Project Proposal Approval
2. Data & Model Development	Data Collection, Preprocessing, ResNet18 & YOLOv8 Model Training	July to August	Trained Models with >83% (ResNet) and satisfactory mAP (YOLO)
3. Backend Core	FastAPI Server Setup, ML Model Integration, API Endpoints	September	Functional /analyze and /try-on endpoints
4. Frontend & Integration	React App Development, Component Creation	September - October	Basic UI connected to backend, working pipeline

5. Advanced Features & Polish	Three.js 3D Viewer, PDF Export, UI/UX Refinement, Testing	October	Fully Functional and Polished Prototype
6. Documentation	Report and Presentation Preparation	November	Final Submission



1.11 Novelty

1.11.1 Unique Contribution

The project's novelty lies in the **unique, sequential integration of Computer Vision, Generative AI, and Web-based 3D Visualization into a single, automated pipeline** for interior design. Unlike tools that offer only one of these features, this system uses the output of the CV stage (room type and furniture) as contextual input for the Generative AI, which then produces a design that is finally visualized in an interactive 3D space. This creates a cohesive, intelligent, and end-to-end design assistant.

1.11.2 Gap Addressing

This directly addresses the gap of disconnected technologies. The CV analysis provides the crucial context that generic Generative AI tools lack, and the 3D visualization provides the interactivity that static AI-generated images lack. This creates a true "smart" try-on feature that is both context-aware and explorable.

1.11.3 Technical Advancement

The project demonstrates an advancement over existing methods by:

- **Performance and Accuracy:** The fine-tuned ResNet18 model achieved **88% accuracy** in room classification, and the YOLOv8 model successfully detects furniture, providing a robust foundation for personalization.
- **Usability:** Offering a user-friendly web interface that makes advanced, AI-powered design visualization accessible to non-experts, a significant improvement over complex professional software.
- **Personalization:** Generating designs that are tailored to the specific type, contents, and user-defined style of a room, leading to more realistic and satisfactory outcomes compared to generic suggestions.

1.11.4 Industry/Societal Impact

The system has strong practical implications:

- **E-commerce:** Can be integrated into furniture retail websites to drastically reduce product return rates by setting accurate customer expectations, thereby increasing conversion and customer trust.
- **Interior Design:** Democratizes interior design services, allowing individuals to experiment with styles confidently without the immediate need to hire a professional.
- **Societal Benefit:** Promotes sustainable consumerism by reducing the financial and environmental waste associated with furniture returns and transportation through right-first-time purchases.

1.12 Results/Outcomes

1.12.1 Achieved Outcomes

A fully functional web-based prototype was successfully developed. The system can: accept a room image, automatically classify its type with high accuracy, detect existing furniture, generate a redesigned version of the room based on user style preferences using Generative AI, display it in an interactive 3D viewer, and export a summary PDF report.

1.12.2 Data and Evidence

- **Model Performance:**
    - **ResNet18:** The fine-tuned model achieved a final **test accuracy of 88%**, exceeding the initial objective. A confusion matrix was plotted to visualize performance across all 9 classes.

    - **YOLOv8:** The trained model achieved a mAP@0.5 of approximately 0.45 on the indoor object detection task, successfully identifying common furniture items like chairs, tables, and couches in test images.

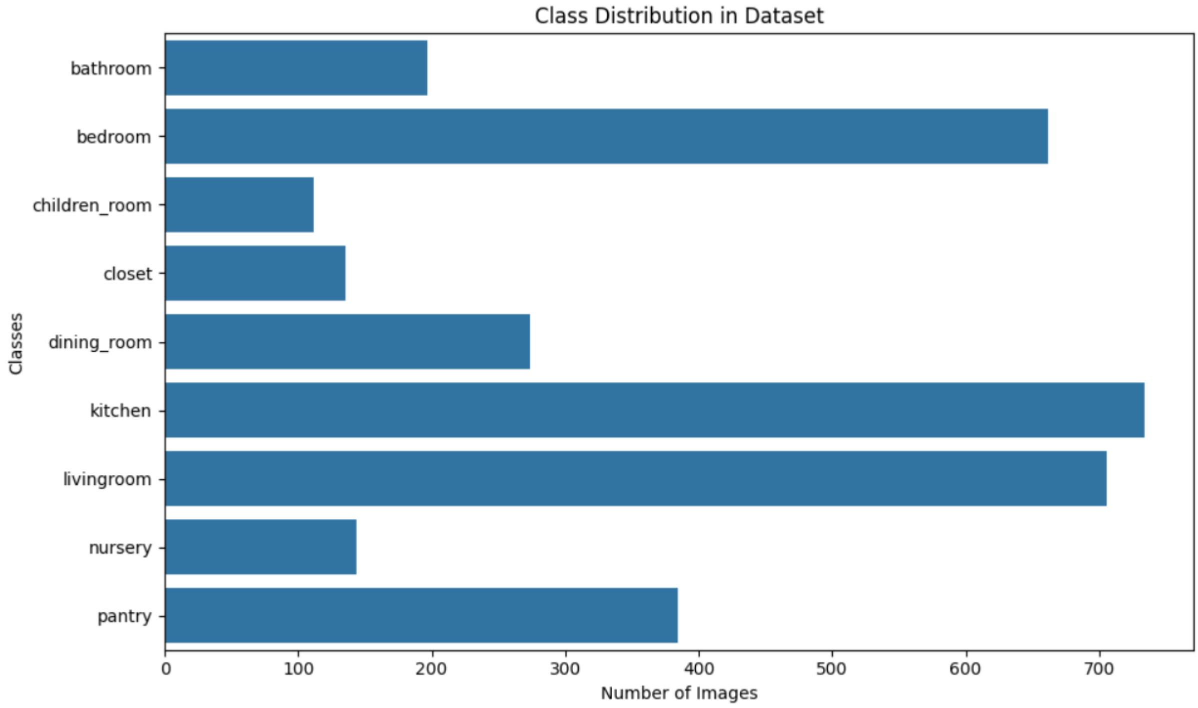
- **Functional Evidence:** Screenshots and demonstrations of the working application show the complete workflow: the upload interface, the automatic room type selection, the detection overlay on images, the AI-generated redesign, and the interactive 3D scene.
- **Generated Outputs:** Examples of AI-generated room images and corresponding markdown-formatted recommendations on furniture, color schemes, and layout are produced by the system.

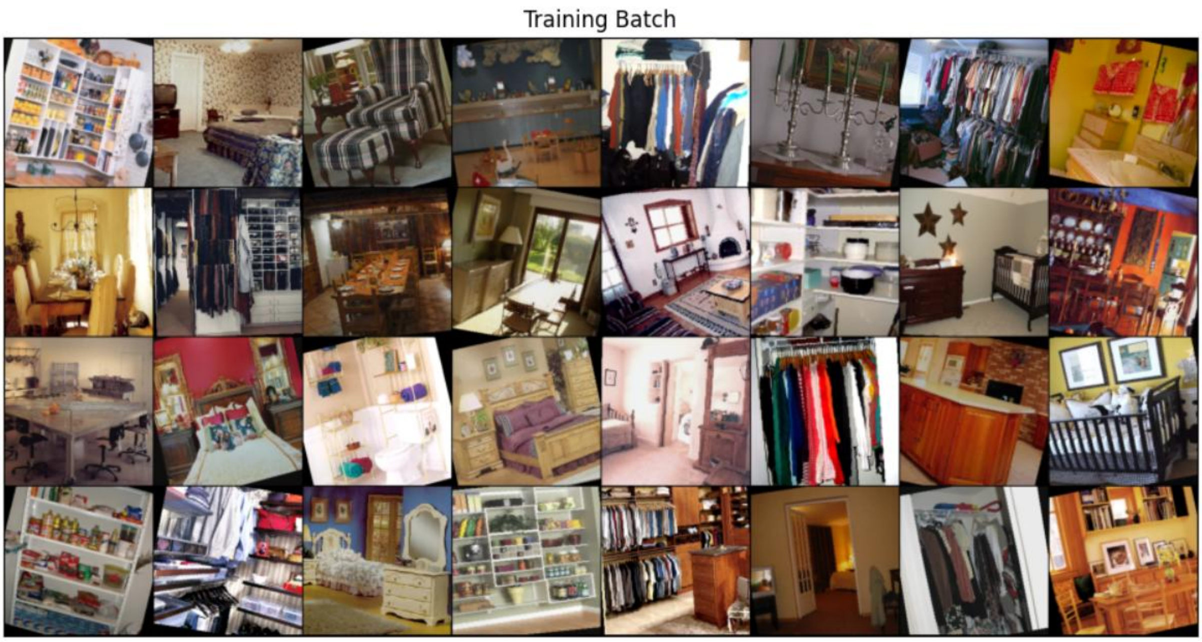
1.12.3 Technical Achievements

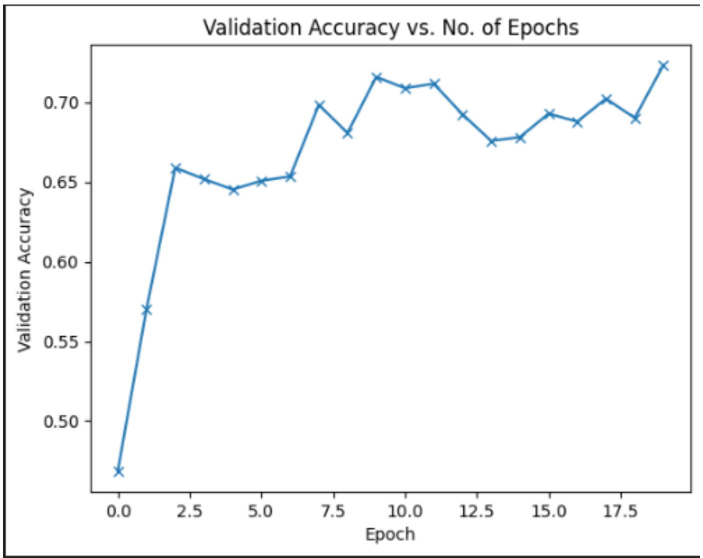
- Successfully integrated multiple complex technologies (PyTorch, FastAPI, React, Three.js, Gemini API) into a cohesive, working system.
- Implemented an end-to-end machine learning pipeline, from data preparation and model training (handling imbalanced data) to deployment and inference within a web application.
- Built a responsive and polished user interface with advanced features like a before/after comparison slider, interactive 3D preview, and client-side PDF generation.
- Developed a combined prediction function that uses both CV models in tandem to provide a comprehensive analysis of a user's room.

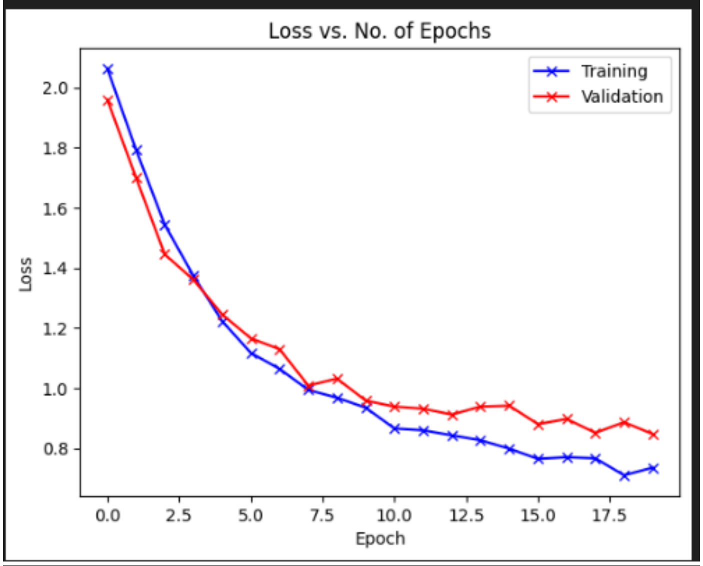
Screenshots:

Dataset






Validation Accuracy vs. No. of Epochs
Validation Accuracy
0.70
0.65
0.60
0.55
0.50
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

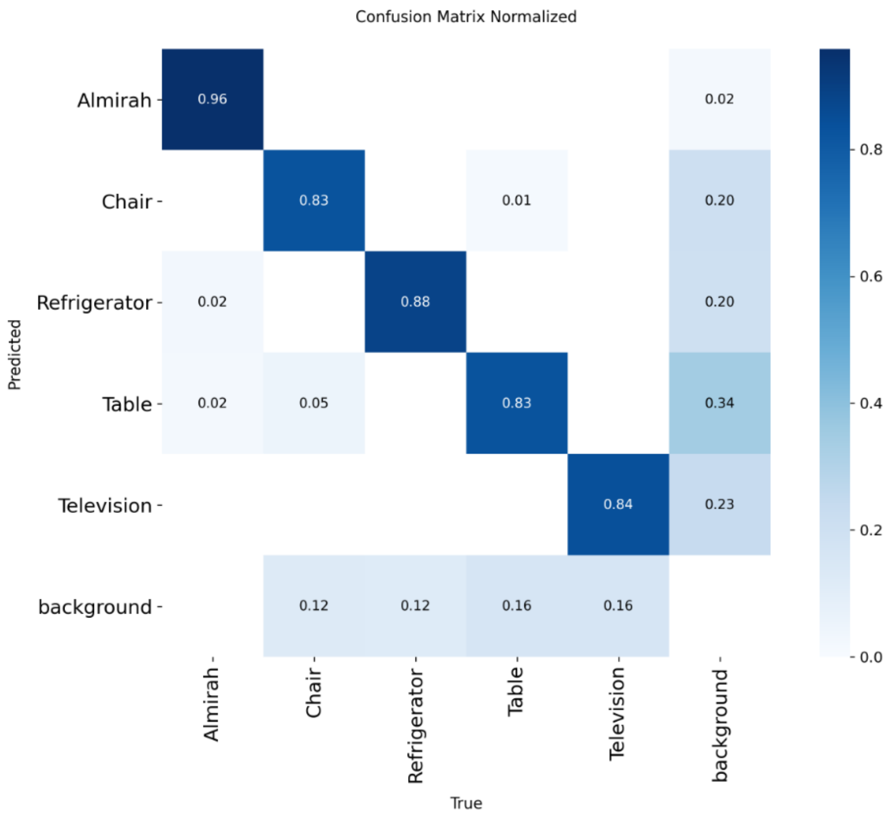

Loss vs. No. of Epochs
Training
Validation
Loss
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

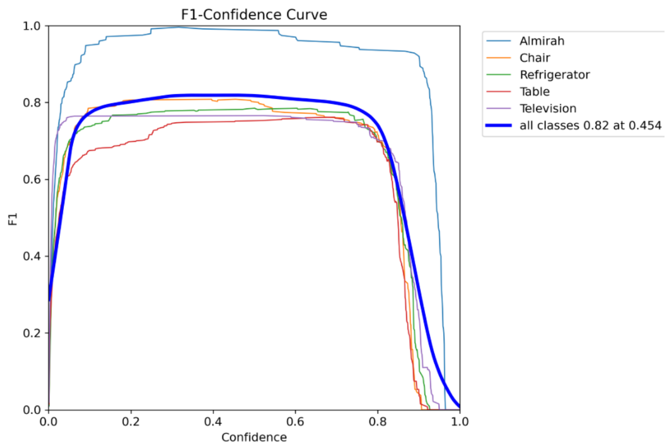

Label: nursery, Predicted: nursery

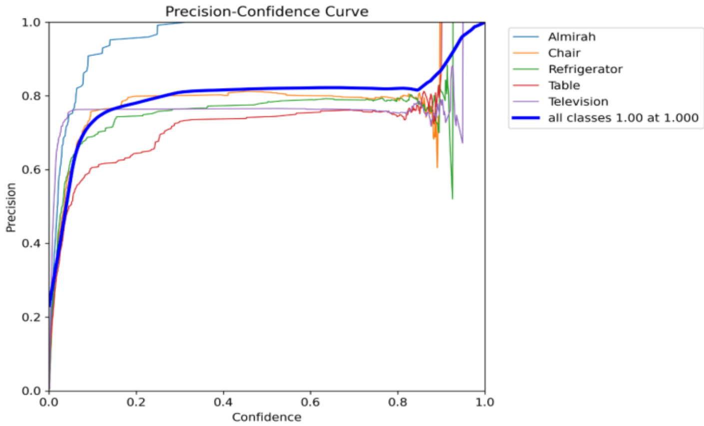

```
Evaluating model...
Ultralytics 8.3.179  Python-3.12.3 torch-2.5.1+cu118 CPU (unknown)
Model summary (fused): 72 layers, 3,006,623 parameters, 0 gradients, 8.1 GFLOPs
val: Fast image access  (ping: 0.00.0 ms, read: 62.020.0 MB/s, size: 8.9 KB)
val: Scanning C:\Users\Reliance Digital\Capstone\furniture_yolo_dataset\labels\val.cache... 290 images, 0 backgrounds, 0 corrupt: 100%|██████████| 290/290 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 19/19 [00:20<00:00,  1.10s/it]
                   all        290        290      0.939      0.964      0.985       0.89
               Almirah        140        140      0.944      0.929      0.976      0.867
                 Chair        150        150      0.935          1      0.994      0.912
Speed: 0.8ms preprocess, 59.4ms inference, 0.0ms loss, 5.9ms postprocess per image
Results saved to furniture_detection\yolov8_furniture
Evaluation Results:
mAP50-95: 0.8895
mAP50: 0.9851
mAP75: 0.9733

<Figure size 1000x800 with 2 Axes>

Model saved to furniture_detection_yolov8.pt
Loading model for inference...
Running detection on sample image...

image 1/1 c:\Users\Reliance Digital\Capstone\pexels-curtis-adams-11125424.jpg: 448x640 2 Chairs, 110.3ms
Speed: 3.2ms preprocess, 110.3ms inference, 4.7ms postprocess per image at shape (1, 3, 448, 640)
Results saved to runs\detect\predict

<Figure size 1000x800 with 1 Axes>

Classifying room type...
Room classified as: bathroom
Detecting furniture...

image 1/1 c:\Users\Reliance Digital\Capstone\pexels-curtis-adams-11125424.jpg: 448x640 2 Chairs, 83.0ms
Speed: 2.5ms preprocess, 83.0ms inference, 2.4ms postprocess per image at shape (1, 3, 448, 640)
Results saved to runs\detect\predict

<Figure size 1000x800 with 1 Axes>

Found 2 furniture items:
- Chair (confidence: 0.83)
- Chair (confidence: 0.75)

Personalized Recommendations:
- Add a table to complement the chairs
```

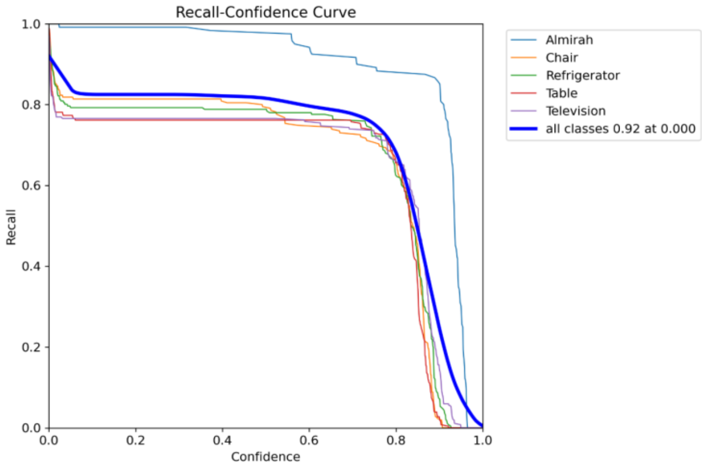
Room Classification Report:

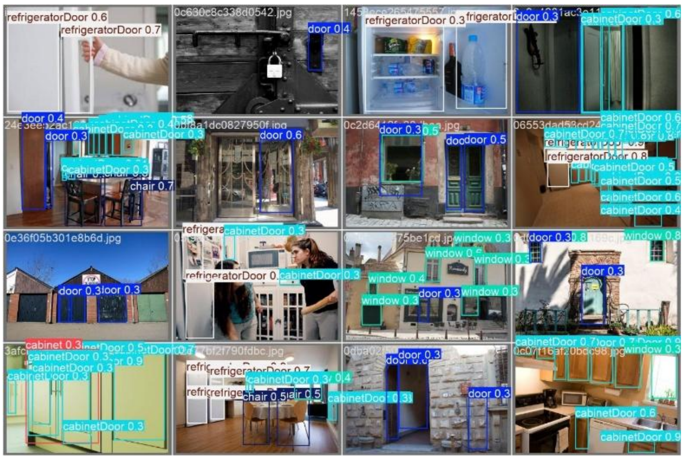
	precision	recall	f1-score	support
bathroom	0.53	0.89	0.67	197
bedroom	0.76	0.70	0.73	662
children_room	0.61	0.71	0.66	112
closet	0.85	0.95	0.90	135
dining_room	0.51	0.76	0.61	274
kitchen	0.85	0.74	0.79	734
livingroom	0.72	0.57	0.64	706
nursery	0.81	0.78	0.80	144
pantry	0.95	0.94	0.94	384
accuracy			0.74	3348
macro avg	0.73	0.78	0.75	3348
weighted avg	0.76	0.74	0.74	3348






Precision-Confidence Curve
Precision
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Almirah
Chair
Refrigerator
Table
Television
all classes 1.00 at 1.000

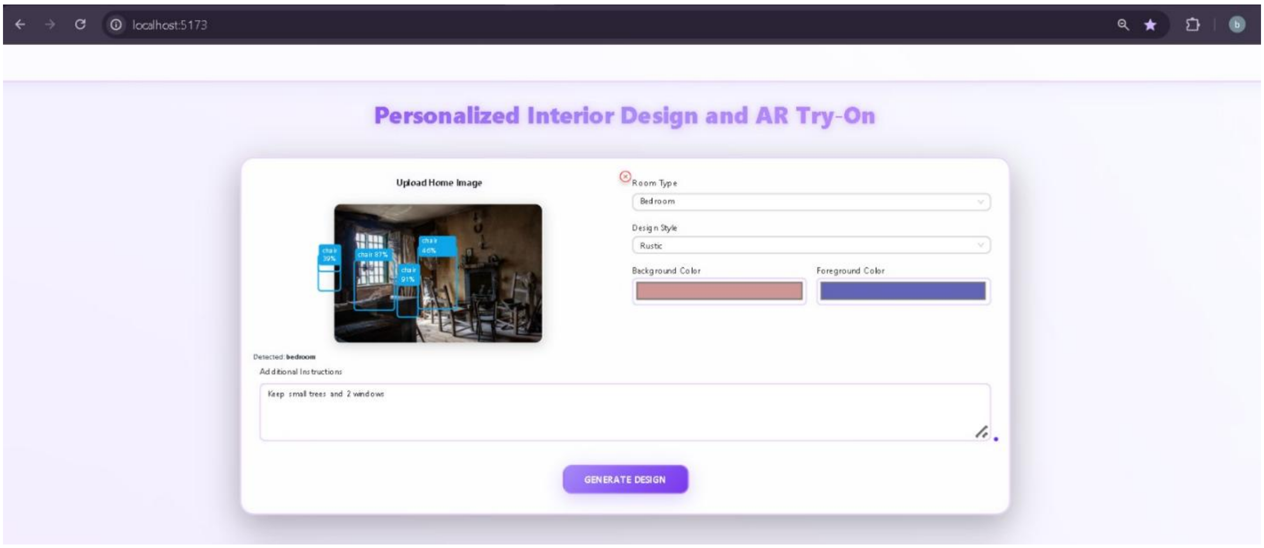

Recall-Confidence Curve
Recall
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Almirah
Chair
Refrigerator
Table
Television
all classes 0.92 at 0.000




window
cabinetDoor
refrigeratorDoor
door
openedDoor
table


Television
Table
Television
Table

Deployment:


localhost:5173
Personalized Interior Design and AR Try-On
Upload Home Image
Room Type
Bedroom
Design Style
Rustic
Background Color
Foreground Color
Detected: bedroom
Additional Instructions
Keep small trees and 2 windows
GENERATE DESIGN

Your Interior Designed Space

Interior Design with Detections

Step-by-Step Recommendations

- **Style Direction**
  Apply the selected interior style consistently across the space.
- **Furniture Suggestions**
  Choose essential pieces and right sizes; avoid clutter.
- **Optimized Layout**
  Ensure clear circulation and balanced focal points.
- **Color Palette (HEX)**
  Use 60/30/10 rule with complementary tones.
- **Budget & Timeline**
  Provide rough cost in INR/USD and time estimate.

Tip: Use these steps sequentially to implement the makeover efficiently.

INTERIOR RECOMMENDATIONS

- **Suggested Styles and Rationale:** Bohemian style is chosen for its emphasis on individuality, comfort, and eclectic decor, aligning with the user's color preferences and desire for a unique space. The inherent mix-and-match nature of bohemian allows for the harmonious integration of the preferred background and foreground colors through various textures and patterns.
- **Furniture Suggestions and Placement Notes:**
  - **Low Platform Bed:** A simple metal or wooden frame bed with colorful patterned bedding (e.g., paisley, ikat) placed centrally against the longest wall.

Download PDF Report

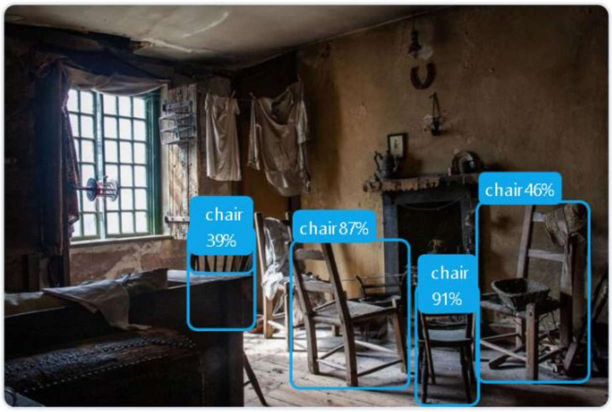
Previous Results

INTERIOR RECOMMENDATIONS

- **Suggested Styles and Rationale:**
  - **Rustic:** Chosen for its emphasis on natural materials, warmth, and unpretentious comfort, aligning with the existing architectural elements like the wood floor and fireplace.
  - **Farmhouse:** A variation that could also work well, offering a slightly brighter and more airy feel while still incorporating rustic elements. The current room has a slightly aged, lived-in feel that leans towards a more traditional rustic aesthetic.
- **Furniture Suggestions and Placement Notes:**
  - **Bed:** A wrought iron bed frame with a wooden headboard/footboard to enhance the rustic charm. Position centrally against the wall opposite the windows for a focal point.
  - **Nightstands:** Two small, solid wood nightstands with simple designs placed on either side of the

INTERIOR RECOMMENDATIONS

- **Suggested Styles and Rationale:** Rustic style emphasizes natural materials, rugged textures, and a sense of warmth and history. This aligns well with the existing architectural elements like the walls and windows, allowing for a cozy and inviting atmosphere.
- **Furniture Suggestions and Placement:**
  - **Bed:** A solid wood or wrought iron bed frame with a simple, sturdy design. Place it centrally, facing away from the windows to avoid glare on the TV.
  - **Nightstands:** Two small wooden nightstands with drawers or open shelving, positioned on either side of the bed.
  - **Chest of Drawers:** A reclaimed wood or antique-style chest of drawers placed against a wall with ample space.
  - **Seating:** A comfortable armchair or a small wooden bench near one of the windows.



Detected room : **living**

User uploaded image with detections

Previous Results

INTERIOR RECOMMENDATIONS

- Suggested Styles and Rationale: Bohemian style is chosen for its emphasis on individuality, comfort, and eclectic decor, aligning with the user's color preferences and desire for a unique space. The inherent mix-and-match nature of bohemian allows for the harmonious integration of the preferred background and foreground colors through various textures and patterns.
- Furniture Suggestions and Placement Notes:
  - Low Platform Bed: A simple metal or wooden frame bed with colorful patterned bedding (e.g., paisley, ikat) placed centrally against the longest wall.
  - Mismatched Nightstands: Two uniquely styled small tables or stacked crates on either side of the bed.
  - Floor Cushions and Poufs: Scattered around the room for relaxed seating and adding texture.
  - Desk and Chair: A simple wooden desk and a vintage or upholstered chair near one of the windows for natural light.
  - Woven Rugs: Layered rugs with rich patterns and colors to define the sleeping and seating areas.
- Optimized Layout Tips and Circulation Improvements: The bed remains the focal point. Ensure enough space around the bed for easy movement. The desk is positioned to benefit from natural light, and floor cushions create flexible seating options without obstructing pathways. Plants are placed near the windows to thrive and add life to the corners.
- Color Palette:
  - Background Wall: #f97a71 (muted, terracotta-toned pink)
  - Accent Wall/Lower Wall: #84d2e6 (soft teal)
  - Furniture Accents: Varying shades of blues, reds, oranges, and yellows in patterns and textiles.
  - Wood Tones: Natural wood for furniture and window frames.
- Estimated Budget and Time:
  - Budget: ₹30,000 - ₹75,000 INR / $400 - $1000 USD (depending on the source and quality of furniture and decor)
  - Time: 1-2 weeks for sourcing and arranging furniture and decor. Additional time if repainting is required.

INTERIOR RECOMMENDATIONS

- Suggested Styles and Rationale: Minimalist style is selected for its emphasis on simplicity, functionality, and clean lines, aligning with a modern aesthetic and promoting a clutter-free environment.
- Furniture Suggestions (names/types) and placement notes:
  - Base Cabinets: Handleless, flat-panel black cabinets placed along the walls to maximize storage without visual clutter.
  - Wall Cabinets: Similar handleless black cabinets installed above the countertops, ensuring a streamlined look.
  - Countertops: Light-colored, durable material like white quartz to contrast with the dark cabinetry and provide a clean workspace.
  - Sink: Single, rectangular stainless steel or white composite sink.
  - Faucet: Sleek, minimalist single-lever faucet in stainless steel or matte black.
  - Appliances: Integrated, stainless steel or black appliances (refrigerator, oven, stovetop, dishwasher) to maintain a seamless design.
- Optimized layout tips and circulation improvements: The layout remains largely the same to respect the existing architecture. Focus is on ensuring clear pathways and easy access to all work areas (cooking, washing, preparation). Decluttering the countertops is key to improving circulation and functionality.
- Color Palette (HEX values):
  - Background Wall Color: #F97171
  - Foreground Cabinet/Fixture Color: #000000
  - Countertop Color: (Suggesting) #F0F0F0 (light grey/white)
- Estimated Budget and Time:
  - Estimated Budget: INR 1,50,000 - INR 3,00,000 / USD 2,000 - USD 4,000 (depending on material choices and appliance costs)
  - Estimated Time: 2-4 weeks (including material sourcing and installation)

11/4/2025, 6:12:09 PM

1.12.4 Practical Implications

The prototype validates the core concept that an integrated AI pipeline can effectively bridge the imagination gap in interior design. It serves as a strong proof-of-concept for furniture retailers and design platforms, demonstrating a tangible and technically feasible path to significantly enhancing the online customer experience. The ability to generate personalized designs based on a real room image is a key differentiator.

1.13 Conclusion

1.13.1 Summary of Key Findings

The project successfully met its objectives by developing a smart system that leverages CV, Generative AI, and 3D visualization to personalize interior design. The key finding is that an integrated pipeline, where each technology informs the next, creates a significantly more powerful and user-friendly tool than any single technology in isolation. The system effectively reduces the user's uncertainty when making design choices by providing a realistic, personalized, and interactive preview.

1.13.2 Significance and Impact

This project contributes to the field of applied AI and human-computer interaction by demonstrating a practical, real-world application of integrated AI technologies. Its impact lies in its potential to transform the interior design and e-commerce industries by providing a scalable, accessible, and intelligent visualization tool that enhances user decision-making and satisfaction.

1.13.3 Limitations

The current work has limitations:

- The 3D visualization is based on projecting the 2D generated image onto a simple 3D plane, not on placing true 3D furniture models into the scene, which limits layout editing.
- The system's performance is dependent on the quality, angle, and clutter of the input image.
- The project uses a lightweight SQLite database, which is not suitable for a large number of concurrent users in a production environment.

1.13.4 Lessons Learned

- **Technical:** Managing the lifecycle, memory usage, and synchronization of multiple ML models in a web server environment requires careful architectural design.
- **Data Handling:** Addressing class imbalance through techniques - like WeightedRandomSampler is crucial for building fair and accurate multi-class classification models.

- **Teamwork:** Clear communication, version control (Git), and modular development were essential for coordinating parallel frontend and backend development.
- **Project Management:** Breaking down a complex project into small, testable milestones was crucial for maintaining progress, tracking issues, and sustaining team morale.
- **Skill Development:** The team gained extensive, hands-on experience in full-stack development, deep learning model deployment, API integration, and user-centric design.

1.14 Recommendations and Future Scope

1.14.1 Improvements in Future

1. **Enhanced Computer Vision Pipeline:** Integrate a depth estimation model (e.g., MiDaS) to understand the 3D geometry of the room. This would enable more accurate scaling and placement of virtual furniture.
2. **User System & Gallery:** Implement user authentication and accounts, allowing users to save, manage, and revisit a history of their design projects.
3. **Model Optimization:** Collect more data to further fine-tune the YOLOv8 model on a wider range of furniture types and styles. Explore quantizing models for faster inference.

1.14.2 Research and Industrial Applications

- **E-commerce Integration:** Partner with furniture retailers to integrate their product catalogs directly into the system. Users could generate designs with purchasable items, and the system could provide direct links for purchase.
- **Advanced Generative Techniques:** Research and integrate more controllable generative techniques (e.g., ControlNet) to have finer-grained control over the AI-generated output, such as preserving the original room layout while changing the style.
- **VR/AR Mode:** Develop a Virtual Reality or true Augmented Reality mode (using device cameras) for a fully immersive design experience.

1.14.3 Long-term Vision

The long-term vision is to evolve this prototype into a market-ready SaaS (Software as a Service) platform. It could serve two main markets: as an AI-powered co-pilot for interior design professionals, enhancing their workflow, and as a direct-to-consumer web and mobile application that empowers homeowners. This tool has the potential to fundamentally change how spaces are designed, furnished, and experienced.

1.15 Acknowledgment

We would like to express our profound gratitude and sincere appreciation to our project guide, **Dr. Bhargav Prajwal Pathri**, Assistant Professor at Woxsen University, for his invaluable guidance, constant encouragement, and insightful feedback throughout the duration of this capstone project. His expertise and patience were instrumental in helping us navigate the technical challenges we encountered and in shaping the direction of our work.

Our thanks extend to the open-source community and the developers of the various libraries and frameworks, such as PyTorch, Ultralytics, React, and FastAPI, which formed the backbone of this project.

Finally, we wish to acknowledge the unwavering support and motivation from our teammates and professors, which sustained us through this demanding yet rewarding journey.

1.16 GitHub Repository and Demo Link:

GitHub Repo Link: https://github.com/id12026/A-Smart-System-for-Personalized-Interior-Design-AR-Try-On.git

Demo Link:
https://drive.google.com/file/d/1XXW9fWHQQUhFWhLOHMcPthoh5N9CLa9r/view?usp=sharing

1.17 References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[3] Ultralytics, "YOLOv8: State-of-the-art computer vision models," Ultralytics Docs, 2024. [Online]. Available: https://docs.ultralytics.com/models/yolov8/

[4] Google, "Gemini API – Multimodal generative AI," Google AI for Developers, 2024. [Online]. Available: https://ai.google.dev/gemini-api

[5] React Team, "React – A JavaScript library for building user interfaces," React Docs, 2024. [Online]. Available: https://react.dev

[6] T. Palpanas et al., "FastAPI: High performance web framework for building APIs with Python 3.7+," FastAPI Documentation, 2024. [Online]. Available: https://fastapi.tiangolo.com

[7] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 8024–8035. [Online]. Available: https://pytorch.org

[8] R. Goyal et al., “Three.js – JavaScript 3D library,” Three.js Official Documentation, 2024. [Online]. Available: https://threejs.org

[9] SQLite Consortium, “SQLite: Small. Fast. Reliable,” SQLite Documentation, 2024. [Online]. Available: https://www.sqlite.org

[10] OpenCV Team, “OpenCV: Open Source Computer Vision Library,” OpenCV Documentation, 2024. [Online]. Available: https://opencv.org

[11] Ant Design Team, “Ant Design – An enterprise-class UI design language and React UI library,” Ant Design Documentation, 2024. [Online]. Available: https://ant.design

[12] M. Everingham et al., “The Pascal Visual Object Classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[13] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[14] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018, doi: 10.1109/TPAMI.2017.2723009.

[15] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019. [Online]. Available: https://arxiv.org/abs/1711.05101

[16] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 10674–10685, doi: 10.1109/CVPR52688.2022.01042.

[19] A. Ramesh et al., “Zero-shot text-to-image generation,” in *Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8821–8831. [Online]. Available: https://arxiv.org/abs/2102.12092

[20] jsPDF Team, “jsPDF – Generate PDF files in client-side JavaScript,” jsPDF Documentation, 2024. [Online]. Available: https://github.com/parallax/jsPDF

[21] html2canvas Team, “html2canvas – Screenshots with JavaScript,” html2canvas Documentation, 2024. [Online]. Available: https://html2canvas.hertzen.com

[22] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” 2016. [Online]. Available: https://www.tensorflow.org