Arab International University

Faculty of Informatics and Communication Engineering

Senior Project Report on

AI Home Decorator

Submitted to

Department of Informatics Engineering

in partial fulfillment of the requirement for the Degree of Bachelor in

Informatics Engineering

Submitted by

Enaam Anjo **Reham Ghazi**

Haya Okar

Under the Supervision of

Dr. Ubai Sandouk **Eng. Khloud Al-Jallad**

July 2023

May 2023

Faculty of Informatics & Communication Engineering

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Informatics Engineering for acceptance, a project report entitled Project **AI Home Decorator** Submitted by: Enaam Anjo, Haya Okar, Reham Ghazi in partial fulfilment for the degree of Bachelor of Engineering in Informatics.

Supervisor	**Head of Department**
Dr. Ubai Sandouk	**Dr. Said Desouki**

Arab International University

The Arab International University (AIU) is a private Syrian university established in 2005. Its academic plans and the documents issued by it are approved and certified by the Ministry of Higher Education in the Syrian Arab Republic.

The university works to achieve the following goals:

- Preparing a distinguished generation of university graduates who are able to meet and advance the specific needs of society.
- Contributing to theoretical and applied scientific research that serves the purposes of national development. Work is being done to urge professors and academic staff to scientific research and participate in conferences and seminars that organize research.
- Achieving partnership with prestigious Arab and foreign universities with the aim of continuous development and modernization of academic work and conducting joint scientific research.
- Attracting distinguished academic and research competencies by providing the appropriate environment for their work.

The Arab International University is one of the first Syrian universities that have been established and inaugurated. It has been able to attract distinguished educational, research and administrative competencies, to create an integrated edifice from the academic, organizational and administrative aspects. It was able to graduate cadres of distinguished innovators by providing an educational environment based on unique qualitative and material ingredients, including:

- Modern and advanced study plans based on the credit hour system.
- Carefully selected educational cadres.
- Modern scientific laboratories and a laboratory for electronic libraries.
- Physical and moral incentives for students.
- Application of interactive teaching methods.
- Academic and educational guidance and counseling.
- A wide range of scientific cooperation agreements with local, regional and international universities of reputable reputation.
- Multiple agreements and memoranda of understanding with many civil society organizations.
- A proper campus with all the facilities of science, sports and entertainment, which we encourage you to visit and learn about their features.
- Student activities and clubs of all kinds: athletic, cultural, scientific and social.

The Arab International University life years are a time to invest in a student's future. The knowledge and experience that students acquire in the lecture hall and laboratories will help them in developing themselves. They will provide them with reasons for success in the chosen specialty. The student activities will help in expanding students' horizon. The activities of training, clubs and sports will enable students to develop their talents, and may even help in discovering new talents.
Students could invest time, mind and spirit in our university in order to reap the benefits of works and the time devoted in the coming years. We will be by our students in every step of their way.

Acknowledgements

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide Dr. **Ubai Sandouk** for his valuable guidance, encouragement and help for completing this work, his useful suggestions for this whole work and cooperative behavior are sincerely acknowledged.

We are also grateful to Eng. **Khloud Al-Jallad** for her constant support and guidance.

At the end we would like to express our sincere thanks to all our families, friends and others who helped us directly or indirectly during this project work.

Abstract

This report presents the development of “**DesignMate**”, an innovative AI home decorator application designed to revolutionize interior design. With three main features powered by artificial intelligence, DesignMate simplifies and enhances the process of home decoration.

The first feature leverages an Autoregressive transformer model trained on the extensive 3Dfront dataset to suggest room decor based on room layouts. By analyzing dimensions, furniture placement, DesignMate generates personalized recommendations, allowing users to effortlessly visualize and implement captivating room designs.

The second feature employs Generative Adversarial Networks (GANs) to enhance the colors of specific room layouts. Trained on the LSUN dataset, DesignMate GAN model produces visually stunning color variations, enabling users to explore harmonious color schemes and unlock limitless creative possibilities.

Best result obtained was 0.56 in scene classification accuracy which is the accuracy of a classifier trained specifically to discriminate between real and synthetic scenes (lower accuracy is better).

The third feature introduces an expert system that tailors decor options to user-entered conditions. By considering factors such as room size, lighting, and personal style, DesignMate's expert system suggests decor choices that create a cohesive and aesthetically pleasing living space.

DesignMate also introduces an integrated e-commerce platform dedicated to furniture, offering users a wide selection of high-quality items that perfectly complement their preferred room designs. With a user-friendly interface, this feature provides a seamless and convenient shopping experience, bridging the gap between design inspiration and tangible products.

Contents

Abbreviations

AI	Artificial Intelligence
CNN	Convolutional Neuron Network
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
GAN	Generative Adversarial Network
UI	User Interface
LOGAN	Local Control Generative Adversarial Network
RGB	Red Green and Blue
ML	Machine Learning
VCAE	Volumetric Convolutional Autoencoder
RANSAC	Random Sample Consensus
3DFRONT	3D Furnished Rooms with layOut and semantics
3DFUTURE	3D FUrniture shape with TextURE
LSUN	Large-scale Scene Understanding
3DSLN	3D Scene Layout Network
SUNCG	Semantic Scene Completion from a Single Depth Image
CGAN	Conditional Generative Adversarial Net

Keywords

Scene generation, Scene enhancement, Transformers, Image generation, Generative Adversarial Network, Flutter.

TABLE OF Tables

TABLE OF FIGURES

Introduction

The world of interior design has traditionally relied on the expertise of professionals and manual processes that can be time-consuming and subjective. However, with the rapid advancements in artificial intelligence (AI), a new era of interior design is emerging. AI-powered technologies have the potential to revolutionize the way of approaching home decoration, making it more accessible, efficient, and personalized.

By harnessing the power of AI, advanced algorithms can now be leveraged to analyze vast amounts of design data and user preferences. This enables the generation of personalized room designs that cater to individual tastes and requirements. With AI-generated designs, individuals can explore various styles, experiment with different layouts, and visualize their dream spaces in a more efficient and accurate manner.

Moreover, AI models equipped with image recognition and enhancement capabilities bring an added layer of sophistication to the interior design process. These models can enhance the aesthetics of input images, allowing individuals to see realistic previews of how different design elements, color schemes, and furniture choices would look in their own rooms. This visualization empowers users to make more informed decisions, ensuring that their design choices align with their vision and desired outcomes.

Furthermore, the integration of an ecommerce online shopping platform dedicated to furniture provides a seamless experience for individuals to browse and purchase items that perfectly complement their preferred room designs. This eliminates the need for extensive research and multiple sources, offering a convenient and efficient way to furnish and accessorize their spaces.

This report introduces an AI home decorator, which combines the power of AI-generated room designs, image enhancement algorithms, and an integrated ecommerce platform. By embracing these technologies, this project aims to revolutionize the interior design experience, making it more accessible, personalized, and visually appealing for individuals seeking to transform their living spaces.

Chapter 1: Project Description

1.1 Background

The field of interior design has traditionally relied on the expertise of professionals and a time-intensive process of manual decision-making. However, the emergence of artificial intelligence (AI) has opened up new possibilities for transforming the way of approaching home decoration. AI-powered technologies can now analyze vast amounts of design data, learn from user preferences, and generate personalized recommendations for room designs. This streamlines the decision-making process and empowers individuals to explore and experiment with different styles, color schemes, and furniture layouts, all with the click of a button.

In addition to AI-generated room designs, image recognition and enhancement algorithms have become increasingly sophisticated. By leveraging AI models specifically designed to enhance the décor of input images, users can obtain a realistic preview of how their existing spaces could be transformed. This technology allows for visualizing the impact of different design elements, such as furniture choices, color palettes, and lighting arrangements. The integration of AI and image enhancement not only saves time and effort but also ensures that individuals can make informed decisions that align with their personal style and vision for their home.

1.2 Problem Statement

The traditional process of interior design and home decoration is often characterized by manual effort, subjective decision-making, and limited access to design resources. This poses several challenges for individuals seeking to transform their living spaces into aesthetically pleasing and functional environments. These challenges include:

1. Time and Effort: Designing and decorating a room from scratch involves extensive research, multiple iterations, and coordination of various design elements. This process can be time-consuming and overwhelming for individuals with limited knowledge or experience in interior design.
2. Limited Design Expertise: Not everyone has access to professional interior designers or possesses the expertise to create visually appealing room designs. This results in a reliance on generic design templates or subjective decision-making, which may not fully capture an individual's unique style and preferences.

3. Visualization of Design Concepts: It can be challenging for individuals to visualize how different design elements, color palettes, and furniture choices would look in their own spaces. This lack of visualization hampers the decision-making process and makes it difficult to assess the potential impact of design choices.
4. Access to Design Resources: Finding suitable furniture and décor items that match the desired aesthetic can be a cumbersome task. Limited access to a wide range of design resources, including furniture options and décor inspiration, restricts the ability to explore diverse styles and make informed choices.

To address these challenges, there is a need for an innovative solution that leverages the power of artificial intelligence and technology to streamline the interior design and home decoration process. Such a solution should provide personalized recommendations, facilitate visualizations of design concepts, and offer access to an extensive range of furniture and décor options. By overcoming these challenges, individuals can efficiently and effectively transform their living spaces into environments that reflect their unique style and preferences.

1.3 Project Objective

The project aims to revolutionize the interior design process through the development of an AI home decorator. By leveraging AI-generated room designs, image enhancement algorithms, and an integrated ecommerce platform, this project strive to provide users with a seamless and personalized experience. Objectives include generating customized room designs, enhancing the aesthetics of input images, simplifying furniture shopping, saving users time and effort, and enhancing accessibility to design resources. Ultimately, this project aims to empower individuals to create personalized and visually appealing living spaces that reflect their unique style and preferences.

1.4 Project Scope

This project entails the development of an AI home decorator application that encompasses an AI model for personalized room design generation, an AI model for image enhancement and visualization, and an integrated ecommerce platform for furniture shopping. The application aims to provide users with a seamless and user-friendly experience, allowing them to input their preferences, view AI-generated room designs, enhance input images, and conveniently browse and purchase furniture items. The main objective is to deliver an innovative solution that empowers users to effortlessly transform their living spaces and personalize their home decoration process.

1.5 Project Features

DesignMate provides the following features:

1. AI Room Design Generation:
    - User Input Parameters: The application will allow users to input specific parameters such as room type, floor plan, and the number of design sequences they want to generate.
    - Personalized Room Designs: An AI model will analyze the input parameters and generate personalized room designs that cater to the user's preferences and requirements.
    - Visual Representation: The generated room designs will be presented to the user in a visually appealing format, including 2D or 3D renderings, floor plans, and furniture layouts.
2. Image Enhancement and Visualization:
    - Image Recognition: The application will utilize AI algorithms to recognize and analyze input images of existing spaces provided by the user.
    - Aesthetic Enhancement: The AI model will enhance the aesthetics of input images by adjusting color schemes, lighting, furniture placement, and other design elements.
    - Realistic Visualizations: Users will be able to visualize how their existing spaces could look after design enhancements, providing a realistic preview of the potential changes before implementation.
3. Expert System for Decor Scheme Suggestions:
    - User Input Parameters: The application will allow users to input their specific preferences, such as the gender of the desired room style, Budget, the available space, and any specific themes or inspirations they have in mind.
    - AI Analysis: The expert system will analyze the user's input parameters and leverage its knowledge base of design principles, color theory, and decor styles to generate decor scheme suggestions that align with the user's preferences.
    - Personalized Decor Schemes: Based on the analysis, the application will generate personalized decor schemes.
4. Integrated Ecommerce Platform:
    - Furniture Catalog: The application will feature an extensive catalog of furniture items, including sofas, tables, chairs, and decor accessories.
    - Convenient Shopping Experience: Users will have the ability to browse and select furniture items directly from the AI-generated room designs.

- Product Details: The ecommerce platform will provide detailed product information, including specifications, dimensions, etc.… to assist users in making informed purchasing decisions.

5. User-Friendly Interface:
   - Input and Customization: The application will provide an intuitive interface for users to input their preferences and customize design elements according to their preferences.
   - Easy Navigation: The application will feature a user-friendly interface that allows for seamless navigation between different features and functionalities.

1.6 Project Feasibility

The proposed project is feasible due to technical, resource, time, economic, and stakeholder factors. It requires access to computing resources such as powerful hardware and sufficient storage space for mode-related datasets, and can be implemented within a reasonable budget. It also meets the interests and needs of stakeholders such as decor enthusiasts, designers, and researchers by providing an easy-to-use platform for image generation and collaboration.

1.7 System Requirements

The AI home decorator application requires a computer or mobile device with an internet connection. The application can be accessed through a mobile app. No specific hardware specifications are required, but a stable internet connection is necessary for seamless access to the application's features and the online furniture catalog. The user interface is designed to be user-friendly and intuitive, providing a smooth and enjoyable experience for users on various devices.

Chapter 2: Theoretical Study

2.1 3D Image generation

According to [1], 3D image generation is a process that involves the creation of three-dimensional visual representations of objects, scenes, or environments using computer-generated techniques. It utilizes mathematical algorithms and rendering techniques to generate realistic and immersive 3D images that simulate depth, perspective, and lighting effects. This technology finds applications in various fields, including computer graphics, virtual reality, gaming, architecture, and product design. By leveraging 3D modeling, texturing, and rendering techniques, 3D image generation enables the creation of lifelike and interactive visual experiences. It allows users to explore and interact with virtual environments, visualize complex structures, simulate real-world scenarios, and bring imagination to life. The advancements in hardware capabilities, software algorithms, and rendering technologies have significantly enhanced the quality and realism of 3D image generation, enabling the creation of visually stunning and immersive virtual worlds.

2.1.1 Indoor scene generation

According to [2], Indoor scene generation is the process of creating virtual representations of indoor environments using computer-generated techniques. It involves the synthesis of 3D models, textures, lighting, and other elements to generate realistic and immersive indoor scenes. Indoor scene generation finds applications in fields such as architecture, interior design, virtual reality, and gaming. By leveraging advanced modeling and rendering techniques, indoor scene generation enables the creation of lifelike virtual spaces that simulate real-world indoor environments. These virtual scenes can be customized and manipulated to showcase different interior designs, furniture layouts, lighting configurations, and material choices. Indoor scene generation allows designers, architects, and users to visualize and explore indoor spaces before the physical construction or renovation takes place. It provides an efficient and cost-effective way to experiment with different design options, evaluate spatial arrangements, and create compelling visual presentations. The advancements in computer graphics, rendering technologies, and virtual reality have significantly improved the realism and interactivity of indoor scene generation, making it a valuable tool in the architectural and design industries.

2.2 Indoor scene Enhancement

According to [3], Indoor scene enhancement is a process that involves improving the visual quality, realism, and aesthetic appeal of virtual indoor scenes. It encompasses various techniques and algorithms that aim to refine the lighting, textures, materials, and overall presentation of the scene. By applying advanced rendering and image processing techniques, indoor scene enhancement can address issues such as inadequate lighting, low-quality textures, or unrealistic reflections in virtual environments. These techniques can also enhance the visual details, colors, and overall ambiance of the indoor scene, resulting in a more visually compelling and immersive experience. Indoor scene enhancement finds applications in fields such as architecture, interior design, virtual reality, and gaming, where creating visually stunning and realistic indoor environments is crucial. By enhancing the visual quality and fidelity of virtual indoor scenes, indoor scene enhancement techniques contribute to creating more engaging and immersive virtual experiences for users and viewers.

2.3 Online shopping

According to [4], Online shopping, also known as e-commerce, has revolutionized the way of shopping by providing a convenient and accessible platform for purchasing goods and services over the internet. With online shopping, consumers can browse and shop for a wide range of products from the comfort of their own homes or on-the-go using their mobile devices. The convenience of online shopping allows users to compare prices, read product reviews, and make informed purchase decisions. Additionally, online shopping offers a vast selection of products from various retailers, eliminating geographical limitations and providing access to global markets. Secure payment gateways and encryption technologies ensure safe transactions, while efficient logistics and delivery services ensure that products are delivered to the customer's doorstep in a timely manner. Online shopping has transformed the retail landscape, offering unparalleled convenience, choice, and flexibility to consumers worldwide, and continues to shape the way of buying and selling goods in the digital age.

2.4 Transformers

According to [5], Transformers are a type of deep learning model that have revolutionized the field of natural language processing (NLP). They were introduced in the paper "Attention Is All You Need" by Vaswani et al. in 2017 and have since become one of the most influential architectures in NLP.

The key innovation of transformers is the self-attention mechanism, which allows the model to capture relationships between different words or tokens in a sequence. Unlike traditional recurrent neural networks (RNNs), which process sequences sequentially, transformers can process all the tokens in a sequence simultaneously. This parallelism makes transformers highly efficient and enables them to capture long-range dependencies more effectively.

Self-attention mechanisms allow the model to assign weights to different parts of the input sequence, indicating the relevance or importance of each token for a given context. By attending to different parts of the sequence, transformers can effectively model relationships between distant words, making them particularly effective for tasks that require understanding long-range dependencies, such as machine translation and document summarization.

The transformer architecture consists of two main components: the encoder and the decoder.

2.4.1 Transformer encoder

The encoder component of the transformer processes the input sequence and generates a representation that captures the contextual information of each token. It consists of several identical layers, typically stacked on top of each other. Each layer has two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network.

- Self-Attention: This is a key component of transformers. Self-attention allows the model to weigh the importance of different words in the input sequence when generating the representation for each word. It attends to all other words in the sequence, generating a weighted representation that reflects the relevance of each word for the current word being processed. The self-attention mechanism is applied multiple times, with different learned attention weights, allowing the model to capture different aspects of the input sequence.

- Feed-Forward Network: After the self-attention mechanism, a feed-forward network is applied to each position in the sequence independently. It consists of two linear transformations with a non-linear activation function in between, typically a ReLU (Rectified Linear Unit). The feed-forward network helps capture complex patterns and interactions between different parts of the sequence.

2.4.2 Transformer decoder

The decoder component takes the representation generated by the encoder and uses it to generate an output sequence. It also consists of multiple identical layers, each with two sub-layers: a masked multi-head self-attention mechanism and a multi-head attention mechanism over the encoder's output.

- Masked Self-Attention: In the decoder, a masked self-attention mechanism is employed to ensure that each position can only attend to earlier positions in the output sequence. This is done to prevent the model from "cheating" by looking at future positions during training. By attending only to the already generated part of the output sequence, the decoder can generate the output in an autoregressive manner.

- Encoder-Decoder Attention: The decoder also incorporates an attention mechanism over the encoder's output. This allows the decoder to attend to different parts of the input sequence while generating the output. By attending to relevant information from the encoder, the decoder can align the input and output sequences and generate accurate and contextually appropriate translations or responses.

2.4.3 Autoregressive Transformers

According to [6], Autoregressive transformers are a type of transformer model specifically designed for sequential generation tasks. They generate output sequences one token at a time, where each token depends on previously generated tokens. Autoregressive transformers use a masked self-attention mechanism during training to ensure that each token can only attend to earlier positions in the output sequence. This allows the model to capture dependencies and generate coherent and contextually appropriate output. Autoregressive transformers are particularly effective for tasks where sequential generation and capturing dependencies are crucial.

2.5 Development tools

2.5.1 Flutter

Flutter [7] is an open-source UI toolkit by Google for building high-quality, cross-platform apps using a single codebase. It uses Dart programming language and offers pre-built UI components, rapid development with hot reloading, and smooth animations. Flutter's widget-based architecture promotes code reuse, and its rendering engine ensures fast and visually appealing apps. With a strong community and excellent tooling support, Flutter is popular for efficient multi-platform app development.

2.5.2 Django

Django [8] is a high-level open-source web framework written in Python, designed to simplify and speed up the development of robust web applications. It follows the model-view-template (MVT) architectural pattern and promotes code reusability. Django offers a wide range of features, including an ORM layer for seamless database integration, a powerful templating engine for flexible content rendering, and built-in user authentication. With Django, developers can quickly build secure and maintainable web apps, thanks to its comprehensive documentation, active community support, and vast ecosystem of reusable packages. Django's focus on convention over configuration allows developers to concentrate on their application's logic while the framework handles the infrastructure.

2.5.3 Postman

Postman [9] is a popular API development tool that simplifies designing, testing, and documenting APIs. It offers a user-friendly interface for creating and sending HTTP requests, making API interaction and response analysis easier. Postman allows developers to organize and test API requests, apply authentication methods, and automate endpoint validation. It also facilitates the generation of comprehensive API documentation. With its features, Postman enhances developer efficiency and productivity in API development and testing.

2.5.4 Experta

In artificial intelligence, an expert system is a computer system emulating the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if–then rules rather than through conventional procedural code. The first expert systems were created in the 1970s and then proliferated in the 1980s. Expert systems were among the first truly successful forms of artificial intelligence (AI) software. An expert system is divided into two subsystems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities. In order to design an expert system, this project uses experta. Experta is a Python library for building expert systems strongly inspired by CLIPS.

2.5.5 Fast transformers

Fast Transformer is a library provided by Katharopoulos et al. [10]. Fast Transformers is a highly valuable and user-friendly Python library that simplifies the implementation of transformer models. Specifically designed for fast attention in transformers, this library streamlines the process of working with transformers in research and development. It provides a range of helpful features and optimizations to enhance the efficiency and performance of transformer models, making it an indispensable tool for researchers and developers working with transformers.

2.6 Similar applications

1. Ikea place [11]: The "IKEA Place" application is an augmented reality app that allows users to place IKEA furniture in their surrounding space using the camera on their smartphones. This application serves as a useful tool for trying out furniture pieces and envisioning how they will look in the actual location before making a purchase. Users can browse through a curated list of furniture pieces within the app and select the item they wish to try in augmented reality.

2. "Houzz" application [12]: The "Houzz" application is a popular platform and app for home remodeling and design. It provides users with a wide range of features and resources to help with interior design, home improvement projects, and finding professionals in the industry.

3. Homestyler Interior Design [13]: The "Homestyler Interior Design" application is a popular tool for interior design and home decorating. It offers users various features to visualize and plan their interior spaces. Here are some key features of the Homestyler Interior. Homestyler provides access to a wide range of furniture and decor items from various brands and retailers. You can browse through the catalog, select items, and place them in your design. The app also provides links to purchase the products directly.

4. Planner 5D [14]: is a popular application and online platform that allows users to create, visualize, and design their own home interiors and exteriors in 2D and 3D. It provides a user-friendly interface and a wide range of features to assist users in creating detailed floor plans, experimenting with different design options, and visualizing their ideas in a realistic manner.

Table 1: similar applications comparison

Application	**E-commerce app**	**Décor generation**	**Décor Enhancement**	**Helper Expert system**	**Augmented reality**
Ikea place	✓	✗	✗	✗	✓
Houzz	✓	✓	✗	✗	✓
Homestyler	✓	✓	✗	✗	✓
Planner 5D	✗	✓	✓	✗	✓
DesignMate	✓	✓	✓	✓	✗

Chapter 3: Literature Review

In this chapter, projects similar to our projects will be mentioned, and discussed and compared with ours.

3.1 Scene generation

3.1.1 Datasets

3.1.3.1 SUNCG Dataset:

The SUNCG dataset is a large-scale synthetic dataset designed for scene understanding and reconstruction in indoor environments. It provides a diverse collection of 3D scenes representing various types of indoor spaces, including homes, offices, and public buildings. The dataset aims to facilitate research in computer vision, robotics, and other related fields by offering a rich resource for training and evaluating algorithms.

The SUNCG dataset includes a vast number of scenes, each consisting of a 3D layout along with textured 3D models of furniture and objects. The layouts define the structural elements of the scene, such as walls, floors, and ceilings, while the 3D models represent furniture items like chairs, tables, beds, etc. The dataset also contains metadata, such as room and object labels, semantic segmentation masks, and camera parameters. The SUNCG dataset has been widely used for various applications, including scene parsing, furniture layout synthesis, scene reconstruction, and object recognition. It provides a valuable resource for developing and evaluating algorithms that aim to understand and interact with indoor scenes.

This dataset is not available to use at the time.

Figure 1:SUNCG Dataset

3.1.3.2 InteriorNet Dataset:
An end-to-end pipeline has been developed to render an RGB-D-inertial benchmark dataset for large-scale interior scene understanding and mapping. The dataset comprises 20 million images generated through the following pipeline steps:

(A) Collection of approximately 1 million CAD models obtained from renowned furniture manufacturers, which are extensively utilized in real-world production.

(B) Utilization of these CAD models by a team of 1,100 professional designers to create around 22 million interior layouts, many of which have been employed in real-world interior decorations.

(C) Generation of multiple configurations for each layout to represent diverse lighting conditions and simulate scene changes over time, mimicking daily life scenarios.

(D) Provision of an interactive simulator called ViSim, which aids in generating ground truth inertial measurement unit (IMU) data, events, as well as monocular or stereo camera trajectories. These trajectories can be created through methods such as hand-drawn, random walking, or neural network-based realistic trajectory generation.

(E) Inclusion of all supported image sequences and corresponding ground truth data within the dataset.

This comprehensive pipeline ensures the availability of a rich and diverse benchmark dataset for advancing research in interior scene understanding, mapping, and related fields.

3.1.3.3 SceneNN Dataset:
An RGB-D scene dataset is presented, comprising over 100 indoor scenes captured from diverse locations, including offices, dormitories, classrooms, pantries, and more. These scenes have been captured at the University of Massachusetts Boston and the Singapore University of Technology and Design.

To enhance the dataset's utility, all scenes have been reconstructed into triangle meshes and annotated with per-vertex and per-pixel information. Additionally, the dataset has been enriched with detailed annotations, including axis-aligned bounding boxes, oriented bounding boxes, and object poses. These additions provide fine-grained information that can be valuable for various scene analysis and understanding tasks.

3.1.3.4 3DFRONT Dataset:
The 3DFRONT dataset [15] is a large-scale 3D dataset that focuses on human-centric scenes. It contains detailed 3D models of indoor scenes, including various types of rooms and furniture arrangements.

This dataset was utilized in the experiment.

DATASET description:
The dataset comprises high-quality 3D models, these models are created by expert designers and artists, ensuring accuracy and attention to detail in their geometrical and textural attributes.

For each 3D model, the dataset includes comprehensive metadata, providing vital information about the object or scene depicted. This metadata encompasses attributes such as category, dimensions, texture details, and annotations, facilitating efficient categorization, search, and retrieval of models based on specific criteria.

Furthermore, Zeng et al. also release Trescope, a light-weight rendering tool, to support benchmark rendering of 2D images and annotations from 3D-FRONT. Unfortunately, it is not available for windows yet.

In addition to the individual 3D models, the 3D Front dataset also offers pre-constructed scenes or environments, combining multiple objects to create realistic and immersive settings. These scene configurations enable us to examine spatial relationships between objects, study lighting effects, and simulate real-world scenarios.

The Dataset includes 6,813 distinct houses and 18,797 diversely furnished rooms and it uses 3DFUTURE dataset to provide 13,151 furniture objects all come with high-quality textures

Figure 2:3DFRONT samples rendered using Terscope

Dataset format

The 3D-FRONT dataset includes house layouts file (3D-FRONT.zip), furniture shapes file (3D-FUTURE-model.zip), and wall / floor texture file (3D-FRONT-texture.zip).

1. house layouts:

The house layouts file has the following form:

3D-FRONT.zip

├── 3D-FRONT

│ ├── 3085ee0a-7da1-466e-9d5f-71ad571a25c0.json

| ├── 87b3932e-b5b2-4539-9451-8e3abadf41b7.json

| |….

Each Json file represents a scene from the dataset, the file contains description of all aspects that make up the scene.

Figure 3: json file structure

- The furniture >> jid in json files is corresponding to the model id in downloaded furniture shapes.
- The furniture >> valid: false in json files represents a decoration (e.g. plants) that do not have a corresponding model.

2. furniture shapes

The furniture shapes file has the following form:

3D-FUTURE-model.zip

├── categories.py

```
├── model_info.json
├── 3D-FUTURE-model
│ ├── 209fdbd6-b95d-4d11-a05e-8fcefa32a60d
│ │ ├── image.jpg
│ │ ├── normalized_model.obj
│ │ ├── raw_model.obj
│ │ ├── texture.png
│ │ ├── model.mtl
│ ├── 209fdbd6-b95d-4d11-a05e-8fcefa32a60d
│ │ ├── image.jpg
│ │ ├── normalized_model.obj
│ │ ├── raw_model.obj
│ │ ├── texture.png
│ │ ├── model.mtl
```

│ │...

- categories. py: includes furniture attributes (e.g. style, material, theme) and categories used in 3D-FRONT dataset, which are given by experienced designers.
- model_info.json: describes the style, theme, material, and category of each furniture, except lights.

3. *wall and floor texture:*

The wall and floor texture file has the following form:

3D-FRONT-texture.zip

├── categories.py

├── texture_info.json

├── 3D-FRONT-texture

│ ├── 4e3ea8a0-9add-4934-99f8-b3bebc804e12

│ │ ├── texture.png

│ ├── 4e880ff5-8633-4925-8965-e281887c53b1

│ │ ├── texture.png

│ │...

- categories. py: includes texture attributes (e.g. style) and categories used in 3D-FRONT dataset, which are given by experienced designers.
- texture_info.json: describes the style and category of each texture.

Datasets comparison

#3DFRs" represents the number of rooms or scenes populated with 3D furniture objects,

"N/A" = "not available"

Table 2: scene generation datasets comparison

Dataset	year	Layout Design	#3DFRs	REF.
SUNCG	2017	Real scan	N/A	[16]
InteriorNet	2018	Professional	N/A	[17]
Real scan	2016	Real scan	100	[18]
OpenRooms	2020	Real scan	1,068	[19]
3DFRONT	2021	Professional	18,968	[20]

3.1.2 Related works and algorithms used

In [21], they've focused on generative models (CNN-based model) trained on SUNCG dataset to select and place objects in room. The input is the type of the room to be decorated and geometry describing floor, walls, and ceiling. The model approach depends on generating a room by adding one object at a time iteratively. They've focused on generating and arranging major functional objects such as furniture which are placed on the floor. They do not address wall-mounted or second-tier objects. The solution contains three main steps: first deciding whether to add another object, second which type of object to be added and where, and finally place the object.

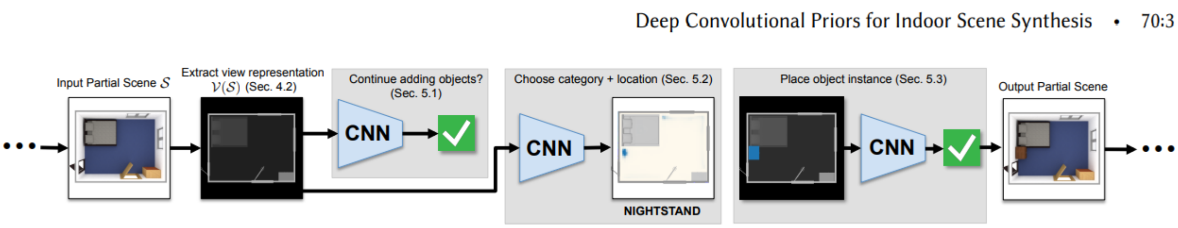


Figure 4: the pipeline of the scene synthesis.

Figure 2 illustrates the process of generating the scene, given an input scene S, it first computes a top-down view of the scene V(S). next step Is to analyze the image to determine whether to add another object, then choose the location and the category of the object. Finally, it selects an instance of that category and adds it to the scene with an appropriate orientation.

In [22], they've presented a new image-based scene synthesis pipeline, based on deep convolutional generative models, and trained on SUNCG dataset, the model is faster than previous works and generates scenes by iteratively adding objects. It factorizes the step of adding each object into a different sequence of decisions, allowing it to reason globally about which objects to add and model the spatial extent of the objects. It requires on average under 2 seconds to synthesize a scene.

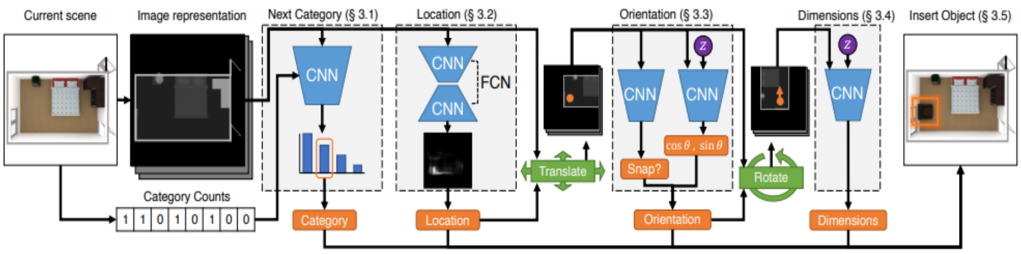


Figure 5: Overview of our automatic object-insertion pipeline.

The pipeline in Figure 3 extracts a top-down, floor-plan image representation of the input scene. This is done in a similar way to [21]. Then, it feeds this representation to four decision modules to determine how to add objects into the scene. These modules decide which kind of object to add, where it should be located, what direction it should face, and its physical dimensions. This is a different factorization than in prior work and leads to both faster synthesis and higher-quality results.

In [23] , they gave a solution for interior layout with user-specified furniture by using CGAN model (conditional generative adversarial net) to divide the room into several functional areas, and then train a simple fully connected network to place furniture in the corresponding functional areas and complete the furniture layout of the room.

Furniture should be separated into functional areas in a way that's user-specified. For instance, if the user doesn't select a sofa, there is no meeting area, or there is a smaller meeting area in the layout result. In other words, furniture information can help the model learn something that geometric features of the room cannot tell.

First step is <u>functional area division</u>:

Model for the functional area division is:

1. <u>Empty Room Input</u>: The initial step involves providing information about an empty room, including its dimensions and layout. Additionally, random noise, typically drawn from a normal distribution (N(0, 1)), and a furniture vector indicating the desired furniture arrangement are also inputted.

2. Functional Area Division: Based on the input, the system performs functional area division, which involves partitioning the room into different functional areas such as living room, dining area, kitchen, etc. This step determines the layout and placement of these functional areas within the given space.

3. Discriminator Training: Using a dataset of pre-divided functional areas and corresponding furniture vectors, a discriminator model is trained. This model learns to distinguish between generated results and real examples.

4. Guided Generation: The discriminator's output is then utilized to guide the generator model. By adjusting the generator's parameters, the system generates a new set of samples that aim to be more similar to real examples than the previous iteration. This generated layout is refined based on the feedback from the discriminator.

5. Iterative Process: Steps 2 to 4 are repeated in an iterative manner. The generated layout is fed back into the system, the discriminator evaluates it, and the generator adjusts its parameters to improve the results. This iterative loop continues until the desired layout quality is achieved.

The result of the functional area division is depicted in an image, where different types of functional areas are labeled with distinct colors. In this experiment, the meeting area is represented by the color green, the dining area by yellow, and the audiovisual area by purple. An algorithm based on the HSV color model is employed to extract the functional area information, including position and size. This algorithm utilizes the predefined color ranges associated with each functional area to accurately identify and delineate the boundaries of each area within the image. By leveraging the properties of the HSV color model, the algorithm can precisely determine the position and dimensions of the functional areas, laying the groundwork for subsequent furniture placement operations.

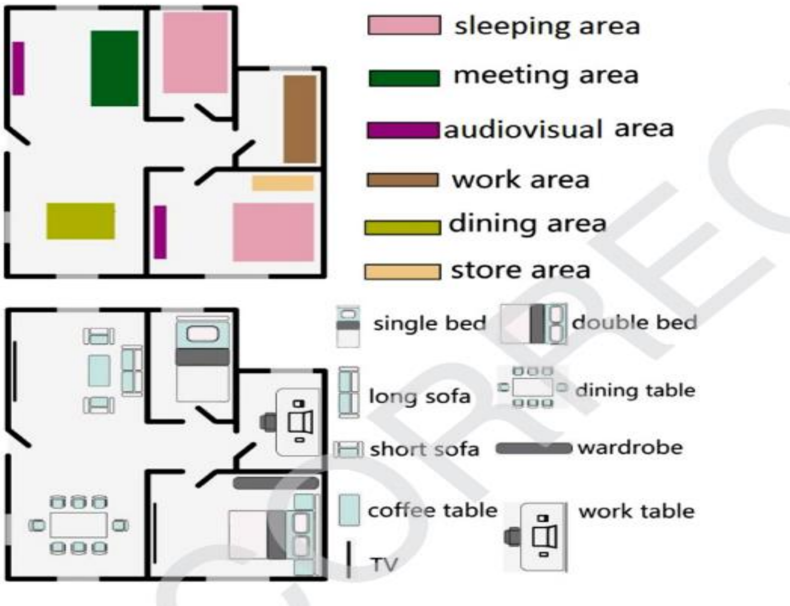


Figure 6: The mapping between the functional areas (indicated by colors) and furniture objects (indicated by legends).

The next step is <u>Furniture placement</u>:

The process of furniture placement within a functional area is simplified to selecting suitable furniture pieces and positioning them within the designated rectangular space without any additional constraints. This simplified approach allows for more flexibility in placing the furniture items, focusing solely on choosing the appropriate pieces and arranging them in the correct locations within the given area.

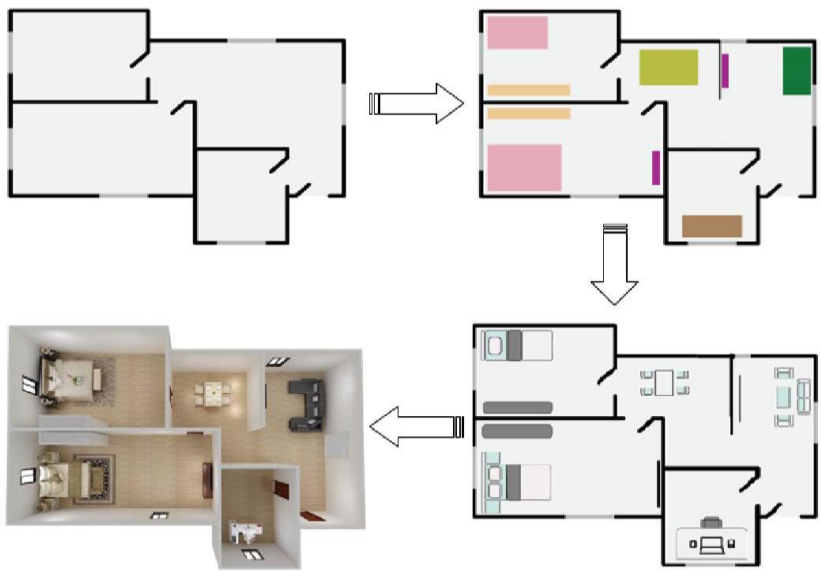


Figure 7:The processing of automatic furniture layout.

Dataset in this paper was generated and gathered from interior design company (anonymous) more than 4000 sets of apartments is chosen.

The limitation of such a process is that the layout result depends on the division of the functional areas, which is related to the training dataset in CGAN model. So, the functional areas division process should be improved.

In [24], Peter K´an, and Hannes Kaufmann introduce a novel approach for automated interior design through the utilization of a genetic algorithm optimization. The method consists of two main steps: furniture selection and positioning, and material assignment optimization.

In the furniture selection and positioning step, the system employs an iterative optimization process to determine the optimal layout. A cost function is formulated based on an extended set of interior design guidelines. The goal is to find a furniture arrangement that satisfies criteria such as avoiding intersections, adhering to ergonomic principles, and achieving aesthetic and functional coherence.

To address the challenges posed by the high dimensionality of the search space and the infinite possibilities of furniture configurations, the genetic algorithm is employed. It allows for simultaneous optimization across multiple dimensions, facilitating the exploration of various design possibilities.

The issue of unacceptable furniture configurations is tackled by incorporating interior design guidelines into the optimization process. These guidelines help guide the algorithm towards acceptable arrangements that meet specific design criteria.

In the second step, material assignment optimization is performed to achieve harmonious color configurations and consistent material types, Peter K´an, and Hannes Kaufmann propose a fast method based on greedy cost minimization, inspired by previous works on color compatibility assessment. A new labeling strategy based on material names and categories is introduced to enhance the material selection process.

In [2], Xinpeng Wang et al. have focused on generating scenes from room layouts by creating a set of objects and their arrangements within the room. Each object is generated with predicted class category, 3D location, angular orientation, and 3D size. Once this sequence is generated, the most appropriate CAD model for each object is selected from a database and placed in the scene at the predicted location. The CAD model selection takes into account factors like size, shape descriptor, texture, and other heuristics to reduce collisions, consider special object properties such as symmetry, and maintain style-consistency across objects.

The main contributions of the paper can be summarized as follows:

- The representation of an indoor scene as a sequence of object properties, which transforms scene generation into a sequence generation task.

- The utilization of self-attention in transformers to implicitly learn relationships between objects within a scene, eliminating the requirement for manually-annotated relations.
- The generation of complex scenes conditioned on room layouts or text descriptions through the use of discretized object coordinates to predict their 3D locations.

Figure 5 showcases the Layout-conditioned SceneFormer approach. The input to the model is the room layout, which includes information about the shape of the room as well as the positions of doors and windows. The SceneFormer model operates in a sequential manner, generating the properties of the next object in the scene and incorporating it into the existing scene. The resulting scene, after all objects have been generated and inserted, is displayed on the right side of the figure.

In [25], Zhang and Andrew proposed a novel model called 3D-SLN, which leverages a variational autoencoder and a graph convolutional network to generate realistic and diverse scene layouts. The key idea behind 3D-SLN is to utilize the relationships provided in a 3D scene graph to synthesize plausible scene layouts.

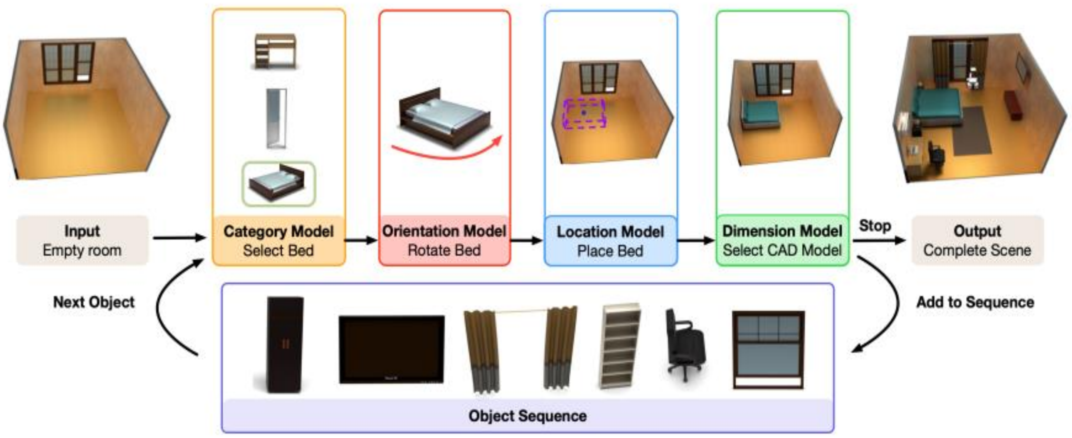


Figure 8: SceneFormer model

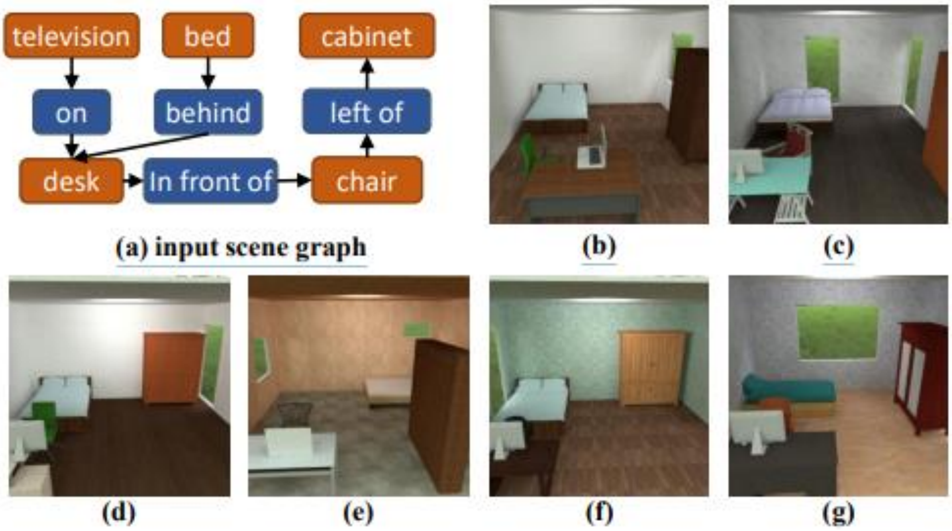


Figure 9: 3D-SLN overview

Figure 7 demonstrates conditional scene synthesis, (a) The input is a scene graph describing object relationships. (b)–(g) Diverse layouts synthesized conforming to the input scene graph.

The introduced model, referred to as the 3D Scene Layout Network (3DSLN), presents a general framework for synthesizing scene layouts from scene graphs. The 3DSLN combines a graph convolutional network with a variational autoencoder to generate diverse and plausible layouts based on the relationships described in the 3D scene graph.

The model produces 3D scene layouts by generating 3D bounding boxes and vertical rotations for each object. To facilitate this, the traditional 2D scene graphs are augmented to 3D scene graphs, incorporating object relationships in the 3D space. The X and Y axes are defined to span the floor plane of the room, while the Z axis represents the upward direction for objects above the floor. This definition allows constraints such as "left of" to limit the X and Y coordinates between object pairs, while the "on" relationship constrains the Z coordinate.

Each node in the scene graph not only specifies the object type but can also include attributes such as object height (tall, short) and volume (large, small). The scene graph is represented by a set of relationship triplets in the form of (oi, p, oj), where oi denotes the type and attributes of the i-th object, and p represents the spatial relationship.

To enable the model to operate on the input graph and generate multiple scenes from the same input, the 3D-SLN framework is proposed. It combines a graph convolution network (GCN) with a conditional variational autoencoder (cVAE). During training, the encoder is responsible for generating the posterior distribution of a given scene layout conditioned on the corresponding scene graph. The encoder takes a scene graph and an exemplar layout as input and outputs the posterior distribution for each object, represented by the mean and log-variance of a diagonal Gaussian distribution. A latent vector is then sampled from the Gaussian distribution for each object. Subsequently, the decoder takes the sampled latent vectors and the scene graph as input to generate a scene layout, represented by the 3D bounding box and rotation for each object.

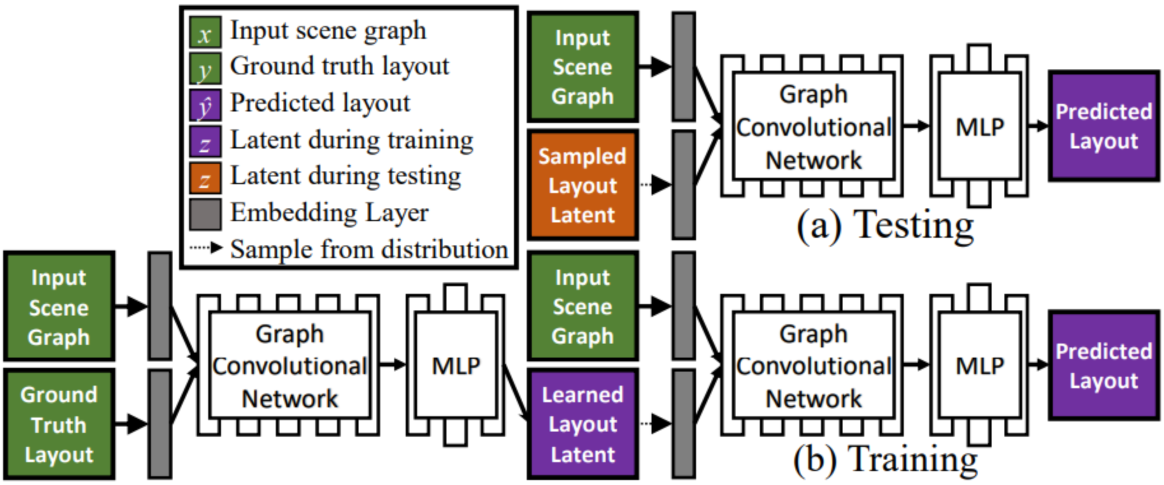


Figure 10: 3D Scene Layout Network (3D-SLN)

The network architecture of the scene layout generator is depicted in Figure 8. At test time, the generator follows the following steps:

(a) A latent code is sampled from a learned distribution. This latent code serves as an input to the decoder, along with the scene graph. The decoder generates the scene layout based on this information.

During training, the process is slightly different:

(b) An encoder takes the ground truth scene layout and the corresponding scene graph as inputs. The encoder converts this input into a distribution. From this distribution, the latent code is sampled. The sampled latent code is then decoded by the decoder to reconstruct the scene layout.

This architecture enables the model to learn and generate scene layouts by sampling latent codes from a distribution, while utilizing the information provided by the scene graph.

3.1.2 Comparisons

In order to assess the performance of scene synthesis models, researchers have employed scene classification accuracy as an evaluation metric. This metric measures the accuracy of a classifier trained specifically to discriminate between real and synthetic scenes. It is important to note that a lower accuracy is better.

Table 3: scene generation related works

Ref.	**Year**	**Dataset**	**Model**	**notes**	**Scene classification accuracy**
[21]	2018	SUNCG	CNN	extracts a top-down, floor-plan image representation of the input scene. Three main steps: • deciding whether to add another object (Continue?). • deciding what category of object to add and where (CategoryLocation). • inserting an instance of that object category into the scene (InstanceOrientation).	84.69
[22]	2019	SUNCG	GAN	• extracts a top-down, floor-plan image representation of the input scene. • which category of object to add to the scene, if any. • , where that object should be located. • what direction it should face. • its physical dimensions.	76.18
[23]	2019	anonymous interior design company generated dataset	CGAN	• CGAN model to divide the room into functional areas • fully connected network to arrange furniture accordingly	Not mentioned
[25]	2020	SUNCG	3D-SLN	• graph convolution network (GCN) • conditional variational autoencoder (cVAE)	72.34
[2]	2021	InteriorNet	Transformer	• represent an indoor scene as a sequence of object • leverage the self-attention of transformers • generate complex scenes conditioned on room layout	68.13

3.2 Scene Enhancement

3.2.1 Datasets

3.2.1.1 LSUN Bedroom:

The LSUN Bedroom dataset is a widely used dataset in computer vision and machine learning research. It is specifically designed for training and evaluating algorithms for tasks related to bedroom scene understanding and image synthesis. Here is a detailed explanation of the LSUN Bedroom dataset:

1. Dataset Description: The LSUN Bedroom dataset contains a large collection of high-resolution images depicting various bedrooms. It includes different styles, layouts, and furniture arrangements commonly found in bedrooms. The dataset aims to capture the diversity and complexity of real-world bedroom scenes.

2. Dataset Size: The LSUN Bedroom dataset has a size of approximately 43GB.

3. Image Characteristics: Each image in the dataset represents a single bedroom scene. The images are typically in color and have high resolution, allowing for detailed analysis and synthesis. The scenes may include various objects such as beds, windows, furniture, decorations, and architectural elements like walls, ceilings, and floors.

4. Annotation and Labels: The LSUN Bedroom dataset provides image-level labels that indicate the presence of a bedroom scene in each image. However, detailed object-level annotations or semantic segmentation masks are not typically available in this dataset. The dataset primarily focuses on capturing the overall bedroom scenes rather than specific object annotations.

5. Training and Evaluation: Researchers and practitioners often use the LSUN Bedroom dataset to train and evaluate algorithms for tasks such as image generation, image-to-image translation, scene understanding, object detection, and semantic segmentation in bedroom scenes. The dataset serves as a benchmark for developing and comparing state-of-the-art models and techniques in these domains.

6. Availability and Usage: The LSUN Bedroom dataset is publicly available and can be downloaded from various sources, including academic websites, online repositories, or dedicated dataset platforms. It is commonly used in conjunction with deep learning frameworks and computer vision libraries for training and evaluating models.

Figure 11: LSUN bedroom dataset

Researchers and practitioners leverage the LSUN Bedroom dataset to develop and evaluate algorithms for a wide range of applications, including interior design, virtual reality, augmented reality, robotics, and smart home technologies. By utilizing this dataset, researchers can explore and address challenges related to understanding and synthesizing bedroom scenes, enabling advancements in computer vision and artificial intelligence.

3.2.1.2 ScanNet:

The ScanNet dataset is a large-scale collection of three-dimensional (3D) data used in the fields of computer vision and image analysis. This dataset was collected using 3D sensing devices such as LiDAR sensors and depth cameras, and it includes scans indoor scenes.

The main goal of the ScanNet dataset is to provide a comprehensive and diverse set of 3D data for research purposes. It includes detailed geometric and semantic information about the scanned scenes, allowing researchers to develop and evaluate algorithms and models for various computer vision tasks.

The dataset consists of thousands of scans, each capturing a different scene or environment. These scans are represented as point clouds, which are dense collections of 3D points that describe the shape and structure of the objects in the scene. In addition to the geometric information, the dataset also provides semantic labels for different objects and regions within the scenes, enabling researchers to perform tasks such as object recognition, scene understanding, and semantic segmentation.

The ScanNet dataset has been widely used in various computer vision applications, including 3D reconstruction, scene understanding, object detection, and virtual reality. Its large size and diversity make it a valuable resource for training and evaluating algorithms in these domains. Researchers can access the dataset and use it to develop new techniques and advance the state of the art in computer vision.

3.2.1.3 Semantic3D:

The Semantic3D Dataset consists of a diverse collection of 3D scenes that provide detailed information about floors, walls, ceilings, furniture, and other static objects within the scenes.

In terms of its use in interior design, the Semantic3D Dataset can serve as a valuable source of 3D data for creating accurate and realistic interior models. Designers and architects can utilize this data to conduct advanced analyses and experiments in interior design and space planning. They can apply 3D visualization techniques to determine design details, furniture arrangement, space optimization, and provide clients with a realistic visual experience.

By using the Semantic3D Dataset in interior design, designers can embody and creatively test their ideas before implementing them in reality. This helps improve design trends, spatial planning, color coordination, and the selection of furniture, lighting, and other interior design elements. Additionally, the dataset can be utilized in the development of virtual reality applications and virtual visits to aid in conceptualization and client interaction with proposed interior designs.

Datasets comparison

Table 4: scene enhancement datasets comparison

Dataset	Size	year	description	REF
LSUN Bedroom	43GB	2015	• contains a large collection of high-resolution images depicting various bedrooms. • includes different styles, layouts, and furniture arrangements commonly found in bedrooms. • aims to capture the diversity and complexity of real-world bedroom scenes.	[26]
ScanNet	30 GB	2017	• consists of thousands of scans, each capturing a different scene or environment. • scans represented as point clouds, which are dense collections of 3D points that describe the shape and structure of the objects in the scene. • the dataset also provides semantic labels for different objects and regions within the scenes	[27]
Semantic3D	100GB	2017	• consists of a diverse collection of 3D scenes that provide detailed information about floors, walls, ceilings, furniture, and other static objects within the scenes. • the Semantic3D Dataset can serve as a valuable source of 3D data for creating accurate and realistic interior models. • Designers and architects can utilize this data to conduct advanced analyses and experiments in interior design and space planning. • They can apply 3D visualization techniques to determine design details, furniture arrangement, space optimization, and provide clients with a realistic visual experience.	[28]

3.2.2 related works

In [29], Chen Zhang, Yujun Shen and Yinghao Xu convey discusses the application of Style-Based Generative Adversarial Networks (StyleGAN) in the field of interior design, specifically in generating realistic and customizable bedroom designs. Chen Zhang et al. highlight the importance of incorporating local control mechanisms within the GAN framework to allow users to specify specific modifications and personalize the generated images according to their preferences.

By leveraging the power of StyleGANs, which are known for their ability to generate high-quality and diverse images, the researchers propose a system that enables users to actively participate in the design process of their bedrooms. The system takes into account various design elements such as furniture arrangement and color schemes and allows users to interactively modify and customize these aspects.

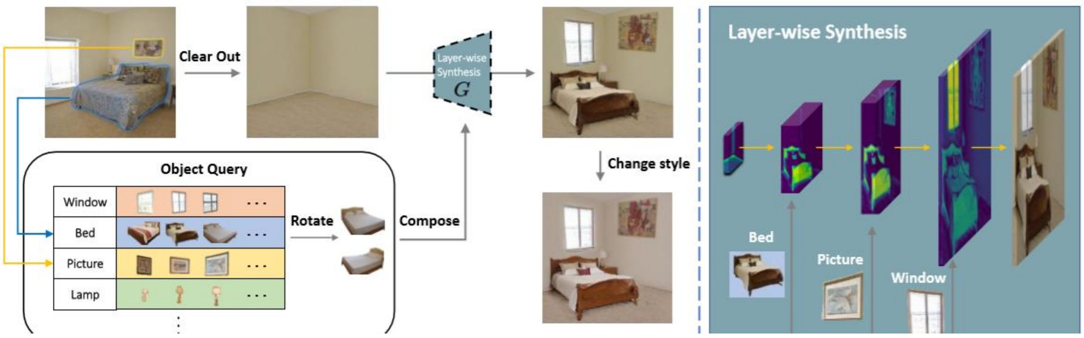


Figure 12: re-decorating bedroom using Logan

The article illustrates the process of re-decorating a bedroom using the LoGAN pipeline (figure 12). It involves clearing the room, selecting objects, Composing the bedroom by inserting each object at the proper position and layer, and rendering the scene with different styles. The figure provides a concise overview of the step-by-step process involved in re-decorating a bedroom using LoGAN.

LoGAN:

Local Control of GANs refers to the ability to manipulate and control specific regions or objects within an image generated by a Generative Adversarial Network (GAN). GANs are deep learning models that consist of a generator network and a discriminator network. The generator network generates synthetic images, while the discriminator network tries to distinguish between real and fake images.

The idea behind Local Control of GANs is to extend the capabilities of GANs beyond generating images as a whole and enable more fine-grained control over the generated content. Instead of modifying the entire image, the focus is on selectively editing and manipulating specific regions or objects within the generated image while preserving the global context and overall image quality.

The concept of Local Control of GANs involves several key components and techniques:

1. Content Modulation: Content modulation allows for the modification or removal of specific objects or regions within an image. It involves identifying the target region or object and applying appropriate techniques to replace or modify the corresponding features in the generated image.

2. Style Modulation: Style modulation enables the control of the visual style or appearance of specific regions or objects in the generated image. It involves manipulating the latent space representation or the style parameters of the generator network to achieve desired changes in the style of localized regions.

3. Priority Mask: The priority mask determines the execution order or importance of different editing operations on overlapping regions or objects. It assigns priorities based on predefined rules or user-defined preferences, ensuring that the desired changes are applied in the correct order and preventing conflicts or inconsistencies.

4. Layer-wise Synthesis: Layer-wise synthesis involves performing editing and manipulation operations at different layers of the generator network. This allows for progressive refinement and control over the generated image, with each layer contributing to the overall synthesis process.

By incorporating these techniques, Local Control of GANs enables applications such as object removal, object insertion, object rotation, and style transfer within localized regions of the generated image. It provides a more interactive and customizable approach to image generation and manipulation, allowing users to have fine-grained control over specific elements of the generated content.

Overall, the concept of Local Control of GANs expands the capabilities of GAN models and opens up possibilities for advanced image editing, synthesis, and customization. It bridges the gap between global image generation and localized content control, enabling more precise and targeted modifications in generated image.

In [3], a technique is proposed to enhance the quality and realism of 3D indoor scenes using guide words as semantic cues. Machine learning techniques are employed to train a model capable of enhancing and understanding the indoor context. The model is trained on diverse 3D indoor scenes and then applied to enhance other scenes. Guide words are utilized to guide the enhancement process, with each word assigned to a specific improvement in the indoor context. The model learns the relationships between guide words and appropriate enhancements. Overall, the technique aims to leverage guide words to enhance 3D indoor scenes, resulting in improved realism and quality.

Figure 13:the impact of frequency and specificity on generating enhancement suggestions for a bedroom using the guide word "girl".

The impact of frequency and specificity on generating enhancement suggestions for a bedroom using the guide word "girl" is demostrated in figure 13. The figure presents three different cases:

a) Only considering frequency: In this case, the suggestions generated are based solely on how frequently certain enhancements related to the guide word "girl" occur. These suggestions may be more general and not specifically tailored to the context of the bedroom.

b) Only considering specificity: In this case, the suggestions generated are focused on the specific characteristics or attributes associated with the guide word "girl". These suggestions may capture the specific details but may not consider how frequently those enhancements occur in the bedroom.

c) Considering both frequency and specificity: Here, the suggestions are generated by taking into account both the frequency and specificity of the enhancements related to the guide word "girl". This approach aims to strike a balance between context-specific enhancements and their overall occurrence patterns in the bedroom.

The figure visually presents the enhancement suggestions for the bedroom under each scenario, allowing for a comparison of the effects of frequency and specificity. By examining the different suggestions, one can understand how the combination of frequency and specificity influences the generated enhancements. This analysis helps in comprehending the significance of considering both aspects in guiding the enhancement process and achieving a more realistic and contextually appropriate 3D indoor scene.

In [30], the concept of 3D object instance re-localization (RIO) is introduced. This concept involves estimating the 6 degrees of freedom (6DoF) poses of objects in a 3D scan taken at a later time compared to a reference scan. The dataset used for this research comprises 1482 RGB-D scans of 478 environments captured at different time intervals. The dataset includes objects that undergo positional changes over time, along with their corresponding 6DoF mappings across the re-scans.

In [31], a computer vision-based approach for interior design is presented. The approach involves object identification and colour assignment using computer vision algorithms. The proposed method is evaluated using a dataset of room images sourced from online platforms. The workflow consists of object identification, colour assignment, and post-processing. The goal is to enhance and expedite the interior design process by leveraging computer vision techniques.

3.2.3 Comparisons

Table 5: scene enhancement related works

Ref.	Year	Dataset	Model	notes	accuracy
[29]	2021	LSUN Bedroom dataset	LoGAN	It demonstrates how GANs can be employed to generate customized bedroom decor images based on specific preferences and design criteria.	discriminator accuracy (Metric that measures how well the discriminator is performing in its classification task)
[3]	2017	Semantic3D	Volumetric Convolutional Autoencoder (VCAE)	it focuses on enhancing semantic 3D indoor scenes using guide words. It proposes a method that leverages linguistic descriptions, referred to as guide words, to improve the visual quality and semantic understanding of 3D indoor scenes. The approach utilizes deep neural networks to learn the relationship between guide words and corresponding scene enhancements.	Not mentioned
[30]	2019	ScanNet	RANSAC.	it introduces RIO, a method for 3D object instance re-localization in dynamic indoor environments.	"Matching accuracy" (accuracy in the matching process between "positive patches" and "negative patches") We evaluate our method against hand-crafted features Matching accuracy of the different methods: RIO-singlescale:62.21 RIO-multiscale :83.98
[31]	2019	The dataset was collected from online sources (No specified dataset name)	the use of computer vision techniques, such as object detection, image segmentation, and style	it discusses computer vision-based room interior design. It presents a method that utilizes computer vision techniques to analyze room images and provide recommendations for interior design elements	Not mentioned

			analysis algorithms (No specified model name)	such as furniture placement, color schemes, and decor choices.	

Chapter 4: System Analysis

4.1 Functional Requirements

<u>The admin should be able to do the following:</u>

1. Login
2. view all the application products
3. Add product
4. Delete product
5. Update product
6. View all the application categories
7. Delete category
8. Add category
9. View all the application users
10. View all the orders
11. View the application statistics
12. Logout

<u>The User should be able to do the following:</u>

1. Signup
2. Login
3. View all the available products
4. View all the available categories
5. View the products of specific category
6. View product details
7. Add item to cart
8. Place order
9. Generate décor for specific floor plan using AI scene generation model
10. Enhance the décor of specific room using AI scene enhancement model
11. Chat with the expert system
12. Logout

4.2 Non-Functional Requirements:

1- Security

2- Efficiency

3- Extensibility

4- Availability

4.3 Use case diagram

Figure 14: use case diagram

4.4 Context Diagram

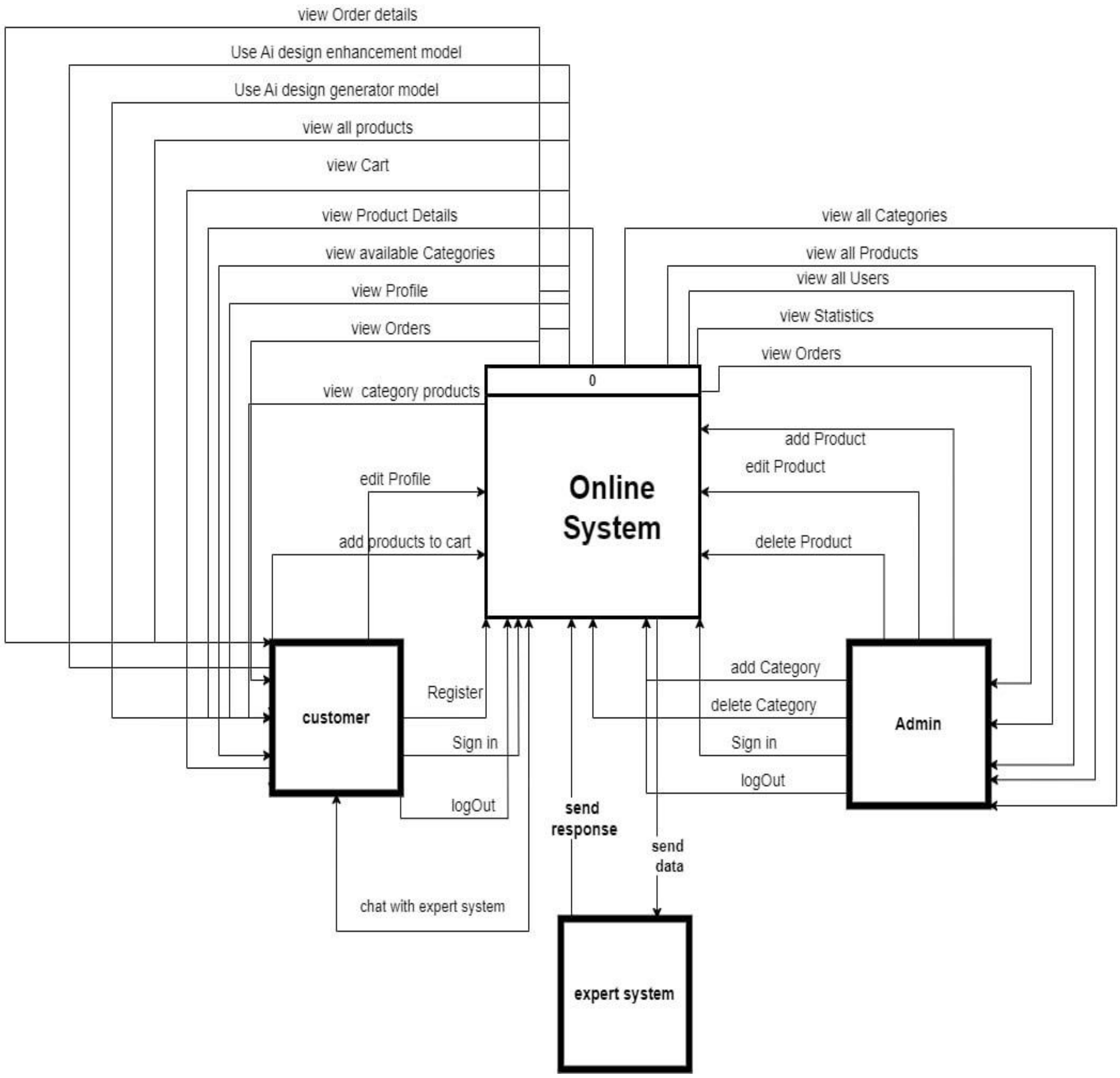


Figure 15: context diagram

4.5 Class diagram

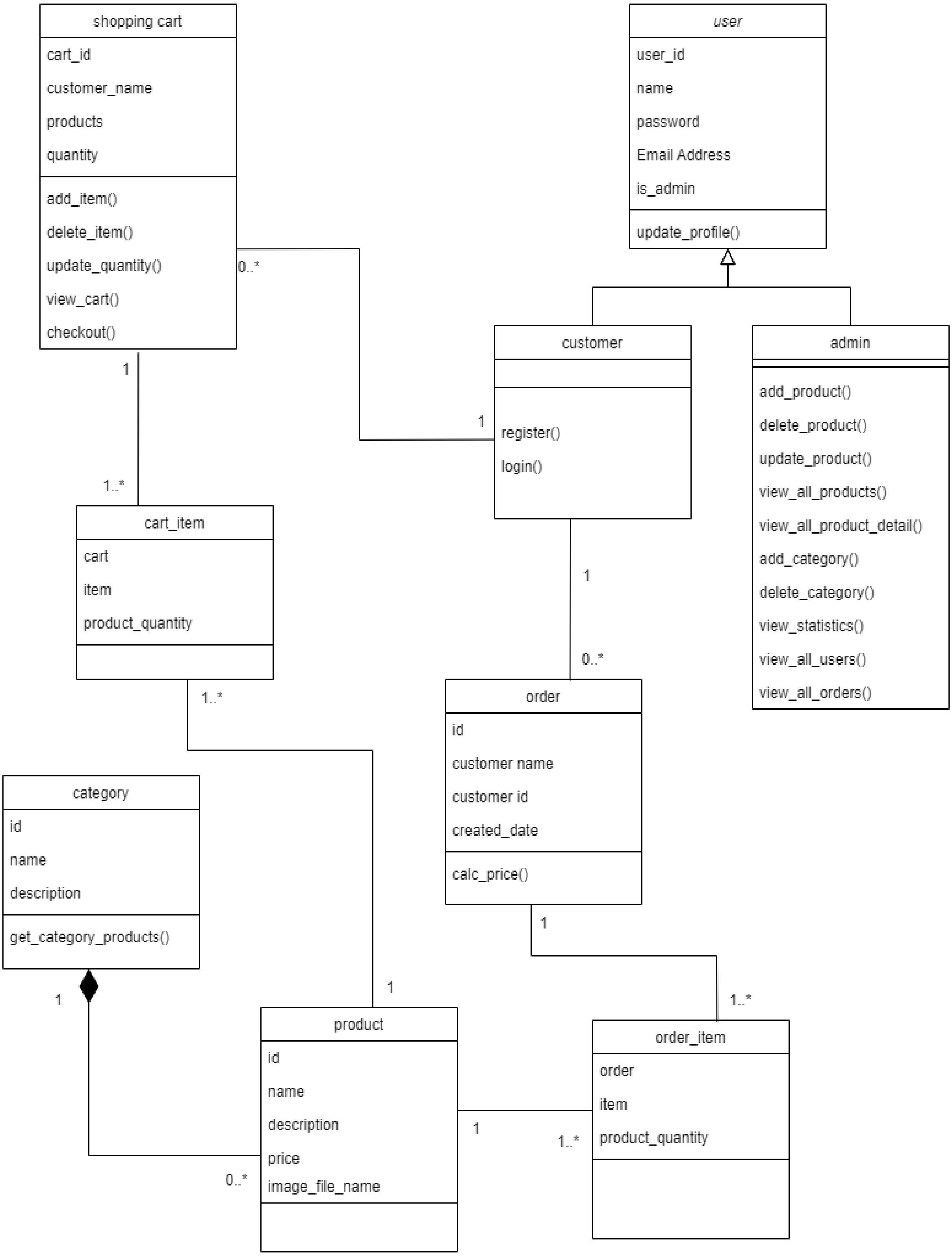


Figure 16: class diagram

Chapter 5: System Design

5.1 ER Diagram

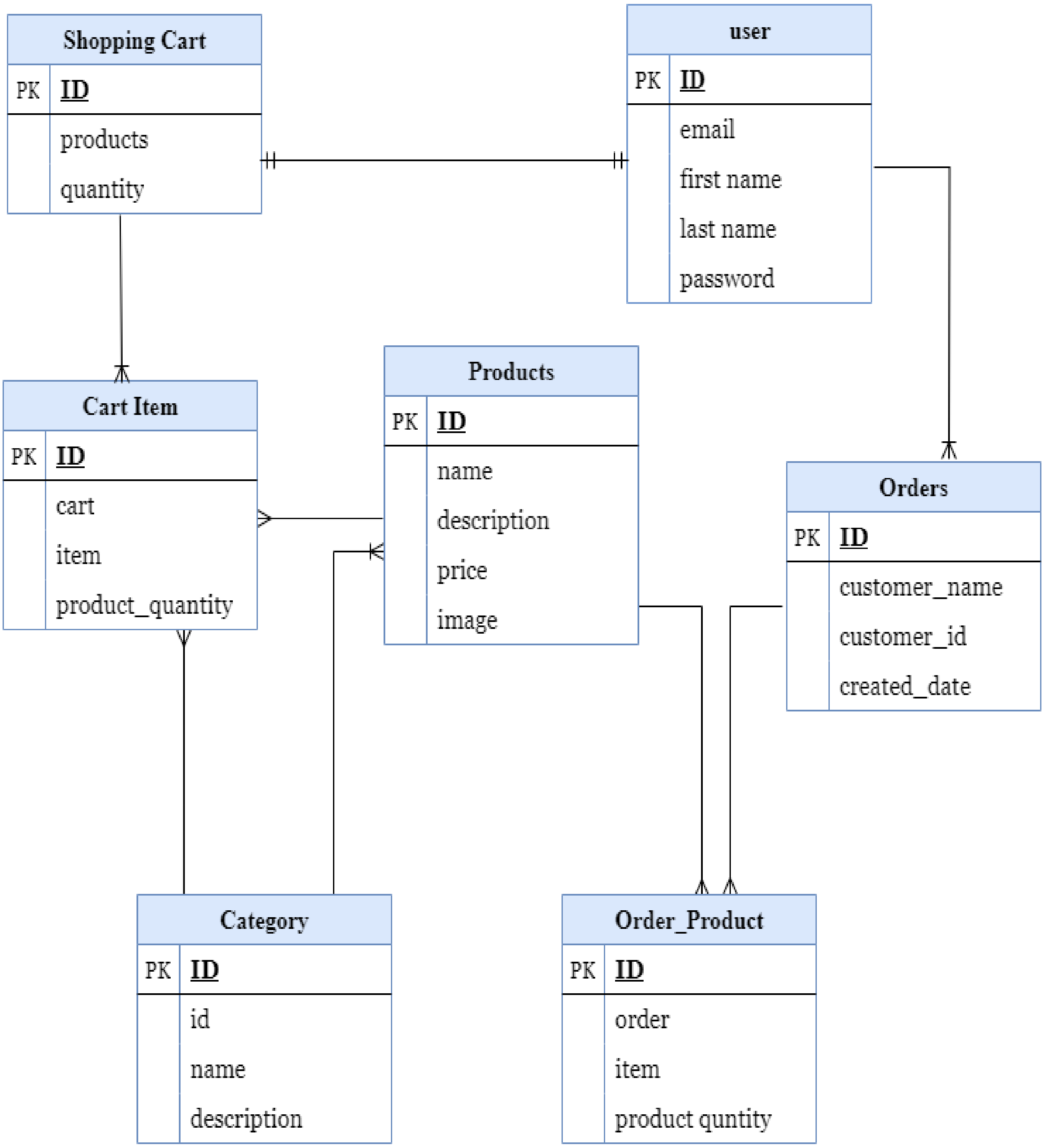


Figure 17: Entity Relation Diagram

5.2 Block diagram

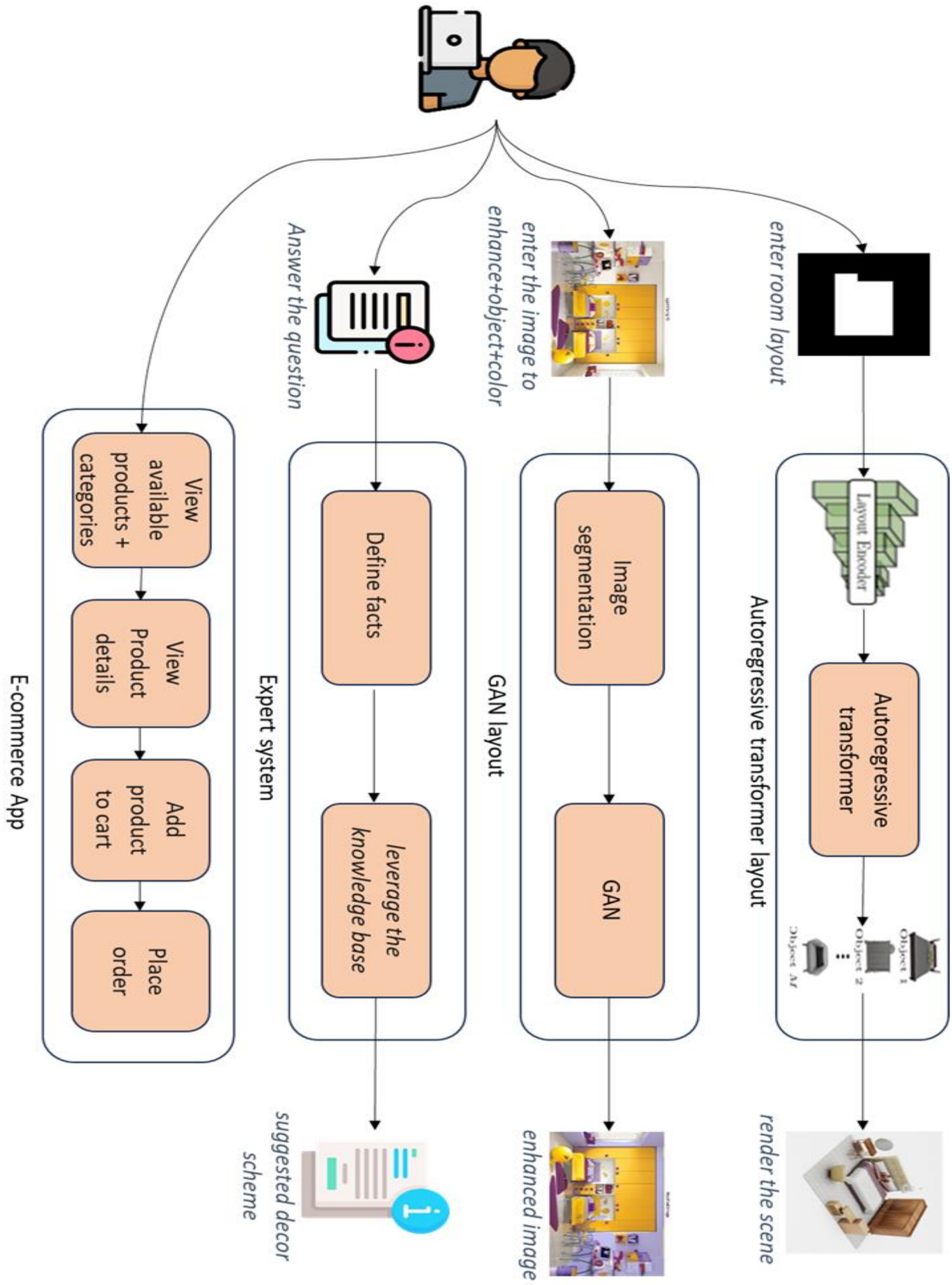


Figure 18: application block diagram

5.3 Scene generation Algorithm

5.3.1 Data Preprocess

5.3.1.1 introduction

In this section, we delve into the intricacies of data preprocessing, outlining the steps undertaken to transform raw data into a clean, structured, and suitable format for further analysis. By addressing data quality issues, handling missing values, and applying necessary transformations, the preprocessing phase sets the stage for reliable insights and robust model performance.

5.3.1.2 Dataset Filtering and Dividing

The 3D-FRONT dataset consists of a collection of 6813 homes with around 14629 planned rooms that are furnished with 3D furniture components. In this experiment, we worked **only on bedrooms because of memory limitations.** According to [32] “3D-FRONT has a number of unnaturally sized rooms, objects that are misclassified, and objects that are in abnormal positions, such as outside the room's bounds, on the floor, overlapping objects, etc. As a result, we had to eliminate troublesome scenes before we could utilize it”

Some previous works on this dataset have filtered the dataset and they have released the names/ids of the filtered rooms so they were dropped from the experiment.

This work divided the preprocessed rooms so that 70% would be utilized for training, 20% for testing, and 10% for validation in order to construct the train, test, and validation splits.

5.3.1.3 parsing the dataset

In order to train and test the model, all scenes from the 3D-FRONT dataset need to be parsed... After going into the structure of the dataset that has been discussed in section 3.1.3.4, we configured that the best way to work with the dataset is to extract all the available rooms and convert them into a convenient format.

In order to process the dataset effectively, a parsing procedure were implemented that systematically extracted relevant information from each scene. This involved iterating through the scenes, which are composed of multiple rooms, and retrieving the furniture and meshes associated with each complete scene.

this work established a mapping between the extracted furniture models and the specific furniture models presented in each room. This allowed us to accurately associate the furniture with their respective rooms.

To avoid redundancy and maintain data consistency, a check was implemented to ensure that duplicate rooms were not included in the dataset. By verifying the uniqueness of each room, the room was added to a set only if it had not been previously encountered.

By executing this refined and systematic procedure, the dataset was parsed efficiently, ensuring that the subsequent analyses and modeling were conducted on a reliable and accurately structured dataset.

5.3.1.4 Rendering the floor plan of the scene

The dataset contains valuable information about the floor plan, which is represented by a combination of points called vertices and interconnected shapes known as faces. These vertices and faces collectively form a mesh, which serves as a blueprint for the floor's structure. To create a 2D floor mask that can be used as input for the network, this work gathers and merges the vertices and faces from multiple "Floor" models the make up the room. This merging process allows us to construct a comprehensive and accurate representation of the entire floor plan, capturing the complete layout of the room. The resulting floor plan serves as essential input for the network, enabling it to effectively analyze and process the spatial characteristics and features of the room.

5.3.2 Autoregressive transformer

5.3.2.1 Overview of the model architecture

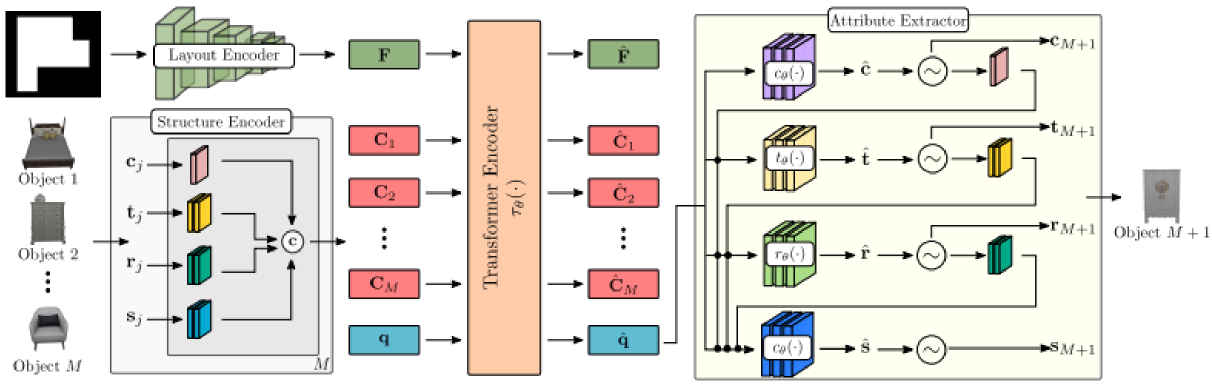


Figure 19: method overview

The model takes a set of scenes as input, where each scene contains 3D labeled bounding boxes representing objects along with their corresponding room shape. The network architecture consists of four key components:

1. Layout Encoder: This component maps the room shape into a global feature representation F. It captures the overall structure and layout of the scene; the layout encoder is implemented with a ResNet-18 architecture

Resent Feature Extractor

ResNet is a popular convolutional neural network architecture that was introduced by Microsoft Research in the paper titled "Deep Residual Learning for Image Recognition." It is widely used for various computer vision tasks, including image classification, object detection, and image segmentation.

ResNet-18 is composed of a total of 18 layers, including convolutional layers, pooling layers, fully connected layers, and skip connections. The key innovation of ResNet-18 lies in the introduction of residual blocks, which address the degradation problem encountered when training deeper neural networks.

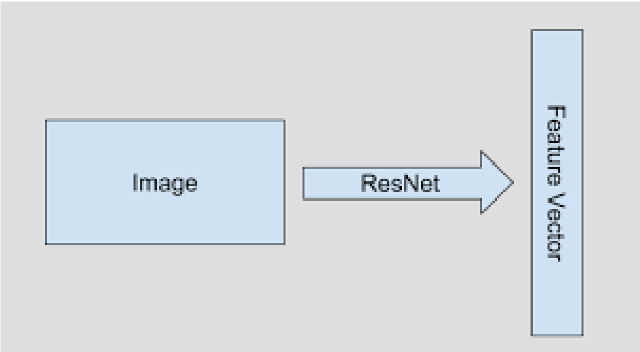


Figure 20:feature extractor

The residual blocks in ResNet-18 allow for the training of deeper networks by utilizing skip connections. These connections enable the flow of information from earlier layers to later layers, allowing the network to learn residual mappings instead of attempting to learn the entire mapping from the input to the output. This mitigates the vanishing gradient problem and helps improve the network's performance.

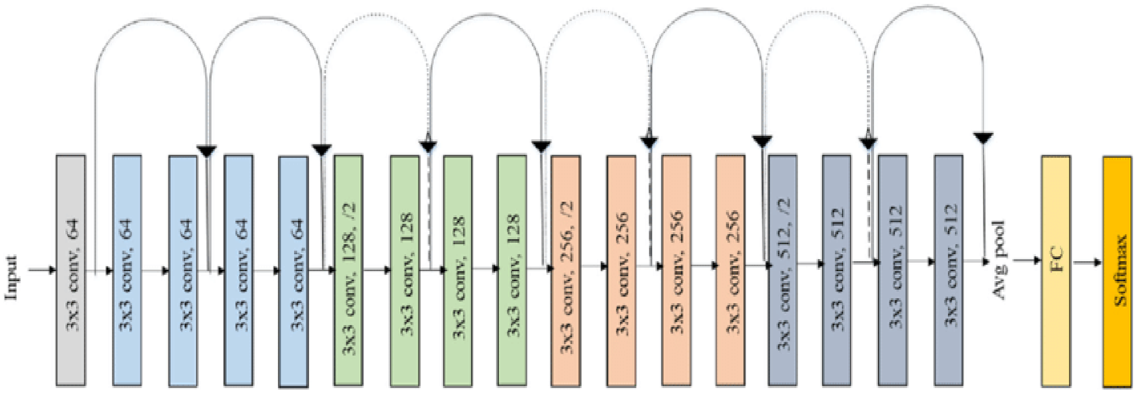


Figure 21: Original-ResNet-18-Architecture

ResNet-18 consists of several residual blocks, each containing two or three convolutional layers, batch normalization, and ReLU activation functions. The network also includes downsampling operations through stride-2 convolutions and average pooling, reducing the spatial dimensions of the feature maps.

The final layers of ResNet-18 typically include global average pooling to aggregate spatial information and a fully connected layer for classification.

ResNet-18 has been pre-trained on large-scale image datasets such as ImageNet, allowing for transfer learning to other related computer vision tasks. By fine-tuning the network or using it as a feature extractor, ResNet-18 has demonstrated strong performance across various image recognition and analysis tasks.

2. Structure Encoder: The structure encoder, denoted as $h\theta$, maps the M objects in the scene to per-object context embeddings. The structure encoder captures the contextual information specific to each object.

3. Transformer Encoder: The transformer encoder, takes inputs from the layout encoder (F), structure encoder (C), and a query embedding (q). It predicts the features $\hat{q}$ for the next object to be generated. The transformer encoder leverages the contextual information from both the global scene layout and individual object embeddings to generate the next object.

4. Attribute Extractor: The attribute extractor is responsible for predicting the attributes of the next object to be generated. It provides additional information about the characteristics or properties of the object.

6.2.2 Structure Encoder

The structure encoder maps the attributes of each object into a per-object context embedding C_j . For the object category c_j , this proposed model use a learnable embedding, which is simply a matrix of size $C \times 64$, that stores a per-object category vector, for all C object categories in the dataset. For the size s_j , the position t_j and the orientation r_j , this work uses the positional encoding of [33] as follows

$$\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p), \ldots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))$$

where p can be any of the size, position or orientation attributes and $\gamma(\cdot)$ is applied separately in each attribute's dimension. In the experiments, L is set to 32. The output of each embedding layer, used to map the category, size, location and orientation in a higher dimensional space, are concatenated into an 512-dimensional feature vector, which is then mapped to the per-object context embedding.

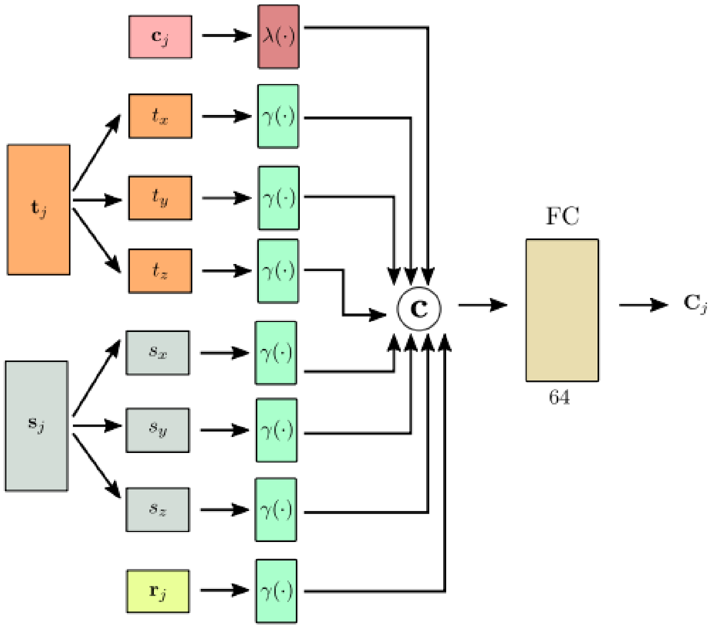


Figure 22: Structure Encoder

5.3.2.3 Transformer Encoder

The Transformer Encoder is implemented as a multi-head attention transformer without positional encoding. The transformer consists of four layers with eight attention heads. The queries, keys, and values in the transformer have dimensions of 64, while the intermediate representations for the MLPs have dimensions of 1024.

To implement the transformer architecture, this project utilizes the transformer library provided by Katharopoulos et al. [10] in the implementation. This library offers convenient functionalities for building and training transformer models.

The input to the transformer is a set of elements represented as

$I = \{F\} \cup \{Cj\}^{M}{}_{j=1} \cup q$, where M represents the number of objects in the scene. F denotes the global feature representation, $\{Cj\}^{M}{}_{j=1}$ represents the per-object context embeddings, and q is a learnable object query vector with a dimension of 64. The object query vector enables the transformer to predict output features $\hat{q} \in R64$, which are used for generating the next object to be added to the scene.

5.3.2.4 Attribute Extractor

The Attribute Extractor plays a crucial the model as it predicts the attributes of the next object to be added to the scene. It operates in an autoregressive manner, meaning it predicts one attribute at a time based on previously predicted attributes.

The prediction process starts with the object category. To predict the object category, this work utilizes a linear layer with 64 hidden dimensions. This layer outputs C class probabilities for each object, where C represents the number of possible object categories.

For the location, orientation, and size attributes, separate MLPs are used. These MLPs predict the mean, variance, and mixing coefficient for K logistic distributions for each attribute. In the experiments, the value of K is set to 10, indicating that the model considers 10 different logistic distributions for each attribute.

To predict the size, location, and orientation attributes, a 2-layer MLP is used with Rectified Linear Unit (ReLU) non-linearities. The hidden layer of this MLP has a size of 128, while the output layer has a size of 64.

By employing these MLPs, the attribute extractor can generate predictions for the attributes of the next object based on the previously generated attributes. The model learns to adjust the parameters of the MLPs during training, allowing it to make accurate attribute predictions while considering the variability and complexity of the scene.

5.3.3 Object Retrieval

During the process of generating scenes, this work employs the 3D-FUTURE dataset to select 3D models that match the predicted category, location, orientation, and size of objects in the scene. To accomplish this, this project perform a nearest neighbor search within the 3D-FUTURE dataset to find the closest matching model in terms of object dimensions.

It is important to note that this approach employs a simple object retrieval strategy, which consistently produces visually plausible room configurations. However, more sophisticated and advanced object retrieval schemes can be explored in future research to further enhance the selection process.

5.3.4 Loss function calculation

During the calculation of the loss value, this project employs a component-wise approach that considers each element of the scene individually. For the class label component, this work utilizes the cross-entropy loss, which measures the dissimilarity between the predicted class probabilities and the true class labels.

In terms of the translation, sizes, and angles components, the work computes the logistic mixture likelihood (dmll). This involves evaluating the likelihood of the target pixel values under the predicted distribution parameters, which include the mean, variance, and mixing coefficients of the logistic distributions.

The loss function takes into account the predicted parameters of the mixtures of logistic distributions and calculates the negative log-likelihood of the target pixel values. This negative log-likelihood quantifies the discrepancy between the predicted distribution and the actual target values.

By incorporating these loss calculations for each component, the model obtains a comprehensive loss value that reflects the dissimilarity between the predicted scene elements and their corresponding ground truth values. This approach allows us to effectively optimize the model by minimizing the overall loss during training.

5.3.5 Training overview

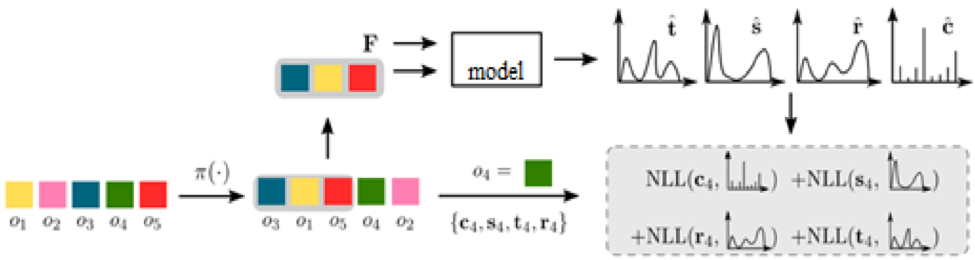


Figure 23: training overview

In the experiment setup, this work begins with a scene containing M objects represented as colored squares. To create a training scenario, we randomly permute the objects and select the first T objects from the permuted sequence, where T represents the number of objects to be considered (in this example, T = 3).

The network is then trained to predict the next object that should be added to the scene, given the subset of objects that have been selected (highlighted in grey) and the floor layout feature F.

To evaluate the accuracy of the predictions, this work employs a loss function known as the negative log-likelihood (NLL). Specifically, calculate the NLL by comparing the predicted attributes of the next object (represented by the green square in the permuted sequence) with the ground truth attributes of that object.

By optimizing the network to minimize the NLL loss, this work aims to enhance the model's ability to accurately predict the attributes of the next object in the scene sequence. This training process allows the network to learn patterns and relationships between objects, enabling it to generate more realistic and coherent scene layouts.

5.4 Scene enhancement Algorithm

5.4.1 Data Preprocess

To preprocess the dataset and improve bedroom decor, this work utilized the **Transforms module**, a commonly used tool in image processing. Here's a summary of the process:

1. Data loading: loading the dataset containing bedroom images, ensuring an adequate number of images for effective training and validation.

2. Resizing: The transforms.Resize() function was employed to resize the images. By specifying the desired output size, the project achieved consistency in dimensions within the dataset. This step facilitated subsequent processing and ensured uniformity.

3. Cropping: In certain scenarios, focusing on specific regions of the bedroom images is essential. For this purpose, this work utilized the transforms.CenterCrop() function. It allowed us to crop the images to a specific size while preserving the central region, eliminating irrelevant or distracting elements and emphasizing key areas.

4. Converting to tensors: After resizing and potential center cropping, this project applied the transforms.ToTensor() function to convert the processed images into PyTorch tensors.

5. Normalization: To further enhance the convergence and stability of the model during training, this work performed normalization on the images after converting them to tensors.

By incorporating these preprocessing steps, including resizing, center cropping, converting to tensors, and normalization, this work effectively processed the dataset, making it compatible with deep learning frameworks like PyTorch.

Lastly, split the pre-processed dataset into training and validation sets. The training set was used to train the enhancement model, while the validation set allowed us to evaluate the model's performance and fine-tune parameters. This splitting process ensured that the model could generalize well to unseen data.

By leveraging the transforms module in the preprocessing step, this work successfully resized, cropped, converted, and normalized the images, preparing them for training and enhancing bedroom decor.

5.4.2 LoGAN Model

The LoGAN model empowers users to visualize and explore different color options for objects within a room. By providing an image of the room as input, the template provides users with the flexibility to choose the object whose color they would like to modify. This functionality enables users to make informed decisions about color customization, ultimately saving time and resources while achieving their desired aesthetic outcomes.

Local Control GAN is a model that consists of two main networks: the Generator and the Discriminator.

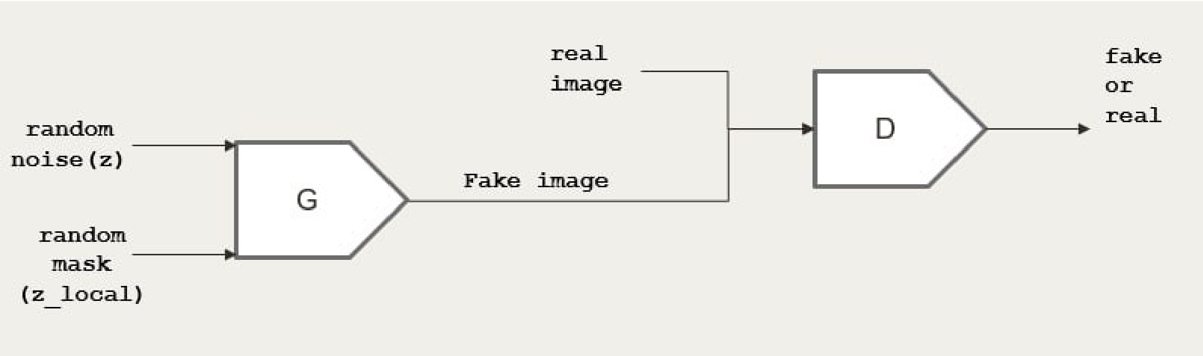


Figure 24: LoGan

5.4.2.1 Generator network

The Generator network comprises the General Generator, the Local Generator and an Attention Layer.

5.4.2.1.1 General Generator

The General Generator is responsible for generating an overall enhanced version of the input image. The General Generator typically consists of transpose convolution layers that increase the spatial dimensions of the image. It applies various transformations such as padding, batch normalization, and activation functions to generate the globally enhanced image.

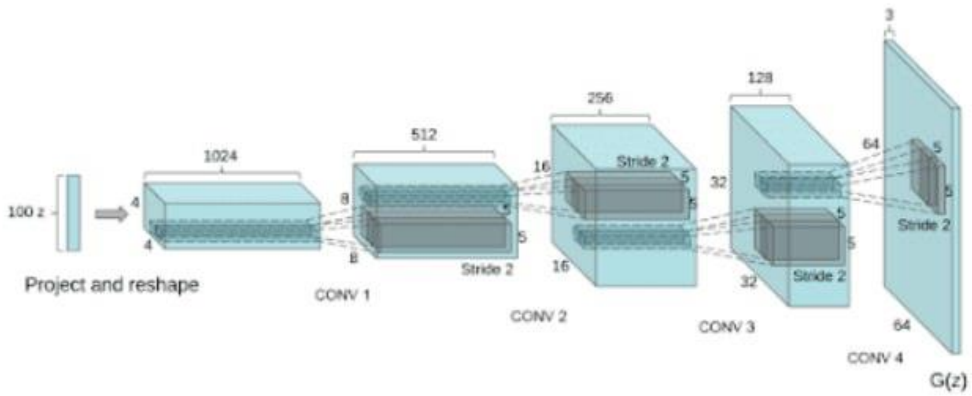


Figure 25: general generator

5.4.2.1.2 Local Generator

The Local Generator focuses on improving a specific object or region within the room image. It takes a local latent vector as input, which contains specific information about the targeted object for enhancement.

The Local Generator utilizes the local information to modify the desired object while keeping the rest of the image unchanged. It allows users to experiment with different color options for the selected object.

5.4.2.1.3 Attention Layer

The Attention Layer is a component within the Local Control GAN that helps the model focus on relevant regions of the image. It takes the concatenated input of the global and local latent vectors and applies convolutional operations. The attention mechanism in this layer helps the model to assign different weights to different spatial locations, emphasizing important areas. By using the attention layer, the model can selectively enhance the targeted object while paying attention to its specific details.

5.4.2.2 Discriminator network

The Discriminator network plays a crucial role in training the Generator networks. It is responsible for differentiating between real images and generated images. the Generator and Discriminator networks work together to enhance the room images, ensuring that the generated images closely resemble real ones.

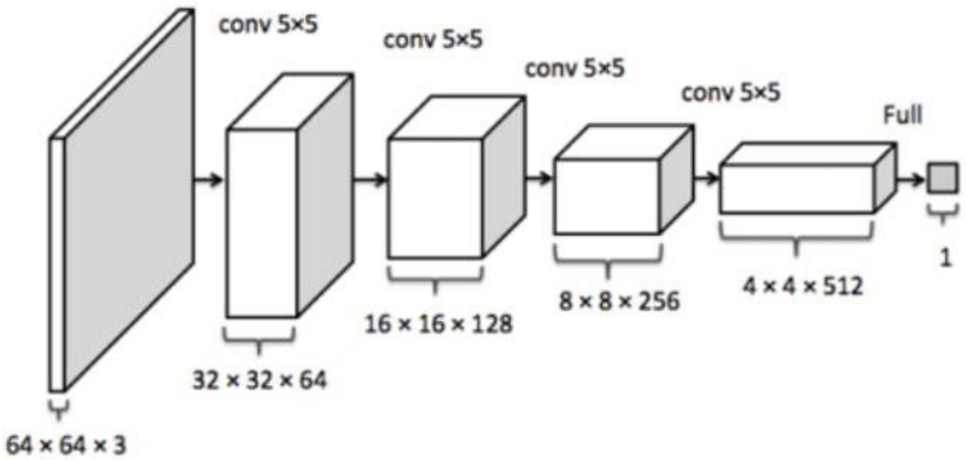


Figure 26: Discriminator

5.4.3 Training

The `LoGAN` class is defined as a subclass of `nn.Module` and takes two arguments: `netG` (the generator network) and `netD` (the discriminator network).

The `forward` method is overridden to define the forward pass of the LoGAN model. It takes an input `x`, which represents the room image. Inside the method, random noise `z` is generated, and `z_local` is set to None. The generator (`netG`) is then used to generate images based on the noise and local information. The generated images are passed through the discriminator (`netD`), and both the noise `z` and the discriminator output `d` are returned.

The `train` method is defined to train the LoGAN model. It takes `train_loader` (the data loader for training data), `valid_loader` (optional data loader for validation data), and `epochs` (the number of training epochs) as arguments.

Inside the `train` method, the code checks if there is a saved checkpoint file for the model. If available, the checkpoint is loaded, including the model weights and optimizer states, to resume training from the last saved epoch. Otherwise, training is started from epoch 0.

The code then enters a loop over the epochs. Within each epoch, it iterates over the batches of training data from the `train_loader`.

For each batch, the discriminator (`netD`) is trained first. Real images from the batch (`x`) are passed through the discriminator, and the output is compared with the corresponding labels using a loss function (`criterion`). The gradients are calculated and the discriminator parameters are updated (`optimizerD.step()`).

Next, the generator (`netG`) is trained. Random noise (`noise`) is generated and passed through the generator to generate fake images. These fake images are then passed through the discriminator, and the output is compared with the real labels. The gradients are calculated and the generator parameters are updated (`optimizerG.step()`).

Throughout the training process, the losses for the generator and discriminator are recorded in lists (`G_losses` and `D_losses`). Additionally, the discriminator accuracy is calculated and recorded in a list (`accuracies`).

At the end of each epoch, the current accuracy is printed, and a grid of generated images is saved for visualization purposes.

Finally, a checkpoint is saved, including the current epoch, model weights, optimizer states, and losses

5.4.4 Loss function

5.4.4.1criterion = nn.BCELoss()

When using criterion = nn.BCELoss() in the context of Local control GAN for improving bedroom decor, this work employs this loss function to train the GAN model.

The BCELoss, or Binary Cross Entropy Loss, is commonly used in binary classification tasks, where the objective is to classify inputs into one of two classes. In the context of GANs, the BCELoss measures the dissimilarity between the predicted outputs and the target labels, which are binary (real or fake).

In the case of Local control GAN for improving bedroom decor, the BCELoss is utilized to guide the training of the discriminator network. The discriminator is responsible for distinguishing between real bedroom decor images and the generated images produced by the generator network. By comparing the discriminator's predictions with the ground truth labels (real or fake), the BCELoss computes the loss, which represents the discrepancy between the predicted and target classifications.

During the training process, the BCELoss is used to update the discriminator's weights by backpropagating the loss and optimizing the network parameters. This enables the discriminator to learn to differentiate between real and generated bedroom decor images more accurately.

In summary, by utilizing the BCELoss in Local control GAN for improving bedroom decor, the project ensures that the discriminator network is trained effectively to distinguish between real and generated images, leading to better-quality generated decor samples.

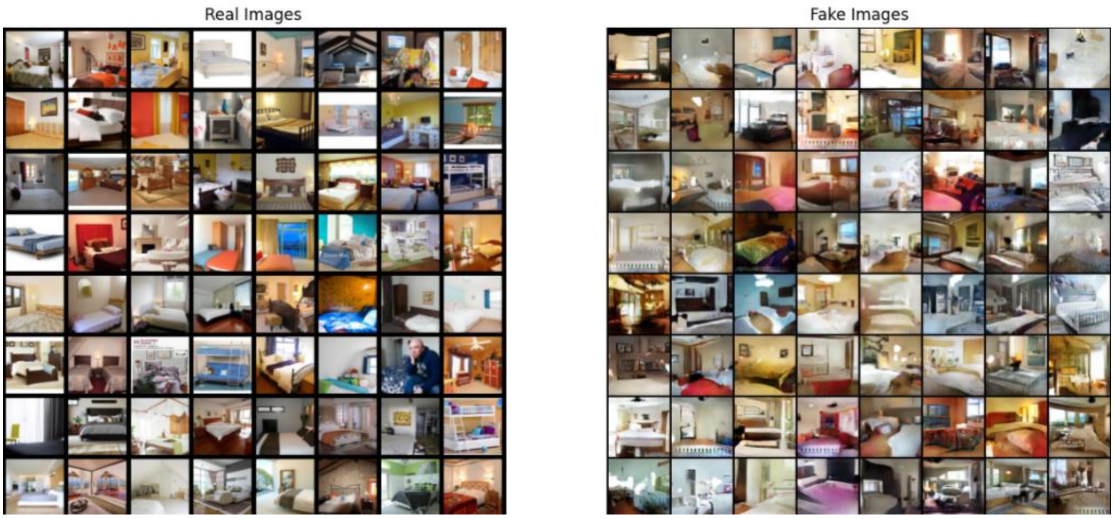


Figure 27: real vs fake images

5.4.4.2 wasserstein_loss

When applying the ` wasserstein_loss ` function with Local control GAN for improving bedroom decor, they were found to be unsuitable due to the following reasons:

The `wasserstein_loss` function calculates the loss based on the Wasserstein distance.

The Wasserstein loss is commonly used in Wasserstein GAN (WGAN) models, which have specific theoretical properties related to the stability of GAN training.

In summary, while the `wasserstein_loss` functions have their uses in certain GAN architectures, they might not be the most suitable choices for improving bedroom decor.

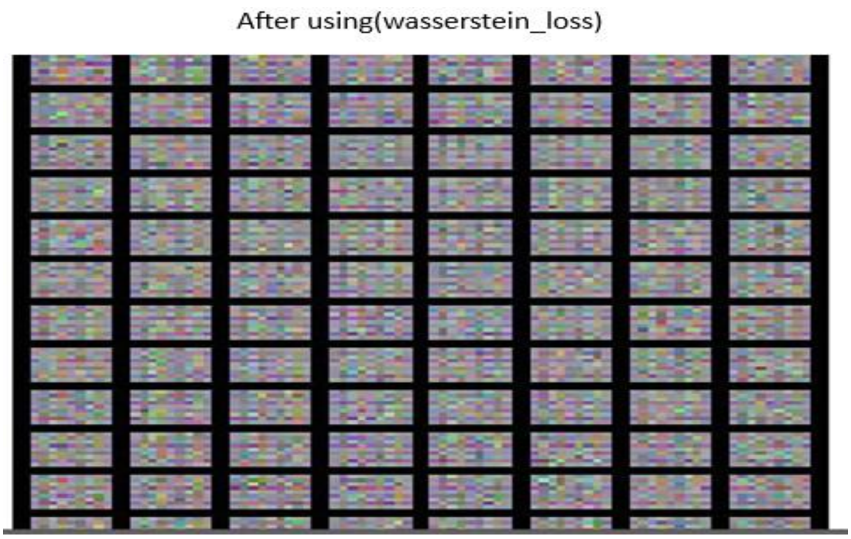


Figure 28: wasserstein loss

5.4.5 Semantic Segmentation

Semantic Segmentation involves dividing an image into meaningful parts and assigning each pixel a specific semantic class, such as walls, floors, furniture, or windows. This allows for precise identification and classification of different objects and regions within the image.

In our project, we utilized a pre-trained model called "nvidia/segformer-b0-finetuned-ade-512-512." This model, developed by NVIDIA, is specifically designed for the task of semantic segmentation.

The "nvidia/segformer-b0-finetuned-ade-512-512" model has been trained on the ADE20K dataset, which is a widely used dataset for image recognition and segmentation tasks. The ADE20K dataset consists of diverse images from different scenes and contains annotations for various semantic classes, allowing the model to learn to accurately segment different objects and regions in an image.

5.5 Expert System

5.5.1 introduction

The expert system integrated into the app offers an innovative feature that assists users in decorating their rooms according to their personalized preferences, even in the absence of a reference picture that is necessary to use the previous AI models. This system engages users in a series of targeted questions to gather specific details about their room type, budget, available space…. etc. By considering these factors, the expert system generates a comprehensive design scheme that caters to the user's unique requirements.

In order to get the accurate information and design schemes according to each situation, all the needed data was obtained from a décor expert.

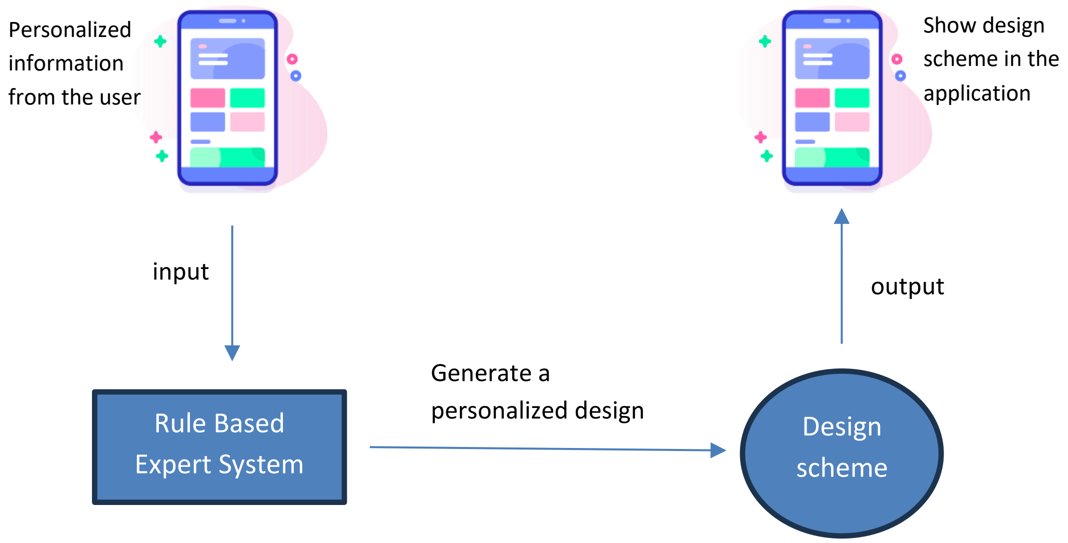


Figure 29: expert system block diagram

5.5.2 decision tree

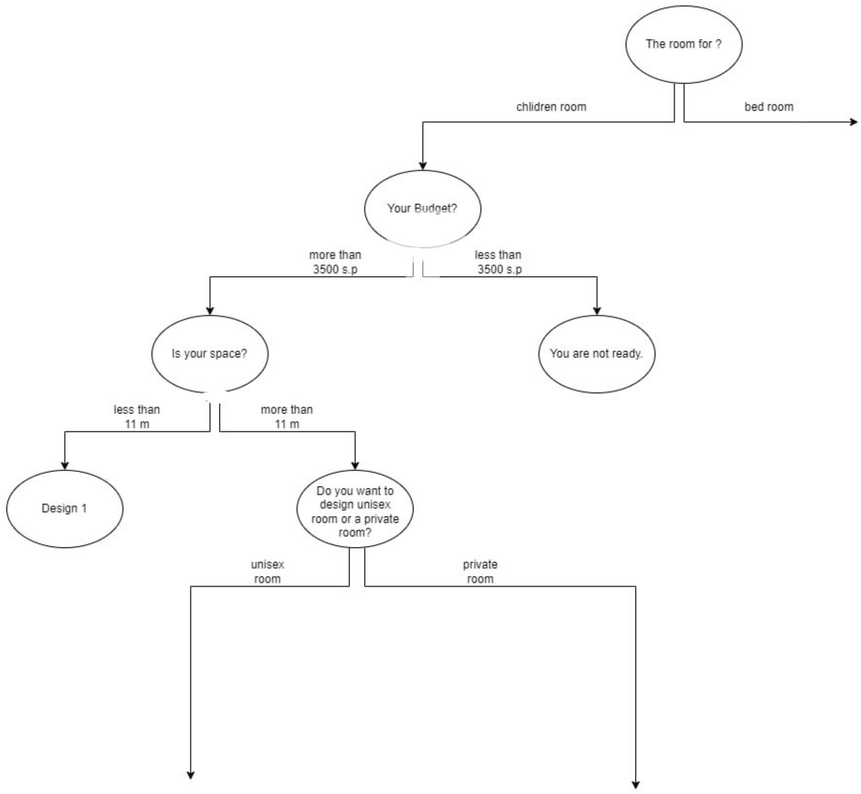


Figure 30: decision tree -1

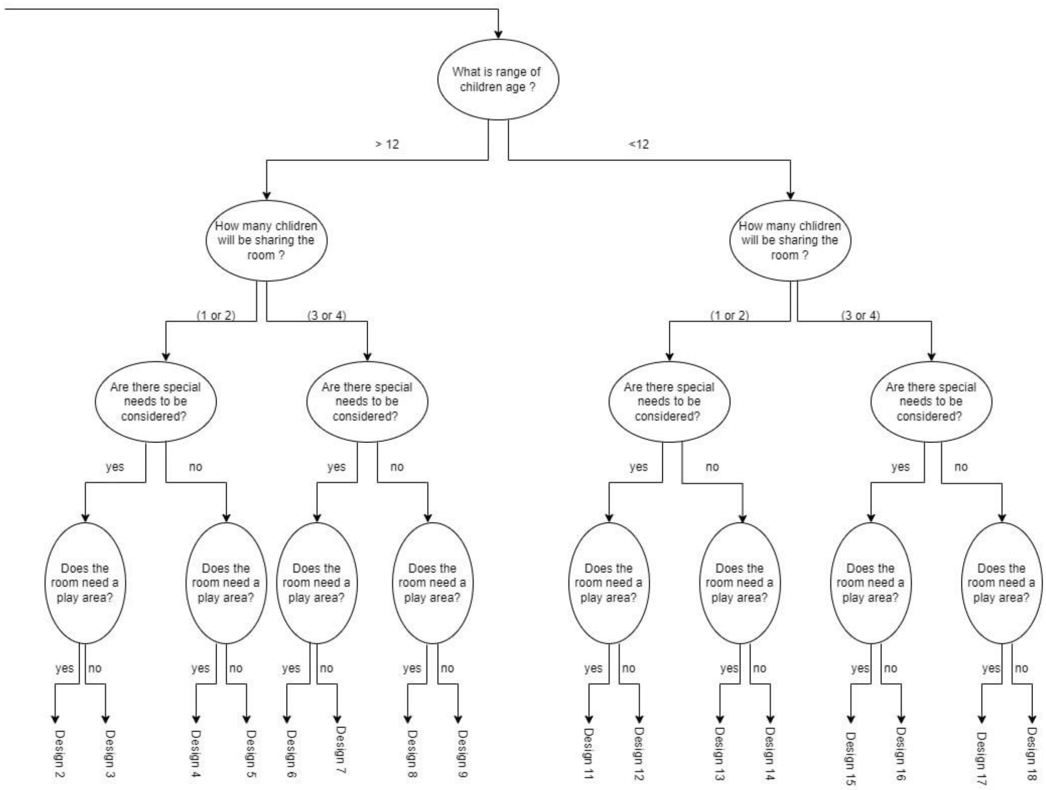


Figure 31: decision tree -2 unisex room

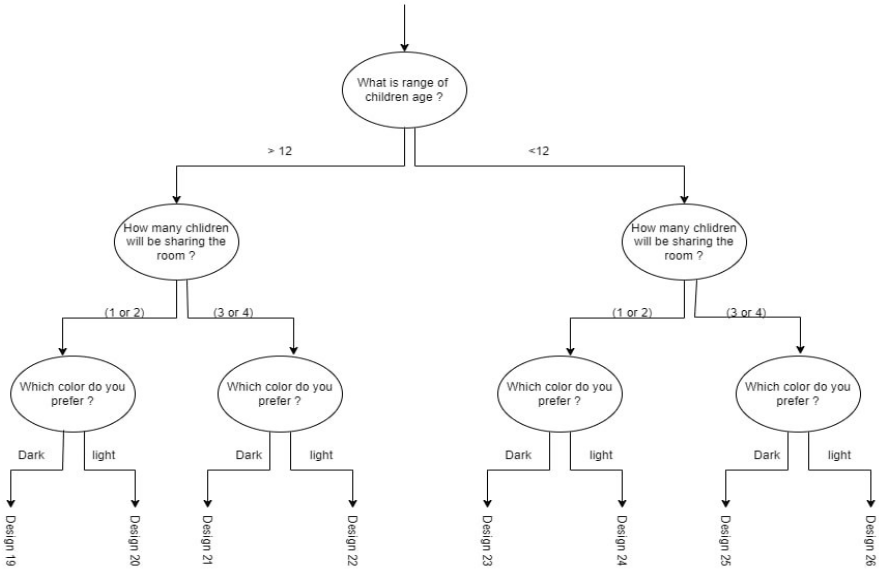


Figure 32: decision tree-3 private room

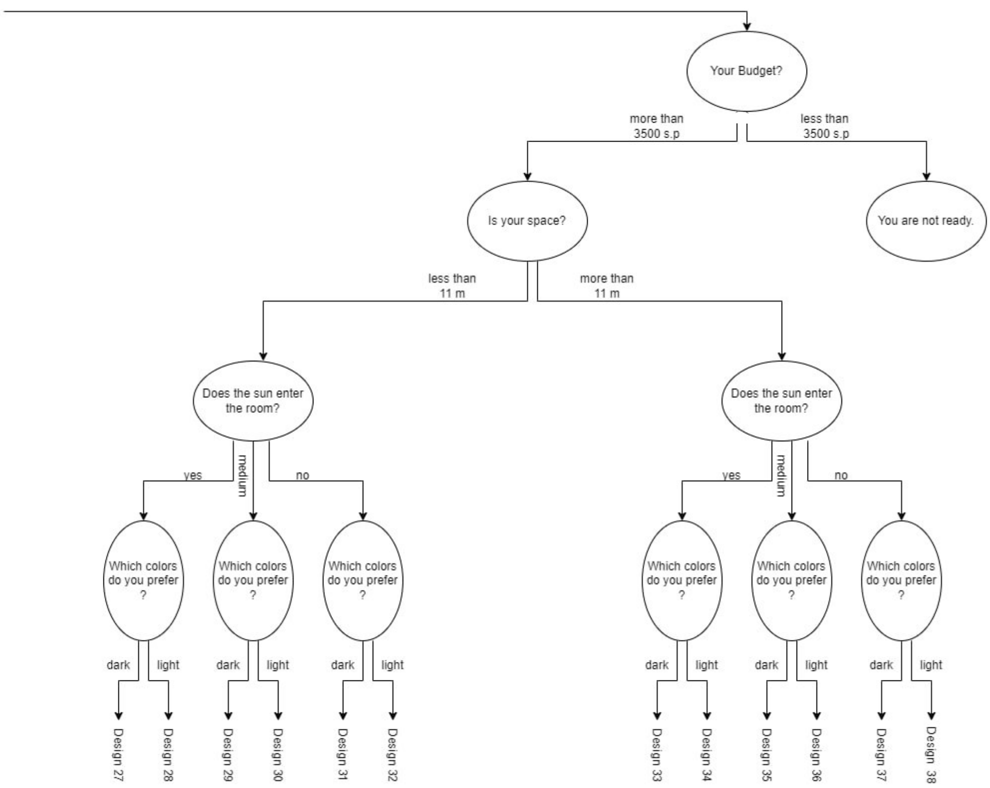


Figure 33: decision tree 4-bedroom

5.6 Features Integration with the application

To fully utilize all the features of the application, a local HTTP server was implemented to host the AI capabilities. The primary server was made accessible locally, and the mobile application was configured to make requests to the local IP address of the main server. This setup allowed the application to retrieve and interpret results from the server, enabling seamless integration of AI features for the client.

The mobile application, specifically designed to communicate with the local server, leverages this connection to request and receive the desired AI-driven functionalities. The server processes these requests, employing advanced algorithms and models to generate accurate and meaningful results. The interpreted results are then transmitted back to the client application for presentation and utilization by the user.

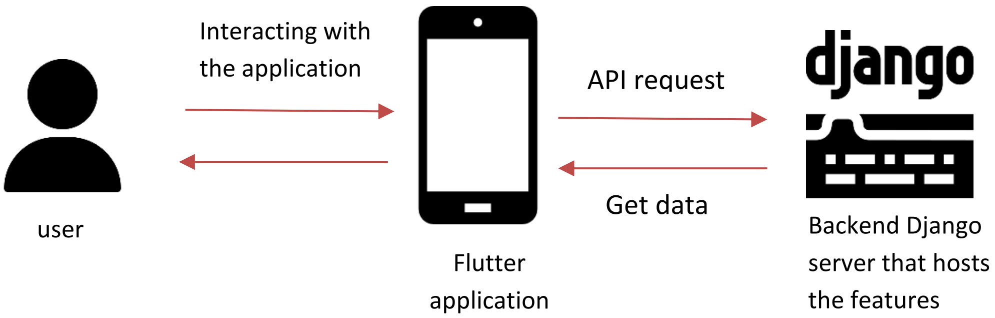


Figure 34: features integration

Chapter6: Results and discussion

8.1 scene generation evaluation

8.1.1 evaluation

Dataset: The model was trained using the 3D-FRONT dataset [21], which comprises a vast collection of 6,813 houses containing approximately 14,629 rooms. These rooms are populated with 3D furniture objects sourced from the 3D-FUTURE dataset [22]. For the purpose of model evaluation, this work specifically focused on bedrooms. By applying preprocessing techniques such as filtering out uncommon object arrangements and rooms with unnatural sizes, this project was able to obtain a subset of 5,996 bedrooms.

When considering the bedrooms, this work utilized 21 distinct object categories. For further details on the preprocessing steps undertaken, please refer to chapter 5.3.1.

Evaluation Metrics:

Scene classification accuracy: to measure the realism of the generated scenes, this work follows prior work [2] and report the classification accuracy of a classifier trained to discriminate real from synthetic scenes.

Table 6: scene classification accuracy

Scene classification accuracy			
reference	[22]	[2]	**DecorMate**
accuracy	0.76	0.68	0.56

KL divergence: (Kullback-Leibler divergence), is a measure of the difference between two probability distributions.

Table 7: Category KL Divergence

Category KL Divergence			
reference	[22]	[2]	**DecorMate**
accuracy	0.0064	0.0052	0.0066

Network parameters comparison: the number of trained parameters comparison of suggested model and [[22], [2]]

Table 8: number of trained parameters

Number of trained parameters in millions			
reference	[22]	[2]	**DecorMate**
Number of trained parameters	38.180	129.298	36.053

Generation Time Comparison: time measure (ms) to generate a scene, conditioned on a floor plan.

Table 9: generation time

Generation time (ms)			
reference	[22]	[2]	**DecorMate**
Generation time	13193.77	849.37	102.38

8.1.2 Results:

Figure 35: scene generation results

8.2 Scene enhancement evaluation

8.2.1 Evaluation

Our model has been trained on the LSUN-bedroom dataset, which includes 303,125-bedroom images. The primary focus of our evaluation was specifically on bedrooms. To measure the accuracy, we employed the discriminator network. A lower accuracy score indicates better performance, indicating that the generator network is generating images that are closer to real images. By utilizing the discriminator accuracy, we can assess how well the generated images resemble real bedroom images in terms of their realism and quality.

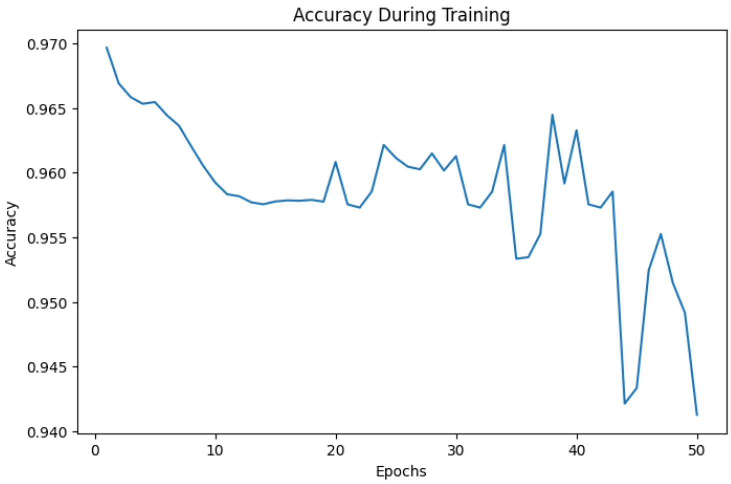


Figure 36: discriminator accuracy

we used loss functions for both the generator and discriminator networks. The loss function helps measure the discrepancy between the generated images and the real images, allowing us to optimize the networks' performance.

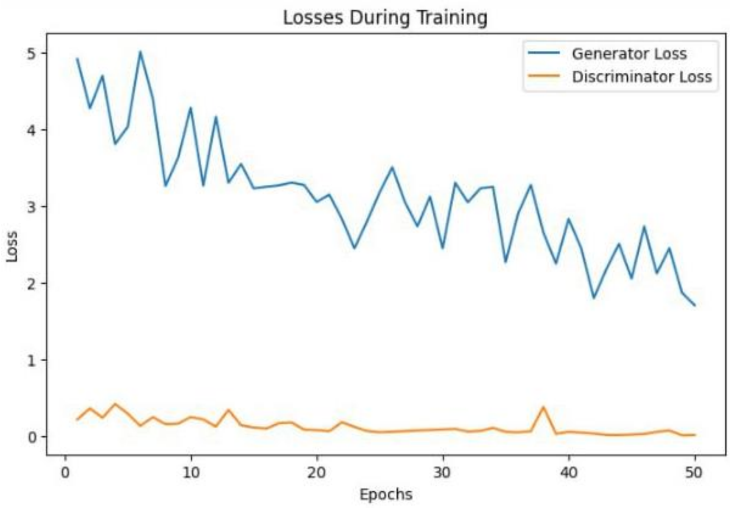
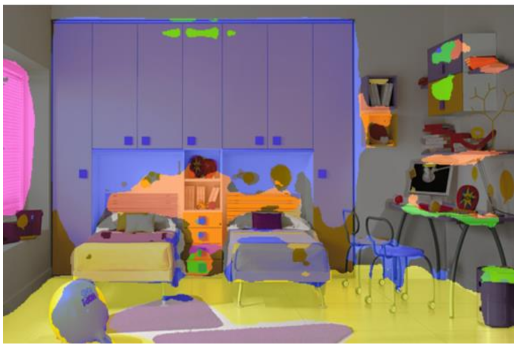


Figure 37: loss during training

8.2.2 results

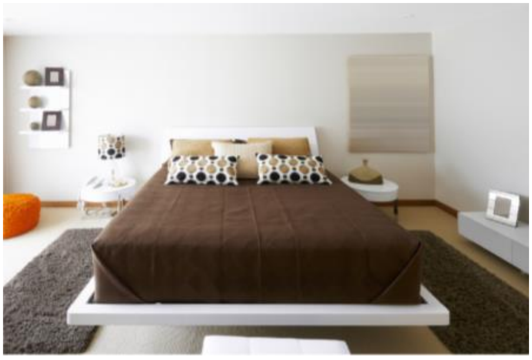
A- original image



B- image after segmentation

C- image after changing the floor color

Figure 38: scene enhancement result-1



A- original image

B- image after segmentation

C- image after changing the wall color

D- image after changing the bed color

Figure 39: scene enhancement result-2

8.3 Expert system

8.3.1 inputs

The system will ask the user about:

1. Room type
2. Budget
3. Available space
4. Degree of illumination
5. Preferred colors
6. Number of children (children room)
7. Play area (children room)
8. Shared or private room

8.3.2 Scenario examples

Scenario 1:

Enter the room type? 1- bed room 2-Children room?

1

Is your budget: 1-more than 3500 2-less than 3500?

1

Is your space 1- <11 m 2- >11 m?

1

Does the sun enter the room? 1- yes 2-no?

1

Which colors do you prefer? 1- Dark colors 2-Bright colors?

1

Result:

```
"Since the space available is small,"
"The room should be coordinated and simple in design, little decoration and
use of cold colors,"
" Not hot colors such as brown and beige tones, paying attention to simply
distributing the furniture and reducing the elements in the room the bed
should be 2m * 1.8 m, with a fixed depth of 60 cm and the length is medium."
 " It is preferable to choose a long curtain for the floor to give a sense of
calm and containment within the light - A colorful bedroom so as not to
completely block the light with thick fabric. In addition to choosing a light
```

```
carpet color to give comfort in the room, as the colors chosen for the
furniture are dark. "
 "As for the lighting, it should be medium, because the room has not strong
sunlight."
```

Scenario 2:

Enter the room type? 1- bed room 2-Children room?

2

Is your budget: 1-more than 3500 2-less than 3500?

1

Is your space 1- <11 m 2- >11 m?

1

Result:

```
"Since you don't have enough space, we are going to suggest simple decor:"
 "Tables:  one small size and 2 small wooden legged table \n"
"Lightning: crystal chandelier, 1 large lamps in the corner and 4 glow-lamps 2
on each wall\n"
 "Curtains: window-scarf curtains \n"
 "Carpet: one small size carpet \n"
 "Pattern: small wall panels \n "
"Masterpieces: small size masterpieces e.g.: vases, crystals, golden
clock...\n"
 "Notes:  suggest you to add a (gilt/vintage/wooden) mirror and console.\n"
 "Since you have chosen dark colors and sun don't enter the room or enters it
rarely, we tried to give you"
  "Our best suggestion even it may will not fit in the room due to space \n")
```

Scenario 3:

Enter the room type? 1- bed room 2-Children room?

2

Is your budget: 1-more than 3500 2-less than 3500?

1

Is your space 1- <11 m 2- >11 m?

2

Do you want to design a unisex room or a private room for each gender? 1-unisex 2-private?

1

What range of children's age? 1- >12 2- <12?

1

How many children will be sharing the room? 1- (1 or 2) 2- (3 or 4)?

2

Are there special needs to be considered? Like allergies disabilities?

1-yes 2-no?

1

Does the room need a play area?1-yes 2-no?

1

Result:

```
"The suggested furniture will be as follow \n"
"Since children are over 12 years old, it is better to have a small office
that matches the size of the room and the space is big you should choose a
medium-sized bed that it is not placed close to the windows and avoiding that
the bed has sharp corners or flammable materials."
"It is preferable to choose covers, bedspreads and curtains from natural
fabrics and their colors should be of light colors that do not reflect the
sun's rays, such as baby pink, baby_blue,white and yellow...Pictured is an
approximate example"
 "Notes: we didn’t add any harmful pieces in order to make it a safe place for
your child"
 "we can put small area to put toy and play in it"
 "TV will be putted on wall since there is no much space for that we didn't
suggest TV table\n"
```

8.4 Application screens

8.4.1 Welcoming screens

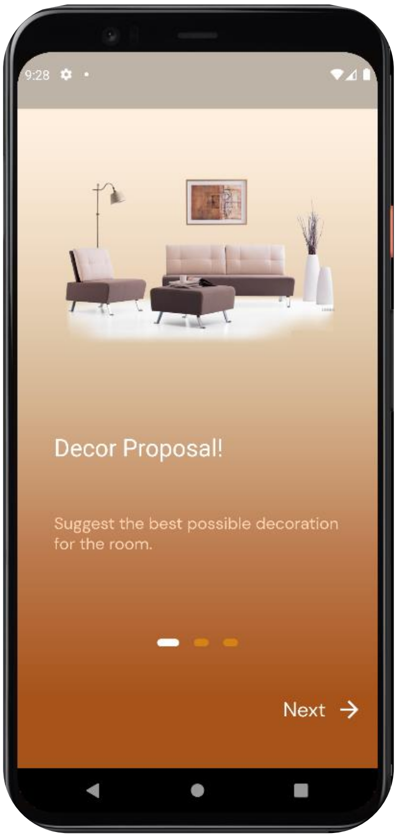


Figure 41: welcome screen 1

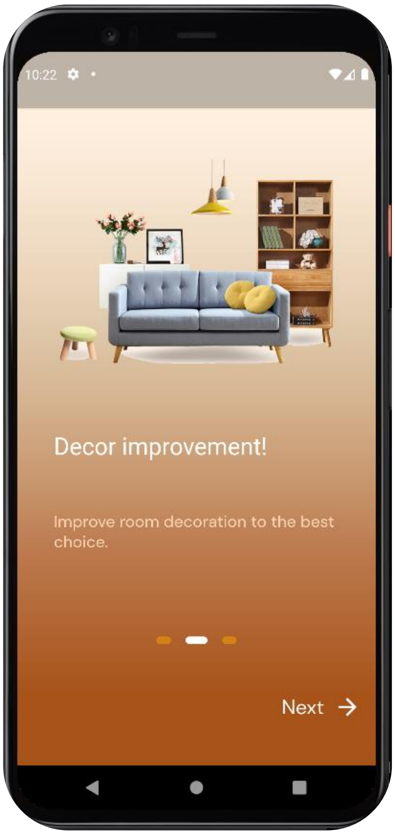


Figure 40: welcome screen 3

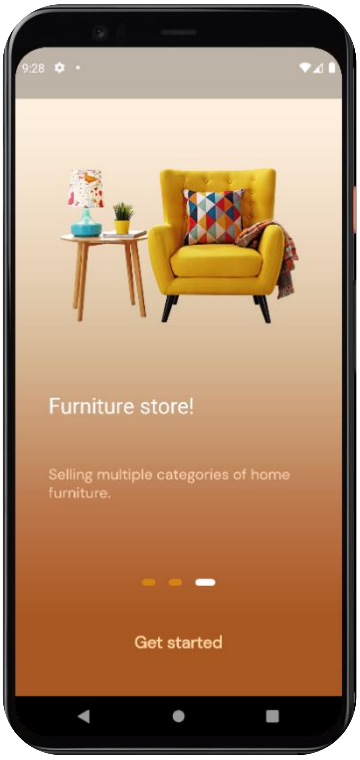


Figure 42: welcome screen 2

Figure 43: entry screen

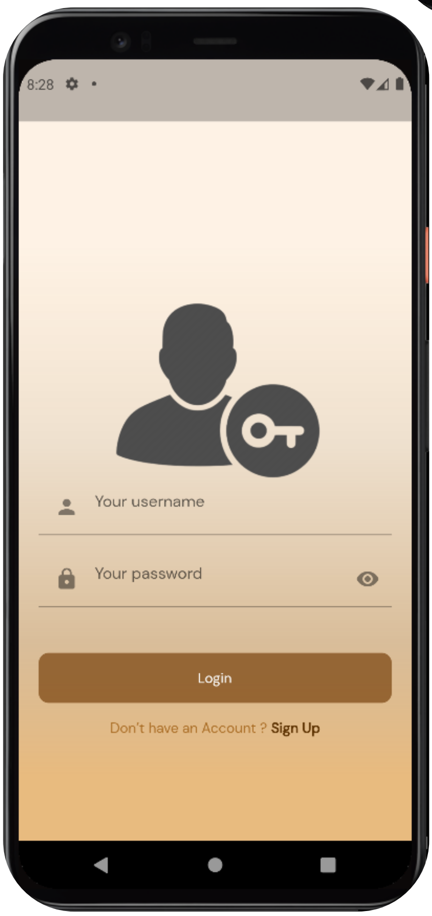


Figure 44: login screen

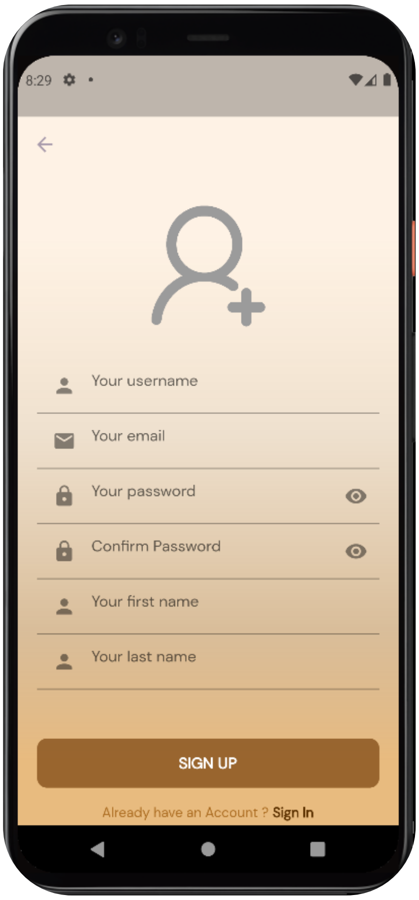


Figure 45: signup screen

8.4.2 Admin side

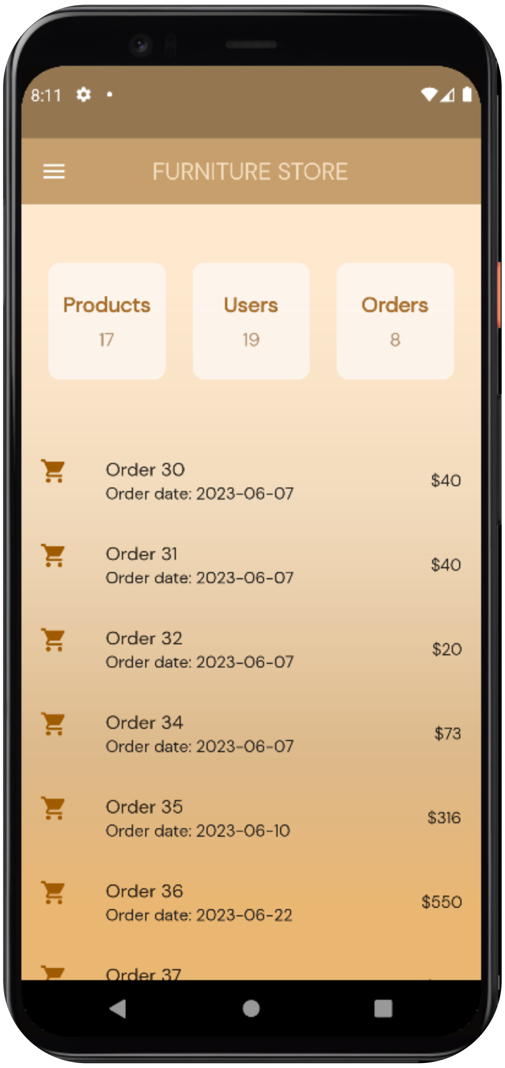


Figure 46: admin dashboard

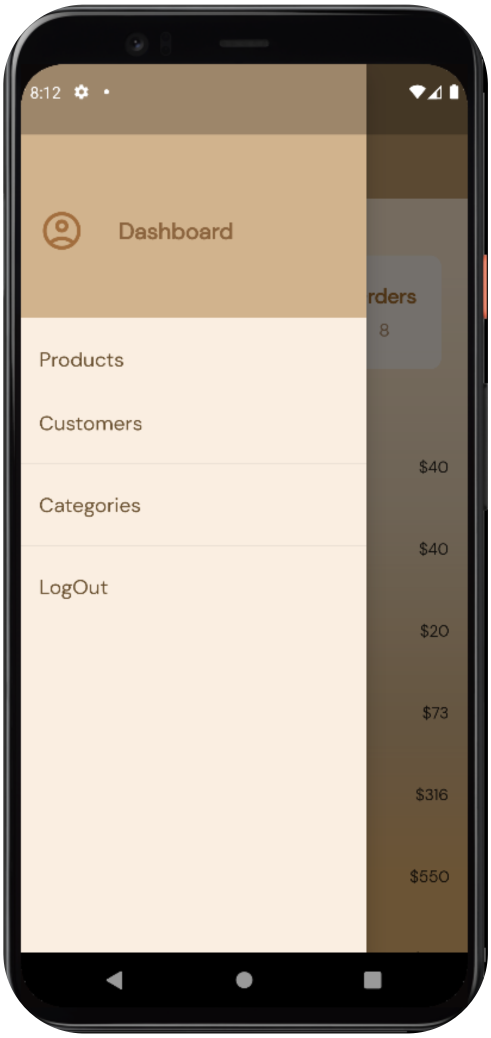


Figure 47: admin dahsboard-2

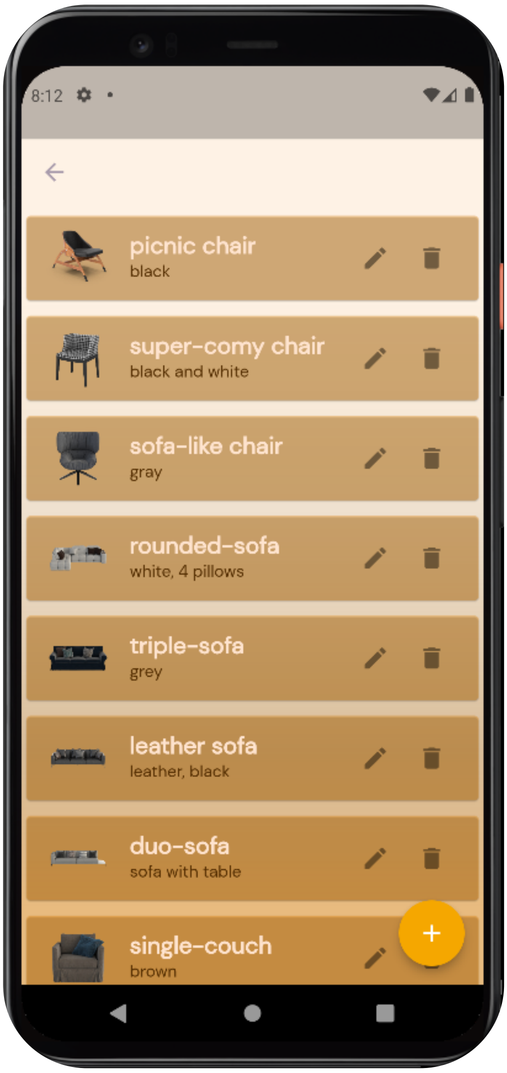


Figure 48: view all products

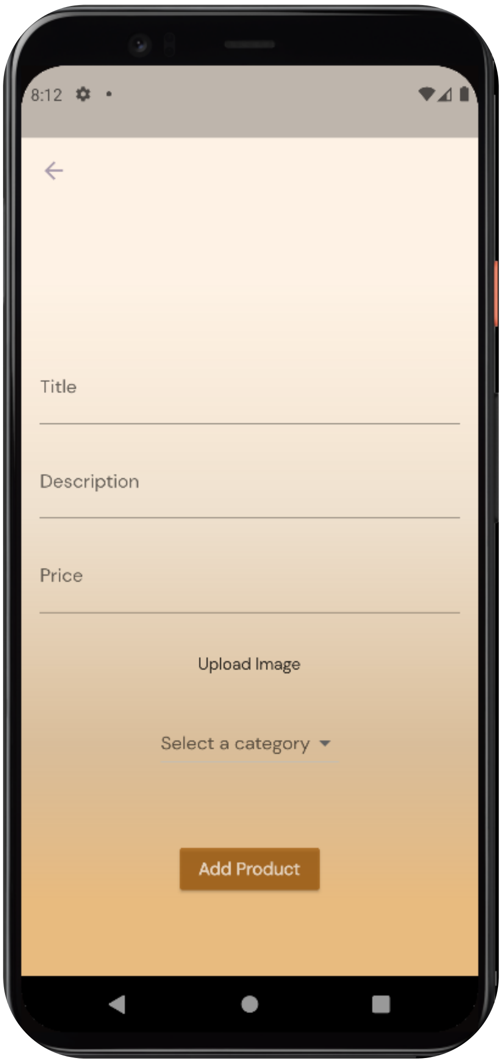


Figure 49: add product

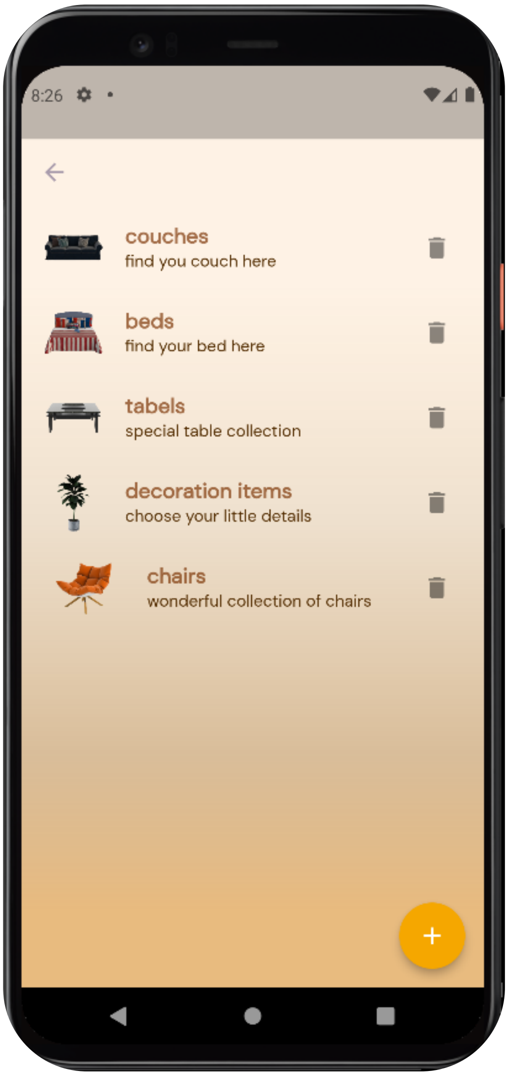


Figure 50: view all categories

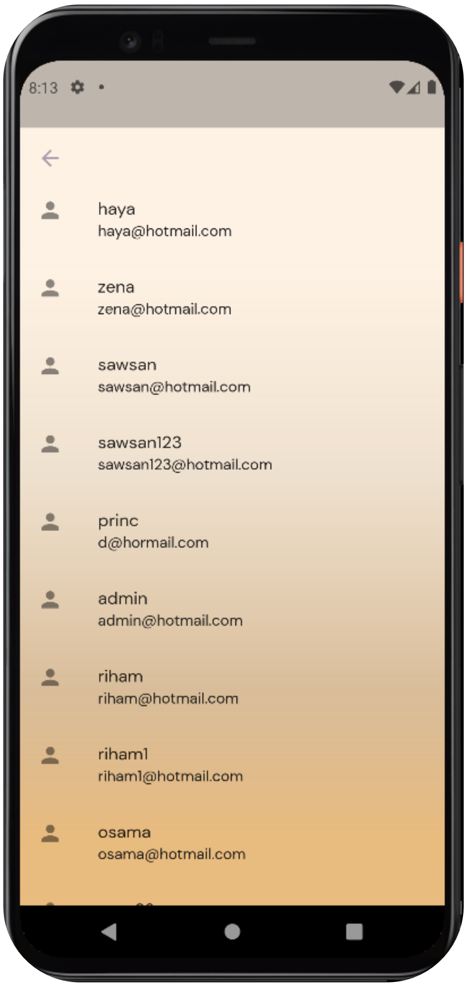


Figure 51: view all customers

5.5.3 User side

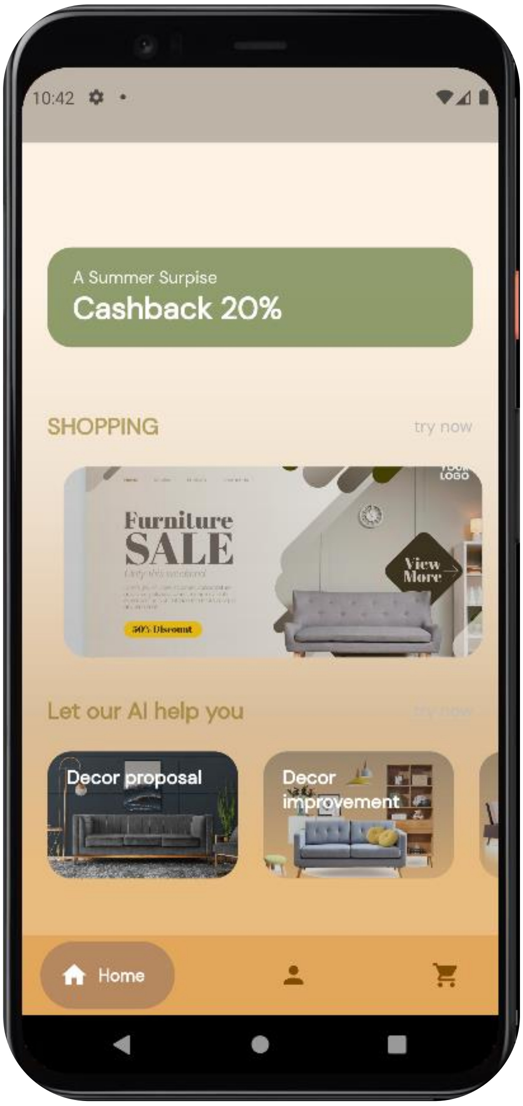


Figure 52: home screen

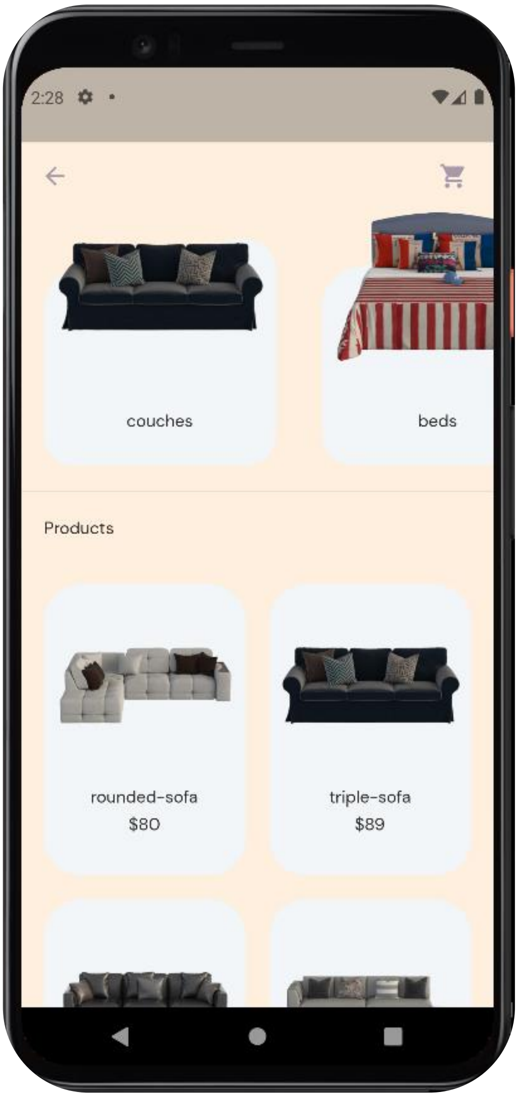


Figure 53: shop screen

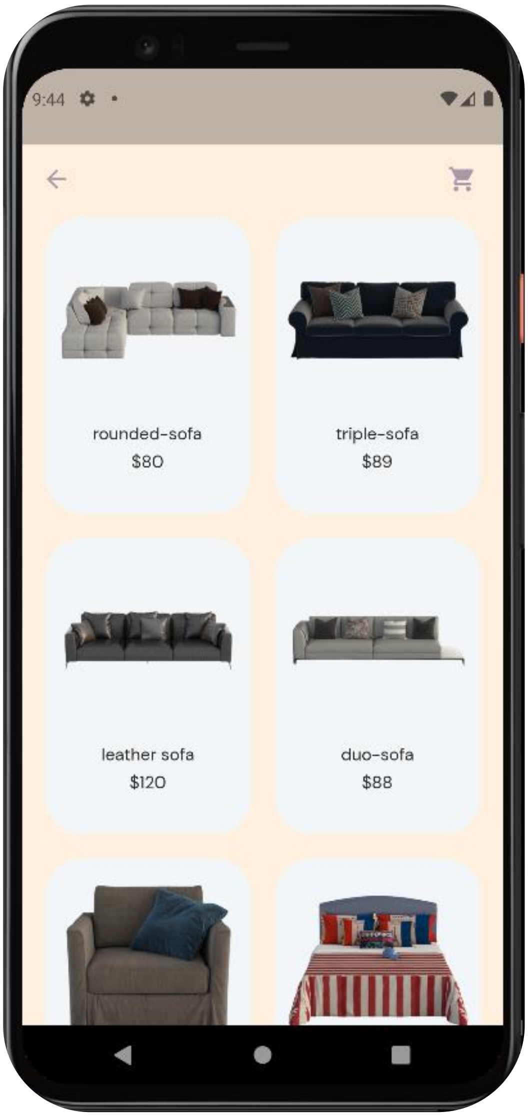


Figure 54: products screen

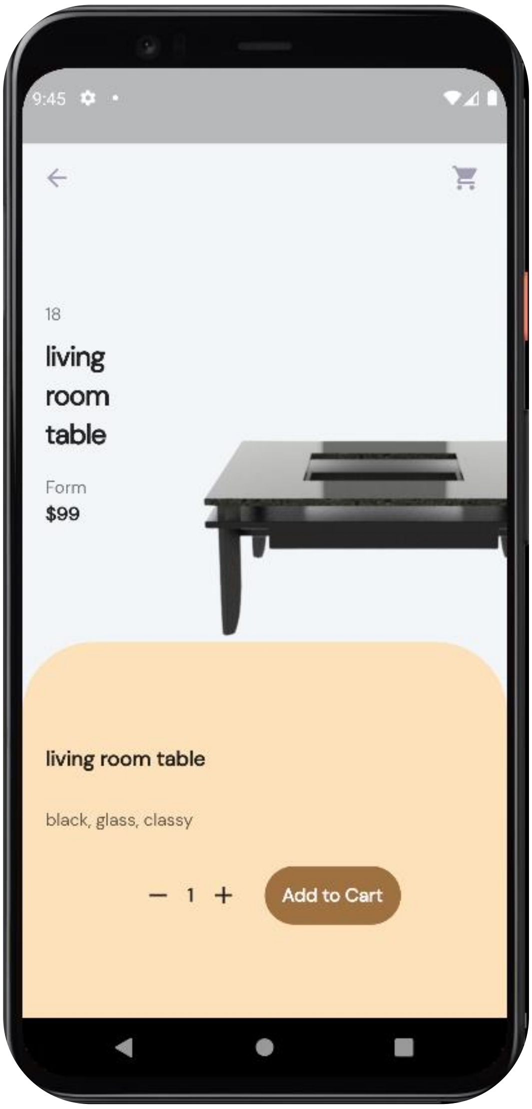


Figure 55: product detail

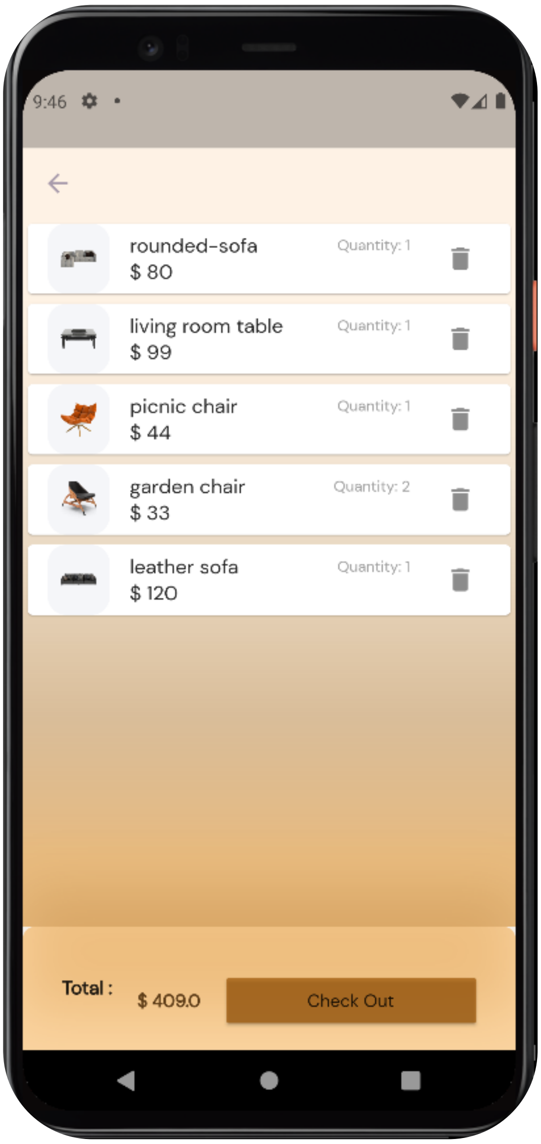


Figure 56: cart screen

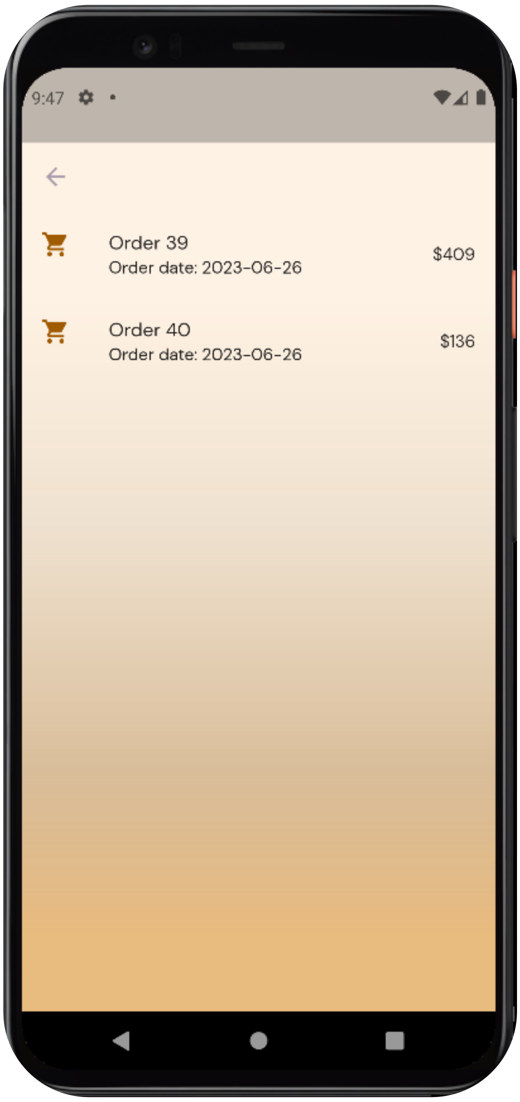


Figure 57: orders screen

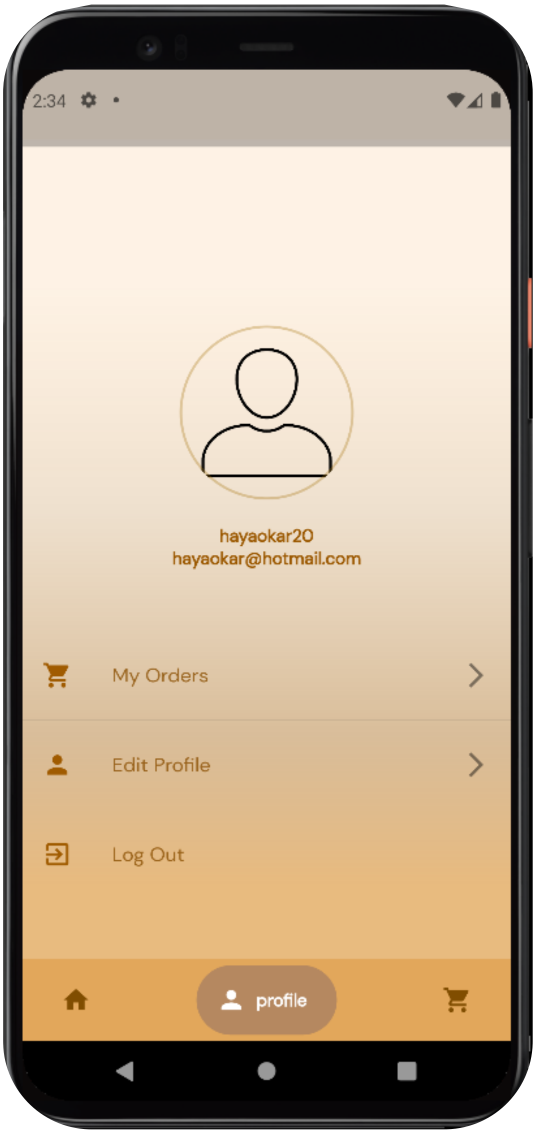


Figure 59: profile screen

Figure 58: expert system screen

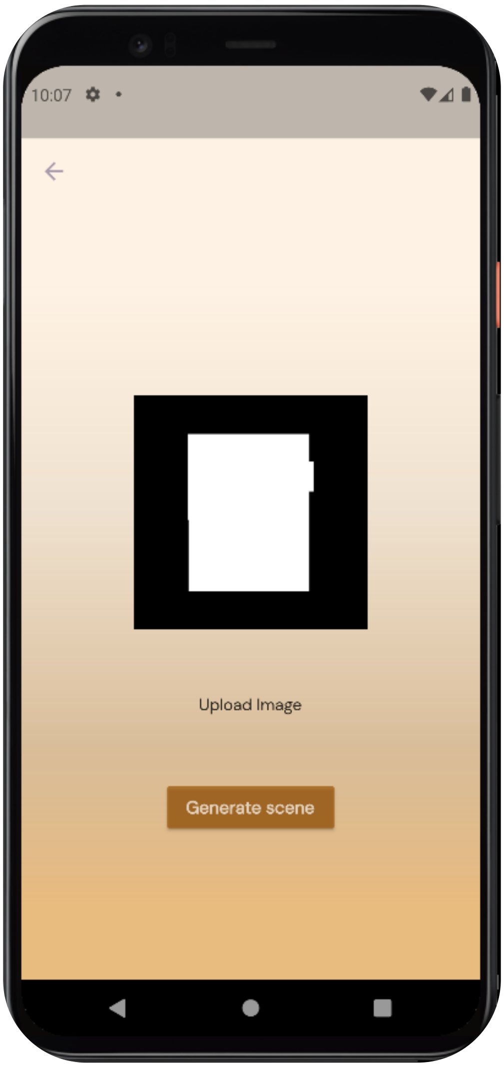


Figure 61: scene generation screen

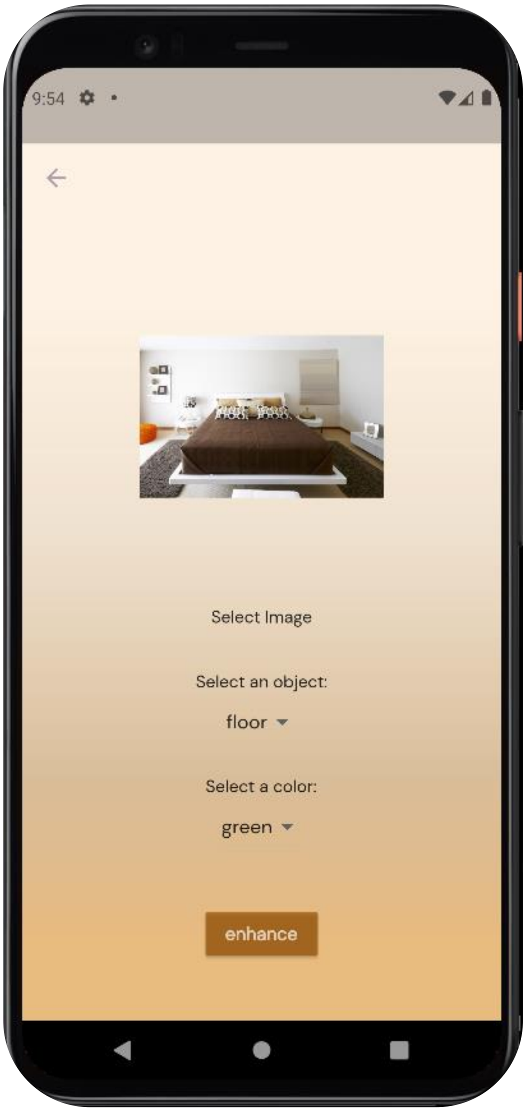


Figure 60: scene enhancement

Future Works

For future works, the DecorMate app holds promising potential for further advancements. One possible direction is training the scene generation model using the furniture catalogue provided by the integrated e-commerce app. This would enable the app to suggest specific products available for purchase, seamlessly integrating design inspiration with practical shopping options. Furthermore, enhancing the app to utilize the trained model in real time would provide users with immediate and dynamic design suggestions as they explore different layouts and decor choices. Additionally, implementing a secure online payment method would further improve the app's functionality, ensuring a seamless and trustworthy shopping experience for users. These professional enhancements have the potential to elevate DecorMate to new heights, providing users with an all-encompassing and efficient platform for home decoration.

To add more, this project has the potential to be expanded to support a wider range of room types, thereby enhancing its capabilities. By extending the current framework, we can accommodate various room configurations and address diverse user needs.

Conclusion

In conclusion, The DecorMate app is an AI-powered solution that simplifies and enhances the home decorating process. By utilizing Autoregressive transformers, it generates intelligent layouts, optimizing furniture and decor placement. Through the integration of GAN technology, DecorMate enhances visual appeal with stunning decor options. The app's expert system provides tailored recommendations by analyzing user preferences, ensuring the best decor choices. With a mini e-commerce platform, users can conveniently browse and purchase curated items within the app. DecorMate empowers users to transform their living spaces with confidence and creativity.

References

[1] 2. a. R. P. D. Maria Kozhevnikov1, "Understanding immersivity: image generation and transformation processes in 3D immersive environments," 2012.

[2] X. W. C. Y. M. Nießner, "SceneFormer: Indoor Scene Generation with Transformers," Technical University of Munich, 2021.

[3] Z. H. R. R. M. H. Z. Suiyun Zhang, "Semantic 3D Indoor Scene Enhancement Using Guide Words," The visual computer, May 2017.

[4] B. J. C. P. &. Y. C. Ph.D., "Online Shopping," 2019.

[5] Maxime, "What is a Transformer?," medium, 4 jan 2019. [Online]. Available: https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04.

[6] A. V. N. P. F. F. Angelos Katharopoulos, "Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention," International Conference on Machine Learning, 2020.

[7] "flutter website," flutter, [Online]. Available: https://flutter.dev/.

[8] "Django website," Django, [Online]. Available: https://www.djangoproject.com/.

[9] Postman, [Online]. Available: https://www.postman.com/.

[10] A. V. N. P. a. F. F. Angelos Katharopoulos, "Transformers are rnns: Fast autoregressive transformers with linear attention," In Proc. of the International Conf. on Machine learning, 2020..

[11] "IKEA place," IKEA, [Online]. Available: https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050.

[12] "houzz application," [Online]. Available: https://www.houzz.com/.

[13] "homestyler application," [Online]. Available: https://www.homestyler.com/room-design-app.

[14] "planner5d application," [Online]. Available: https://planner5d.com/.

[15] u. F. B. C. L. G. L. Z. C. L. Q. Zeng1, "3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics," 21 4 2021.

[16] F. Y. A. Z. A. X. C. M. S. T. F. Shuran Song, "Semantic Scene Completion from a Single Depth Image," 2017.

[17] S. S. J. M. R. C. Wenbin Li, arXiv, 2018.

[18] Q.-H. P. D. T. N. Binh-Son Hua, "Scenenn: A scene meshes dataset with annotations," IEEE, 2016.

[19] T.-W. Y. S. S. S. W. S. Zhengqin Li, " Openrooms: An end-to- end open framework for photorealistic indoor scene datasets," arXiv preprint arXiv, 2020.

[20] H. F. B. C. L. G. L. Z. J. Wang1, "3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics," 2021.

[21] M. S. A. X. C. D. R. KAI WANG, "Deep Convolutional Priors for Indoor Scene Synthesis," 2018.

[22] K. W. Y.-a. L. Daniel Ritchie, "Fast and Flexible Indoor Scene Synthesis via Deep Convolutional Generative Models," 2018.

[23] L. L. C. S. Q. ,. Z. J. Y. L. Bailin Yang, "Automatic interior layout with user-specified furniture," Elsevier Ltd., 2020.

[24] H. K. Peter K´an, "Automated Interior Design Using a Genetic Algorithm," 2018.

[25] A. L. Z. Zhang2, "End-to-End Optimization of Scene Layout," arXiv, 2020.

[26] A. S. Y. Z. S. S. T. F. J. X. Fisher Yu, "Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop," arXiv, 2015.

[27] A. X. C. M. S. M. H. T. F. M. N. Angela Dai, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," arXiv, 2017.

[28] N. S. L. L. J. D. W. K. S. M. P. Timo Hackel, "Semantic3D: A new Large-scale Point Cloud Classification Benchmark," 2017.

[29] Y. X. Y. S. Chen Zhang, "Decorating Your Own Bedroom: Locally Controlling Image Generation with Generative Adversarial Networks," arXiv, May 2021.

[30] A. A. N. N. F. T. a. M. N. Johanna Wald, "Environments, RIO: 3D Object Instance Re-Localization in Changing Indoor," arXiv:1908.06109v1 [cs.CV] , Aug 2019.

[31] K. A. N. C. S. H. K. Nasir Ahmad, "Computer Vision Based Interior Design," December 2018.

[32] N. W. KURT LEIMER, "LayoutEnhancer: Generating Good Indoor Layouts from Imperfect Data," 2022.

[33] N. S. N. P. J. U. L. J. A. N. G. L. Ashish Vaswani, "Attention is all you need. In Advances in Neural Information Processing Systems (NIPS)," p. 5998–6008, 2017.

ملخص

نقدم في هذا المشروع تطبيق DesignMate الذي يهدف إلى مساعدة المستخدمين في عملية اختيار التصاميم المنزلية بسرعة وبأفكار مميزة وإبداعية. يعتمد هذا التطبيق على الذكاء الاصطناعي ويقدم ميزات متنوعة لتلبية احتياجات المستخدمين.

واحدة من الميزات الرئيسية للتطبيق هي إقتراح المفروشات المناسبة لشكل غرفة محددة. يتمكن المستخدم من إدخال صورة شكل الغرفة، وبناءً عليه يقوم التطبيق بتقديم اقتراح تصميم مناسب للمفروشات.

بالإضافة إلى ذلك، يوفر التطبيق خدمة تحسين تصميم غرفة محددة. يمكن للمستخدم تحميل صورة للغرفة التي يرغب في تحسينها، ومن ثم يقوم التطبيق بتوفير الصورة المحسنة للتصميم.

كما يحتوي التطبيق على نظام خبير، حيث يقدم اقتراحات للتصاميم الأنسب استنادًا إلى الأسئلة التي يجيب عليها المستخدم.

وأخيرًا، يتضمن هذا المشروع متجرًا إلكترونيًا مصغرًا يعرض مجموعة من المفروشات المنزلية للمستخدمين للاطلاع والشراء.

باختصار، يهدف تطبيق DesignMate إلى تسهيل عملية اختيار وتحسين التصاميم المنزلية من خلال استخدام تقنيات الذكاء الاصطناعي، ويوفر تجربة مستخدم متكاملة وإبداعية لتحقيق تصاميم منزلية فريدة ومناسبة لاحتياجات الأفراد.

الجامعة العربية الدولية

الجامعة العربية الدولية AIU جامعة سورية خاصة أُحدثت عام 2005، خططها الدراسية والوثائق الصادرة عنها معتمدة ومصدقة من قبل وزارة التعليم العالي في الجمهورية العربية السورية.

تعمل الجامعة على تحقيق الأهداف الآتية:

- إعداد جيل متميز من الخريجين الجامعيين القادرين على تلبية الحاجات النوعية للمجتمع والنهوض به.
- الإسهام في البحوث العلمية النظرية والتطبيقية التي تخدم أغراض التنمية الوطنية، ويتم العمل على حث الأساتذة والعاملين الأكاديميين على البحث العلمي والمشاركة في المؤتمرات والندوات التي تنظم الأبحاث.
- تحقيق الشراكة مع الجامعات العربية والأجنبية المرموقة بهدف التطوير والتحديث المستمرين للعمل الأكاديمي والقيام ببحوث علمية مشتركة.
- استقطاب الكفاءات الأكاديمية والبحثية المتميزة عن طريق توفير البيئة المناسبة لعملها.

الجامعة العربية الدولية من الجامعات السورية الأولى التي جرى تأسيسها وافتتاحها، وقد تمكنت من اجتذاب الكفاءات التعليمية والبحثية والإدارية المتميزة، لإنشاء صرح متكامل من النواحي الأكاديمية والتنظيمية والإدارية. وتمكنت من تخريج كوادر من المبدعين والمتميزين من خلال توفير بيئة تعليمية ترتكز إلى مقومات نوعية ومادية فريدة منها:

- الخطط الدراسية الحديثة والمتطورة المستندة إلى نظام الساعات المعتمدة.
- الأطر التعليمية المنتقاة بعناية كبيرة.
- المختبرات العلمية الحديثة، ومختبر للمكتبات الإلكترونية.
- المحفزات المادية والمعنوية للطلبة.
- تطبيق طرائق التدريس التفاعلي.
- التوجيه والإرشاد الأكاديمي والتربوي.
- مجموعة كبيرة من اتفاقيات التعاون العلمي مع جامعات محلية وإقليمية ودولية ذات سمعة مرموقة.
- اتفاقيات ومذكرات تفاهم متعددة مع العديد من مؤسسات المجتمع المدني.
- الحرم الجامعي اللائق والمزود بكافة المرافق العلمية والرياضية والترفيهية، والذي نشجعك على زيارته والتعرف على مزاياه.
- الأنشطة والأندية الطلابية بمختلف أنواعها: الرياضية والثقافية والعلمية والاجتماعية.

في الجامعة العربية الدولية سنوات الحياة الجامعية هي وقت للاستثمار في مستقبل الطالب. فالمعارف والخبرات التي يحصلها في قاعة المحاضرات والمختبرات ستساعده في تطوير ذاته، وستمنحه أسباب النجاح في التخصص الذي اختاره، والنشاط الطلابي الذي يمارسه سيساعده في توسيع أفقه، وفعاليات التدريب والأندية والرياضة ستمكنه من تطوير مواهبه، ولربما تساعده في اكتشاف مواهب جديدة.
ليستثمر وقته وذهنه وروحه في جامعتنا كي يجني فوائد عمله والوقت الذي كرسه في السنين القادمة. ونحن سوف نكون بجانب طلبتنا في كل خطوة على دربهم.

الجامعة العربية الدولية

كلية الهندسة المعلوماتية والاتصالات

مشروع التخرج

مساعد تصميم منزلي ذكي

تم تقديمه إلى

قسم الهندسة المعلوماتية

تقديم

إنعام أنجو **رهام غازي**

هيا عوكر

بإشراف

الدكتور أبي صندوق **المهندسة خلود الجلاد**

تموز 2023