FLOORVERSE - A Universe of Floor Plan Possibilities

DSN4092 - CAPSTONE PROJECT

Phase – II Report
Submitted by

S.No	Registration No.	Name
1.	21BCE11618	Ansh Prakash
2.	21BCE11425	Snigdha Pandey
3.	21BCE11527	Naveen Kumar Singh
4.	21BCE10815	Esther George Sam
5.	21BCE10485	Munish Thakur

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING AND TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

VIT Bhopal University
Bhopal
Madhya Pradesh

APRIL 2025

Declaration of Originality

We hereby declare that this report entitled **'FLOORVERSE: A Universe of Floor Plan Possibilities '** represents our original work carried out as part of our project at VIT Bhopal University. To the best of our knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any other degree or diploma at VIT Bhopal University or any other institution. Works of other authors cited in this report have been duly acknowledged under the section **'References'**.

Date: 03-04-2025

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to everyone who contributed to the successful completion of "**FLOORVERSE - A Universe of Floor Plan Possibilities"**
First and foremost, we express our deepest appreciation to**<Supervisor>**, our project supervisor, our reviewers, whose invaluable guidance, support, and expertise played a crucial role throughout this journey. Their insights and encouragement kept us focused, ensuring that our work remained rigorous and impactful.

We also acknowledge the dedication and collaboration of our fellow team members, whose diverse skills and perspectives enriched our discussions, problem-solving, and overall team dynamics. Their commitment was instrumental in shaping this project into what it is today. Lastly, we extend our sincere gratitude to our families, friends, and loved ones for their unwavering support, patience, and understanding throughout this endeavor. Their encouragement provided us with the motivation and resilience needed to overcome challenges and achieve our goals.

This project is the result of collective effort, innovation, and collaboration, and we deeply appreciate the contributions of all those involved.

Thank you.

Bonafide Certificate

Certified that this project report titled **"FLOORVERSE - A universe of floor plan possibilities"** is the Bonafide work of (21BCE11618 Ansh Prakash, 21BCE11527 Naveen Kumar Singh, 21BCE11425 Snigdha Pandey, 21BCE10815 Esther George Sam, 21BCE10485 Munish Thakur) who carried out the project work under my supervision.

This project report (DSN4095-Capstone Project) is submitted for the Project Viva-Voce examination held on 3rd April 2025.

Supervisor

1. INTRODUCTION

Designing a home involves orchestrating complex spatial relationships where each room—from kitchens and bedrooms to living areas—must not only meet strict functional requirements but also embody aesthetic qualities that promote comfort and usability. Traditionally, this process requires close collaboration between architects and clients, with numerous rounds of feedback to address aspects such as lighting, ventilation, and overall flow. This iterative approach, however, is both time-consuming and expensive, often limiting access to personalized design services.

FLOORVERSE reimagines this traditional model by automating the floorplan generation process through advanced machine learning techniques. In our current implementation, which is detailed in the accompanying code files, we employ a Conditional Variational Autoencoder (CVAE) integrated within a Flask-based backend and paired with an interactive, user-friendly frontend. Unlike conventional design software that relies heavily on manual input and expert oversight, our system learns to capture intricate spatial relationships from a large dataset of floorplan images, enabling it to generate realistic and functionally efficient layouts based solely on user-defined criteria.

A key challenge in this automation is balancing essential functional requirements—such as room adjacency and ease of access—with the subtler elements of design, including visual harmony and spatial balance. Rather than relying on rigid, hard-coded rules, FLOORVERSE leverages deep learning to understand these complex relationships, thereby reducing the need for continuous manual intervention while still adhering to established architectural standards and individual preferences.

Furthermore, the platform addresses another common challenge faced by homeowners: visualizing how furniture will fit within a space. By offering interactive tools that allow users to adjust design parameters and immediately preview the resulting floorplan, FLOORVERSE empowers individuals to make informed decisions about interior layout without the need for professional consultation. In essence, FLOORVERSE is dedicated to democratizing home design. Our platform simplifies the intricate process of residential planning, saving both time and cost, and ultimately empowers users to take charge of creating spaces that are not only functionally sound but also aesthetically pleasing.

1.1 Motivation

The motivation behind FLOORVERSE stems from the pressing need to transform the conventional, labor-intensive process of home design into a more accessible, efficient, and cost-effective experience. Traditional architectural design requires extensive collaboration between professionals and clients, characterized by multiple iterations to balance strict functional requirements with aesthetic considerations. This process is not only time-consuming and expensive but also often restricts personalized design solutions to a privileged few.

FLOORVERSE seeks to address these challenges by leveraging advanced machine learning techniques to automate the generation of floor plans. By learning intricate spatial relationships from a vast dataset of floorplan images, the system can generate realistic, functionally efficient layouts based solely on user-defined criteria. This not only reduces the dependency on continuous manual intervention but also democratizes home design, making personalized architectural planning accessible to a broader audience.

The project is driven by a vision to simplify and revolutionize residential design—empowering users to create spaces that meet both their functional needs and aesthetic preferences, without the traditional barriers of time, cost, and professional expertise.

1.2 Objective

- Automate the generation of residential floorplans using quantum-powered AI.
- Offer user-friendly tools for non-professionals to explore and visualize various design styles.
- Reduce the cost and time involved in traditional architectural design.
- Provide adaptive algorithms for practical and aesthetic layouts.

2. EXISTING WORK / LITERATURE REVIEW

Table 2: Literature Review/Pre-existing work

Sr. No.	Title	Year of Publication	Description	Authors
1.	Graph2Plan: Learning Floorplan Generation from Layout Graphs	2020	1. Combined generative modeling using deep neural networks and user-in-the-loop design for a floorplan generation model. 2. The user inputs a building boundary and can also specify constraints on room numbers, locations, and adjacencies. (b) Possible graph layouts are retrieved from a dataset based on the user input. The retrieved graphs are adjusted to the input boundary. The graphs guide the network in generating corresponding floorplans, encoded as a set of room bounding boxes and a floorplan raster image. We post-process this output to obtain the final, vectorized floorplan.	Ruizhen Hu, Zeyu Huang,Yuhan Tang,Oliver Van Kaick,Haozhang, Hui Huang
2.	DANIEL: A Deep Architecture for Automatic Analysis and Retrieval of Building Floor Plans	2017	1. The results of the study show that the proposed deep learning framework, DANIEL, outperforms state-of-the-art methods for floor plan retrieval. The results are evaluated using precision-recall plots and mean average precision (MAP) values.	Divya Sharma ,Nitin Gupta ,Chiranjoy Chattopadhyay. Sameep Mehta

3.	Data-driven Interior Plan Generation for Residential Buildings	2019	1. A two-stage data-driven approach is used for generating residential floor plans based on given boundaries. The approach involves first locating rooms and then determining wall positions in the floor plan generation process. Initially, the living room is positioned based on its centrality and connectivity to other rooms, followed by an iterative process to add additional rooms. Finally, walls are identified using an encoder-decoder network and refined into vector representations. 2. The average score for floor plans generated by the proposed method was 18.72 for designers and 20.11 for general users, indicating a preference over competitors.	Wen Ming Wu ,Xiao-Ming Fu, Rui Tang,Yuhan Wang,Yu-Hao Qi, Ligang Liu
4.	Improved Automatic Analysis of Architectural Floor Plans	2011	1. The proposed method for floor plan analysis involved three main steps: ● Information Segmentation: Segmentation algorithms are applied to separate different types of information, starting with wall detection followed by text and graphics segmentation. ● Structural Analysis: The structure of the extracted information is analyzed to retrieve the layout of the rooms.	Sheraz Ahmed,Andreas Dengel,Marcus Liwicki,Markus Weber

			● Semantic Analysis: A semantic analysis is conducted to extract the functions of the rooms based on the detected elements. 2. Introduced new preprocessing methods, including line differentiation and convex hull component removal. 3. Achieved a room recognition accuracy of 79%, outperforming previous methods by 10%.	
5.	Floor Plan Generation and Auto Completion Based on Recurrent Neural Networks	2017	1. Trained LSTM(Long Short-Term Memory) to mimic behavior through sequences from simple floor plans. 2. Approaches used - Block Generation Sequencer and Vector Prediction Sequencer. 3. Error rates: Block (65.78%), Vector (66.08%) Approach shows potential but requires further performance improvements for practical use.	Johannes Bayer,Syed Saqib Bukhari,Andreas Dengel
6.	Automatic Extraction of Indoor Spatial Information from Floor Plan Image: A Patch-Based Deep Learning Methodology Application on Large-Scale	2021	1. Used a CNN-based framework for automatic floor plan analysis of large-scale complex buildings. 2. Utilized patch-based deep learning to handle varied scales and high-resolution inputs. 3. Achieved a detection rate of 87.77% and recognition accuracy of 85.53%,	Hyunjung Kim, SeongyongKim, KiyunYu

	Complex Buildings		comparable to existing studies.	
7.	Evaluation of Deep Learning-Based Automatic Floor Plan Analysis Technology: An AHP-Based Assessment	2021	1. It discusses the evaluation of deep learning-based automatic floor plan analysis technology using an AHP-based assessment. 2. The proposed technology aims to automatically extract indoor spatial information from raster images of floor plans through a deep learning algorithm, enabling critical indoor elements extraction. 3. The proposed evaluation framework underscores the effectiveness of automatic floor plan analysis technology in acquiring indoor spatial information, showcasing its potential in comparison to traditional manual methods.	Hyunjung Kim
8.	Automation in Interior Space Planning: Utilizing Conditional Generative Adversarial Network Models to Create Furniture Layouts	2023	1. The study finds that the BicycleGAN model outperformed the two other CGAN models (pix2pix and SPADE) in generating furniture layouts, and that the post-processing method improved the generated results by 30-50% in both CGAN and architectural metrics. 2. The post-processing method suggested was shown to improve generated results by 30–50% in	Hanan Tanasra, Tamar Rott Shaham ,Tomer Michaeli ,Guy Austern , and Shany Barath

			both CGAN and architectural metrics.	
9.	Integrating Multimodal Generative AI and Blockchain for Enhancing Generative Design in the Early Phase of Architectural Design Process	2024	1. AI is used to generate designs based on textual and visual inputs, offering architects the ability to explore diverse design options. The AI enhances creativity by iterating between sketches and AI-generated outputs, using tools like Midjourney. 2. There are challenges related to data security, intellectual property, and legal concerns in human-AI collaboration. Blockchain addresses these by offering immutable records for design metadata, preventing tampering.	Adam Fitriawijaya and Taysheng Jeng
10.	Floor Layout Planning Using Artificial Intelligence Technique	2017	1. The proposed system uses a Genetic Algorithm to generate multiple layout designs for furniture in a master bedroom, considering various constraints and relationships between furniture elements. 2. The algorithm used presents a hybrid behavior by combining Evolutionary Strategy with an Stochastic Hill Climbing technique, uses adaptive operators to perform the geometric transformations of the rooms, their walls and connections, and openings according to previously stored	Mitali Chavan, Nidhi Menon, Ritika Kumar, Shivani Rana

			information. 3. The system generates optimal results with an initial population size of 20,000 and 5 generations.	
11.	Space Layouts & GANs.	2020	1. The article presents three use-cases for designing housing floor plans using GANs: free plan generation, program-specific generation, and structure-specific generation. The results show that GANs can generate relevant floor plan designs that take into account various constraints and initial conditions. 2. We propose showcasing possibilities offered by Generative Adversarial Neural Networks models (GANs), and their ability to generate relevant floor plan designs.	S. Chaillou
12.	An Efficient Computable Metric for Comparing Polygonal Shapes.	1989	1. The method developed in this research satisfies several desirable properties, including being a metric, invariant under translation, rotation, and change-of-scale, and reasonably easy to compute. The method works for both convex and nonconvex polygons and runs in time O(mnlog mn) where m is the number of vertices in one polygon and n is the number of vertices in the other.	Esther M Arkin,L Paul Chew, D P Huttenlocher , K Kedem

			2. The key findings of the study are: • The distance function, d2(A, B), achieves its minimum at one of mn discrete points on [0,1], which are called critical events. • The distance function can be computed exactly in time O(mn(m + n)) and approximately in time O(mn log mn).	
13.	GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models	2018	1. GraphRNN is a deep autoregressive model for generating graphs, addressing challenges in modeling complex graph distributions with non-local dependencies. 2. It learns by decomposing graph generation into sequential node and edge formations, outperforming baselines and scaling to larger graphs efficiently.	Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, Jure Leskovec
14.	Generative Floor Plan Design Using Deep Learning: An AI-Powered Approach	2024	1. The paper presents a combination of a sequential and graph model to generate floor plans, using recurrent neural networks and generative adversarial networks (GANs) to process user inputs and produce realistic floor plans.	Balraj Vaidya, Padmakar Pimpale, Guruprasad Khartadkar
15.	Automatic Generation of AI-powered Architectural Floor Plans using	2023	1. The results show that the method using grid compression shows relatively stable results even though it uses only about 0.2% of the training data size and	Hun Lim

	Grid Data		about 30% of the training time compared to the method without grid compression. 2. The results showed that similar levels of automatic drawing generation were possible with only 0.2% of the original data capacity after preprocessing. However, the generated drawings showed limitations in the ambiguity of the boundaries of the room structure, and not all generated labels were used in the testing process of the CGAN model.	
16.	HouseDiffusion: Vector Floorplan Generation via a Diffusion Model With Discrete and Continuous Denoising		1. The study uses a diffusion model based architecture, which employs a Transformer architecture at the core, to control the attention masks based on the input graph-constraint. The approach represents a floorplan as a set of 1D polygonal loops, each corresponding to a room or a door, and generates 2D coordinates of room/door corners.	Mohammad Amin Shabani, Sepidehsadat Hosseini, Yasutaka Furukawa
17.	Generating floorplans for various building functionalities via latent diffusion model		1. The model outperformed existing generative models (e.g., Stable Diffusion, DALL-E 2) in terms of image fidelity and architectural quality, as measured by metrics likeFrechet Inception Distance (FID)	Mohamed R. Ibrahim, Josef Musil, and Irene Gallou

			and Structural Similarity Index (SSIM).	
18.	Space Layouts & GANs	2020	1. Includes the ability of GANs to generate relevant floor plan designs. The study also identifies three main use-cases for designing housing floor plans, including free plan generation, program-specific generation, and structure-specific generation. 2. We turn to GAN models, and Pix2Pix, to help us design housing floor plans, given a set of initial conditions & constraints	Stanislas Chaillou; Architect Data Scientist Spacemaker
19.	Generating Interior Design from Text: A New Diffusion Model-Based Method for Efficient Creative Design		1. This introduces a novel AI-based method for generating interior designs from text descriptions using a diffusion model. The authors address inefficiencies and lack of creativity in traditional interior design by creating a unique dataset (IDSSF-64), proposing a new loss function, and fine-tuning a diffusion model. 2. The proposed model outperformed mainstream diffusion models (Midjourney, DALL-E2, Stable Diffusion) in generating high-quality, realistic interior designs with specified decoration styles and spatial functions.	Junming Chen , Zichun Shao, and BinHu

20.	Performance Prediction of AI-generated Architectural Layout Design: Using Daylight Performance of Residential Floorplans as an Example		1. The study uses a diffusion model to generate building floor plans, fine-tuned using the Low-Rank Adaptation (LoRA) model. The pix2pixHD model is used to predict daylight performance. The study employs a parametric algorithm to generate a dataset of floor plans.	Xiao Hu, Hao Zheng, Dayi Lai
21.	Graph-Based Generative Representation Learning of Semantically and Behaviorally Augmented Floorplans	2020	1. the proposed model is effective for floorplan representation and generation, and the use of an attributed graph and an LSTM VAE can capture the design semantics and human behavioral features of floorplans. 2. With floorplans annotated with edge and/or behavioral attributes, with about 77.6% of neighbors accurately ordered in their sequences based on their perceived similarity	Vahid Azizi; Muhammad Usman; Honglu Zhou et al
22.	Room Classification on Floor Plan Graphs using Graph Neural Networks	2021	1. The key findings of the study are that GraphSAGE and Topology Adaptive GCN outperform other models, including multilayer perceptron, graph convolutional networks, and graph attention network, with an accuracy of 80% and 81% respectively. 2. The best performance is achieved by TAGCN with 81.07% test accuracy followed	Abhishek Paudel; Roshan Dhakal; Sakshat Bhattarai

			by GraphSAGE with test accuracy of 80.26%.	
23.	Self-Organizing Floor Plans	2021	1. Includes the potential of self-organizing floor plans in generating automatic building floor plans and spatial configurations, and the importance of collaboration between designers and computer scientists in developing new methods for automated spatial solutions.	Silvio Carta
24.	Generative AI models for different steps in architectural design: A literature review	2024	1. The results show that generative AI has the potential to significantly enhance the architectural design process, with various models being utilized to generate diverse content, including text, images, music, videos, and 3D models	Chengyuan Li; Tianyu Zhang; Xusheng Du et al.
25.	Tell2Design: A Dataset for Language-Guided Floor Plan Generation	2023	1. The evaluation of the Sequence-to-Sequence model using macro and microIntersection over Union (IoU) scores between the ground-truth and generated floor plans	Sicong Leng; Yang Zhou; Mohammed Haroon Dupty et al.

3. SYSTEM ARCHITECTURE & REQUIREMENTS

- System Design:

1. *System Design: Conditional Variational Autoencoder (CVAE) Workflow*

   This pipeline outlines the systematic approach used to train, evaluate, and deploy the CVAE model for automated floorplan generation.

   - Data Preparation

i. Load Data: The raw floorplan images and corresponding metadata are loaded from the RPlan dataset (comprising 80,788 images and associated CSV files).

ii. Preprocess Data: Images are resized and normalized using defined transformations and categorical attributes are encoded, ensuring consistency for model input.

- Model Setup

i. Define CVAE Architecture: attention mechanisms for node representation learning.

  - **Encoder:** Utilizes convolutional neural network (CNN) layers to process grayscale floorplan images and a multi-layer perceptron (MLP) to encode condition features.

  - **Decoder:** Employs transposed convolutions to reconstruct floorplan images from the latent space, guided by the user-defined conditions.

ii. Initialize Model: The CVAE model's parameters are initialized and configured for training, with appropriate device allocation (CPU or GPU) for efficient computation.

- Training

i. Train Model: The model is trained using a loss function that combines binary cross-entropy for reconstruction accuracy and KL divergence for regularizing the latent space. The training process involves iterating over the preprocessed data to adjust model weights via backpropagation.

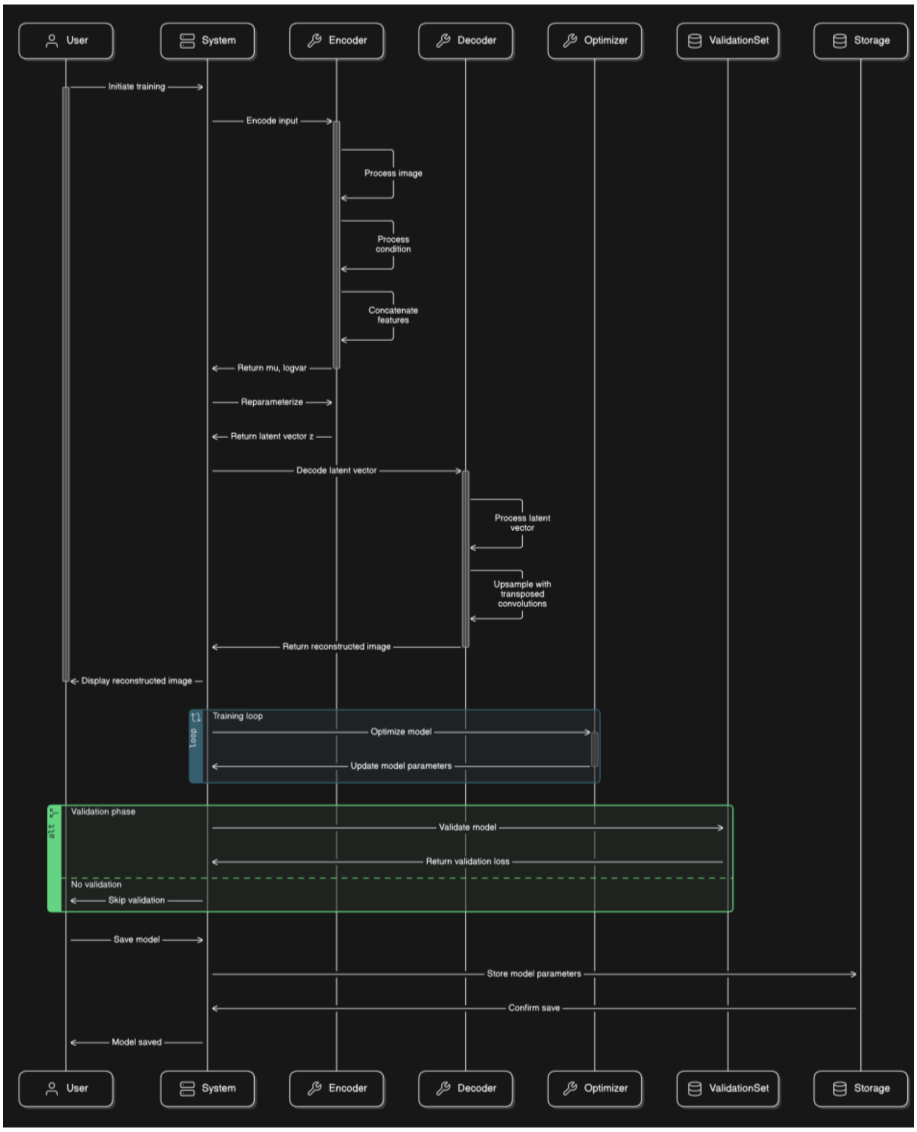


Figure 3 : Conditional VAE Training Process

ii. Evaluate Model: Performance is assessed on a separate validation set, comparing the original and reconstructed floorplans to ensure the model generalizes well to unseen data.

iii. Save Model: The best-performing model checkpoint is saved for future deployment and further refinements.

- Visualization

i. Generate Floorplans: The Flask-based backend receives user-defined parameters and uses the trained CVAE model to generate floorplans in real time.

ii. Visualize Results: The generated floorplan is combined with efficiency metrics (e.g., space utilization, room ratio) and rendered as an image for user review via an interactive web interface.

The design of the system is driven by several core objectives that ensure robust, real-time floorplan generation. These objectives are implemented as follows:

- *Data Acquisition and Preprocessing*:
  The system begins by loading raw floorplan images and corresponding metadata from the RPlan dataset. As demonstrated in our dataset preparation scripts, the data undergoes rigorous preprocessing—including image resizing, normalization, and augmentation—to ensure consistency. Numerical and categorical features are also normalized, ensuring that all inputs are appropriately scaled for the model.
- *Model Architecture and Initialization*:
  At the heart of the system is a Conditional Variational Autoencoder (CVAE) as detailed in the model code. The architecture is composed of an encoder that processes grayscale images via convolutional layers and integrates condition features through a multi-layer perceptron, and a decoder that reconstructs floorplans using transposed convolutions. The model is carefully initialized to set up the training process, ensuring compatibility with both CPU and GPU environments.
- *Training and Optimization*:
  The training process, as outlined in our training scripts, involves iterating over the preprocessed dataset to minimize a composite loss function that combines binary cross-entropy for image reconstruction and KL divergence for latent space regularization.
  Parameter optimization is achieved through backpropagation, and model performance is continuously evaluated on a validation set to guarantee accuracy and generalization.
- *Visualization and Evaluation*:
  To assess the model's performance, the system generates visual outputs comparing original, reconstructed, and newly generated floorplans. Efficiency metrics—such as space utilization and room ratio—are calculated and displayed alongside the generated images. This visualization is seamlessly integrated with the Flask backend and interactive frontend.

○ <u>HARDWARE REQUIREMENTS:</u>

➔ Processor: Intel Pentium i3

➔ Hard Disk: 500 GB

➔ Monitor: 15" LED

➔ Input Devices: Keyboard, Mouse

➔ RAM: 6 GB

- <u>SOFTWARE REQUIREMENTS:</u>
  - ➔ Operating System: Windows 10 Pro
  - ➔ Programming Language: Python 3.10.9
  - ➔ Web Framework: Flask
  - ➔ Machine Learning Library: PyTorch & Torchvision

4. ROADMAP

Phase 1: Conceptualization and Requirements Gathering

- Literature Review & Market Analysis:
  Conduct an in-depth review of existing floorplan generation methodologies and identify gaps in current architectural design tools.
- Objective Definition:
  Establish clear project objectives, including the automation of floorplan generation and the creation of a user-friendly platform accessible to non-professionals.
- Technical Requirements:
  Define system requirements encompassing data sources, hardware and software specifications, and key performance metrics.

Phase 2: System Design and Prototyping

- System Architecture Development:
  Design a comprehensive architecture that integrates a Conditional Variational Autoencoder (CVAE) for floorplan generation with a Flask-based backend and interactive front-end.
- User Interface Design:
  Develop the initial landing page and interactive controls using HTML, CSS, and JavaScript, ensuring a clean, intuitive, and responsive layout.
- Backend Prototyping:
  Implement a prototype backend to manage API requests and serve generated floorplan images, establishing the foundation for real-time user interaction.

Phase 3: Model Development and Training

- CVAE Model Implementation:
  Build the CVAE architecture, including the encoder, reparameterization, and decoder components, as detailed in the project's codebase.
- Data Preprocessing Pipeline:
  Develop and refine the `RPlanDataset` class to handle image resizing, normalization, and feature scaling from the RPlan dataset (80,788 images).
- Training and Optimization:
  Train the model using the defined loss function (combining binary cross-entropy and KL divergence) and optimize parameters with an efficient training pipeline. Monitor performance on validation datasets to ensure robust generalization.
- Evaluation:
  Use dedicated evaluation scripts to compare original, reconstructed, and generated floorplans. Visualize results and compute efficiency metrics to assess the model's quality.

5. PROJECT TIMELINE

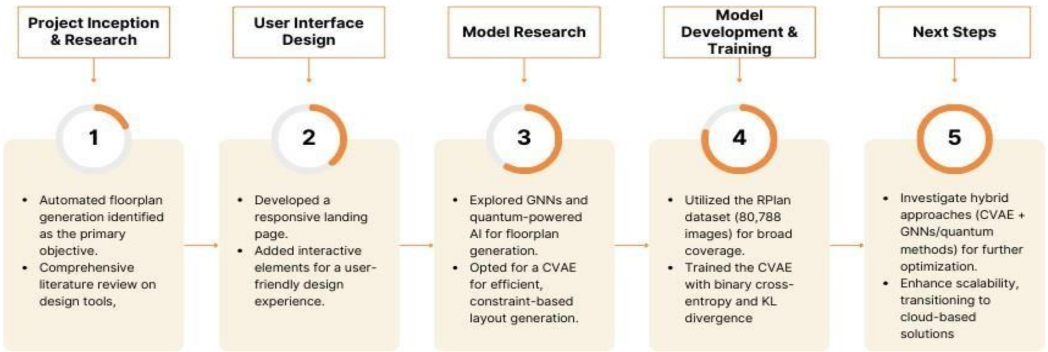


Figure 5 : Project Progress in Phases (post-phase-2)

6. WORKING PRINCIPLE:

- Data Preprocessing:
  - Convert floorplan images to grayscale, resize, and normalize.
  - Normalize design parameters (e.g., room counts, total area) for consistency.

- Encoding:
  - Extract spatial features from images using convolutional layers.
  - Integrate user-specified conditions via a multi-layer perceptron.
  - Map combined features into a latent space by computing mean and log-variance.
- Reparameterization:
  - Sample a latent vector from the latent space using the reparameterization trick.
- Decoding:
  - Use the latent vector and condition inputs to reconstruct or generate floorplan images with transposed convolutions.
  - Normalize the output image to ensure pixel values range between 0 and 1.
- Training:
  - Optimize the model using a loss function that combines binary cross-entropy and KL divergence.
  - Ensure accurate reconstruction and proper regularization of the latent space.
- Generation:
  - Generate new, customized floorplans in real time based on user-defined parameters.

7. WORK DONE

Front-End Development

- ***Landing Page:*** The FLOORVERSE platform launches with a landing page (refer to *index.html*), designed with modern HTML, CSS, and JavaScript. This page offers a clean, responsive, and intuitive interface that allows users—regardless of technical expertise—to configure key design parameters such as unit selection, total area, and room configurations. The design focuses on ease of use, enabling users to effortlessly explore the floorplan generation features.

Machine Learning Component:

Conditional Variational Autoencoder (CVAE):

The core of the system is a Conditional Variational Autoencoder (CVAE), as implemented in *model.py*. This model processes grayscale floorplan images along with normalized condition features (e.g., number of rooms, total area, etc.) to generate customized floorplans.

- **Encoder:** Utilizes convolutional layers to extract image features and a multi-layer perceptron to process condition data. The combined representation is then transformed into a latent space, capturing the essential characteristics of the floorplan.
- **Decoder:** Employs transposed convolutions to reconstruct floorplan images from the latent space, guided by the user-defined conditions.

  The CVAE is trained using a loss function that blends binary cross-entropy (to ensure accurate reconstruction) and KL divergence (to regularize the latent space), as detailed in *train_model.py* and evaluated in *evaluate.py*. This approach enables the system to generate realistic and functionally coherent floorplans based solely on user input.

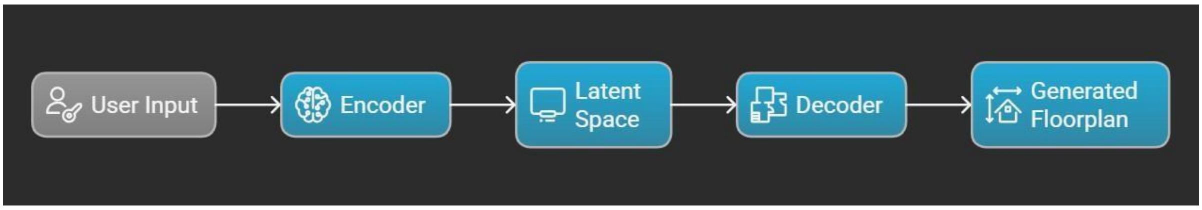


Figure 7 : CVAE Working

Key Features of the CVAE Model

1. **Robust Loss Function:**
   - Combines binary cross-entropy for image reconstruction accuracy with a KL divergence term to regularize the latent space. This dual-objective loss ensures the generated floorplans are both realistic and coherent.

2. **Comprehensive Data Normalization and Conditioning**:
   - The RPlanDataset class pre-processes floorplan images (resizing, normalization, and transformation) and normalizes numerical features using MinMaxScaler.
   - A dedicated normalize_condition method prepares user-defined condition inputs, ensuring consistent and effective conditioning of the model.

3. **Flexible Generation Capability**:
    - Includes a generate method that allows the model to produce new floorplans by sampling from the latent space conditioned on user input. This enables the generation of diverse designs tailored to specific constraints.

4. **Modular and Scalable Design:**
    - The model's modular implementation facilitates seamless integration into a Flask-based backend (as seen in *app.py*), enabling real-time floorplan generation and dynamic interaction through the front-end interface.
    - This structure also supports future enhancements and scalability, ensuring the system can evolve with additional features or more advanced conditioning techniques.

Conditional Variational Autoencoder (CVAE) Architecture:

- Encoder:
    - Processes grayscale floorplan images through a series of convolutional layers with batch normalization and ReLU activations, extracting robust spatial features.
    - Simultaneously, it processes design condition features (e.g., number of rooms, total area) via a multi-layer perceptron.
    - The image features and condition representations are concatenated and passed through fully connected layers to compute the latent space's mean and log-variance.
- Reparameterization:
    - Implements the reparameterization trick to sample latent vectors from the computed distribution, ensuring differentiability during training.
- Decoder:
    - Takes the latent vectors and condition data to reconstruct floorplan images using fully connected layers followed by transposed convolutions, effectively upsampling the latent representation back to image dimensions.

DATASET

The **Rplan dataset** is a publicly available dataset containing **80,788 residential floor plan images**. It is widely used for **architectural design automation** and **generative AI applications**. The dataset provides **vector-based representations** of floor plans, allowing machine learning models to analyze spatial structures, room layouts, and connectivity between different sections of a house.

8. METHODOLOGY

The FLOORVERSE project employs a structured, multi-phase methodology to develop a robust, automated system for generating residential floorplans using a Conditional Variational Autoencoder (CVAE). The process encompasses data collection, preprocessing, model design, training, evaluation, and deployment, ensuring both functional and aesthetic requirements are met.

1. *Data Collection and Preprocessing:*
   I. Dataset Acquisition: The primary data source is the RPlan dataset, containing 80,788 floorplan images alongside detailed metadata (e.g., room counts, total area). This dataset provides a diverse and comprehensive foundation for training the generative model.
   II. Data Loading: The `RPlanDataset` class is used to load and parse the CSV metadata, retrieving image paths and corresponding architectural parameters. If an image directory is specified, the class locates and loads each floorplan image accordingly.
   III. Image Preprocessing: Each floorplan image is converted to grayscale and uniformly resized (to 256×256 pixels) using a series of transformations. The images are then normalized and transformed into tensors, which are the required format for input into the deep learning model.
   IV. Feature Normalization: Numerical features are scaled using MinMaxScaler to ensure consistency across samples. Binary features (indicating the presence or absence of specific room types) are also processed accordingly. A dedicated `normalize_condition` method standardizes user input conditions, ensuring they are compatible with the model's expected input format.

2. *Model Architecture:*
   a. **Conditional Variational Autoencoder (CVAE)**
      I. Encoder: The encoder extracts spatial features from the input floorplan images using four convolutional layers, each followed by batch normalization and ReLU activations. In parallel, a multi-layer perceptron processes the condition vector (which includes attributes such as the number of rooms and total area). These two streams are concatenated and passed through fully connected layers to compute the mean and log-variance vectors that define the latent space.
      II. Reparameterization: The model applies the reparameterization trick to sample a latent vector from the computed distribution, ensuring the process remains differentiable for effective backpropagation.

III. Decoder: The decoder reconstructs the floorplan image from the latent vector by combining it with the condition vector. This combined input is passed through fully connected layers and subsequently through transposed convolution layers, which upsample the representation back to the image dimensions. A final sigmoid activation normalizes the output pixel values between 0 and 1.

3. *Training and Optimization:*

I. Training Setup: The training process, implemented in *train_model.py*, involves splitting the dataset into training and validation sets using PyTorch's random_split function. Data loaders are then created to facilitate efficient batch processing.

II. Loss Function: The CVAE model is trained using a composite loss function:

III. Reconstruction Loss: Binary cross-entropy ensures the reconstructed image closely matches the original input.

IV. KL Divergence: Regularizes the latent space by encouraging the learned distribution to approximate a standard normal distribution.

V. Optimization: The Adam optimizer is employed to adjust model parameters through backpropagation over multiple epochs, with continuous monitoring on a validation set to ensure robust generalization.

4. *Evaluation:*

I. Quantitative and Qualitative Assessment:

An evaluation script (evaluate.py) compares original floorplan images, their reconstructions, and newly generated samples. Visualizations are generated using matplotlib to present side-by-side comparisons, while efficiency metrics (such as space utilization and room ratio) are computed to quantitatively assess the model's performance.

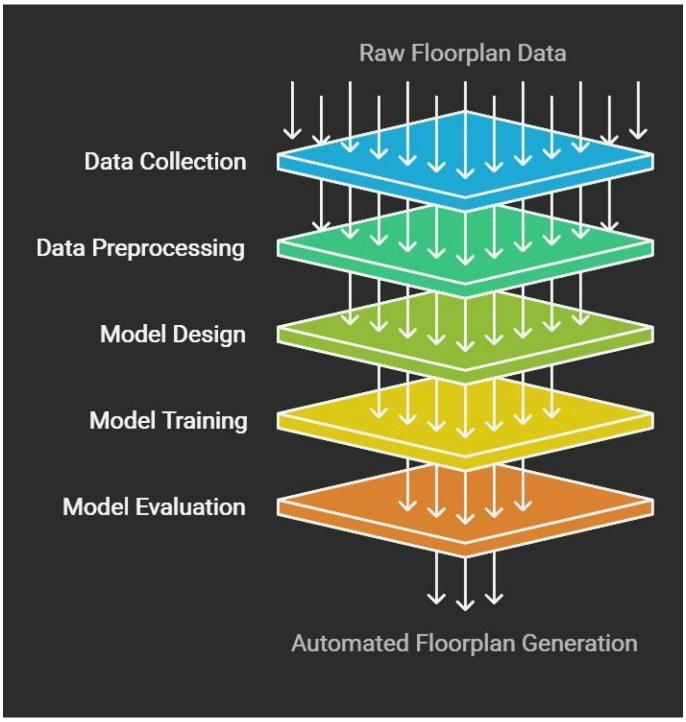
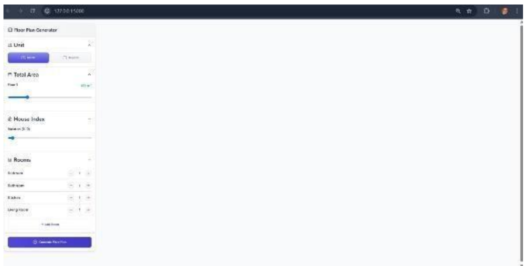
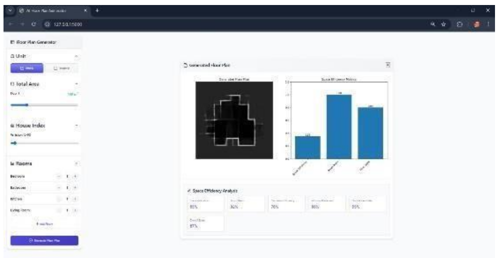
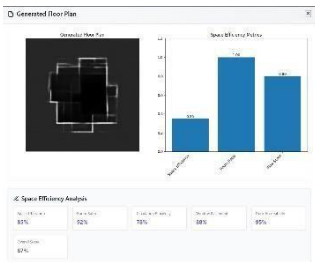


Figure 8: FLOORVERSE Methodology

9. RESULTS & DISCUSSION

Table 9 : Observation & Output

Test Case	*Result/Observation*	*Screenshot*
Landing Page:	Displays the main panel where users can set key parameters.	
Generated Floor Plan & Metrics:	Shows the system's output after users click "Generate Floor Plan." A grayscale floor plan is produced alongside a bar chart illustrating key space efficiency metrics.	
Detailed Efficiency Analysis:	Offers a closer look at the generated floor plan and its associated efficiency metrics, including space utilization, room ratio, and flow score	

CVAE Model Performance

The performance of the Conditional Variational Autoencoder (CVAE) model has been evaluated through both quantitative metrics and qualitative visual assessments. The results indicate that the model is capable of effectively capturing the underlying structure of floorplan images and generating realistic, user-specific designs.

- Reconstruction Quality:

  The CVAE consistently reconstructs floorplan images with high fidelity. During training, the model minimizes a composite loss function—combining binary cross-entropy for image reconstruction and KL divergence for latent space regularization—which has led to stable convergence on both the training and validation datasets. Side-by-side comparisons of the original, reconstructed, and newly generated floorplans demonstrate that the model retains essential spatial details and design features.

- Latent Space Regularization:

  The effective application of the reparameterization trick has resulted in a well-structured latent space. This regularization is critical for generating diverse and plausible floorplan variations when sampling from the latent distribution.

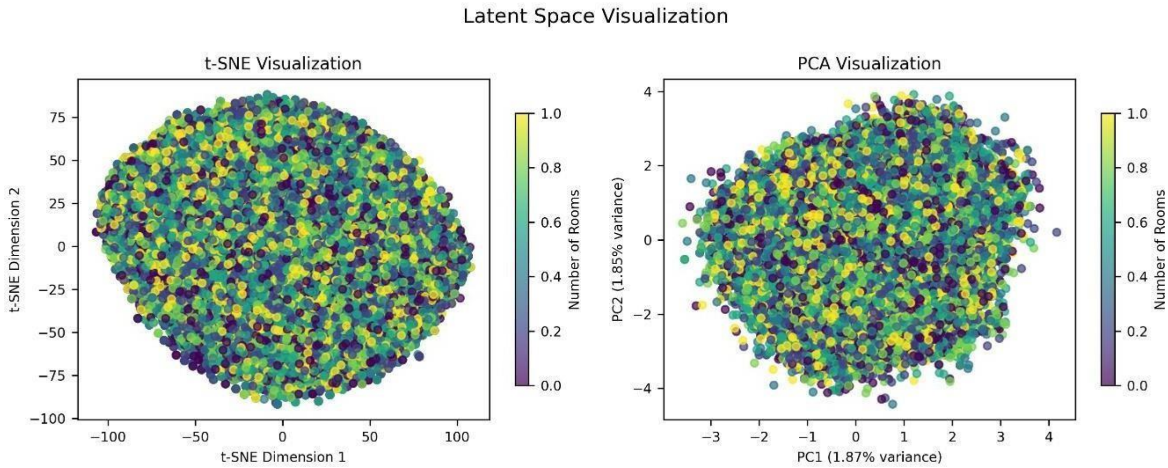


Fig 9.1: Latent Space & PCA Visual Plot

- Efficiency Metrics:

  In addition to visual assessments, the model's outputs are evaluated using efficiency metrics such as space utilization, room ratio, and flow score. These metrics provide quantitative insights into how well the generated floorplans meet practical design requirements. The results have shown that the generated designs are not only visually coherent but also exhibit favorable efficiency characteristics, closely aligning with user-defined constraints.

- Generalization Capability:

The model has been rigorously tested on unseen data, and the evaluation scripts indicate robust generalization across a wide range of floorplan configurations. This suggests that the model is not overfitting to the training data and can adapt to varying design inputs effectively.

- Visual and Quantitative Assessment:

  Evaluation scripts generate composite plots comparing original images, reconstructions, and new generations. These visual comparisons, along with computed efficiency metrics, highlight the model's ability to produce high-quality floorplans that balance both functional and aesthetic considerations

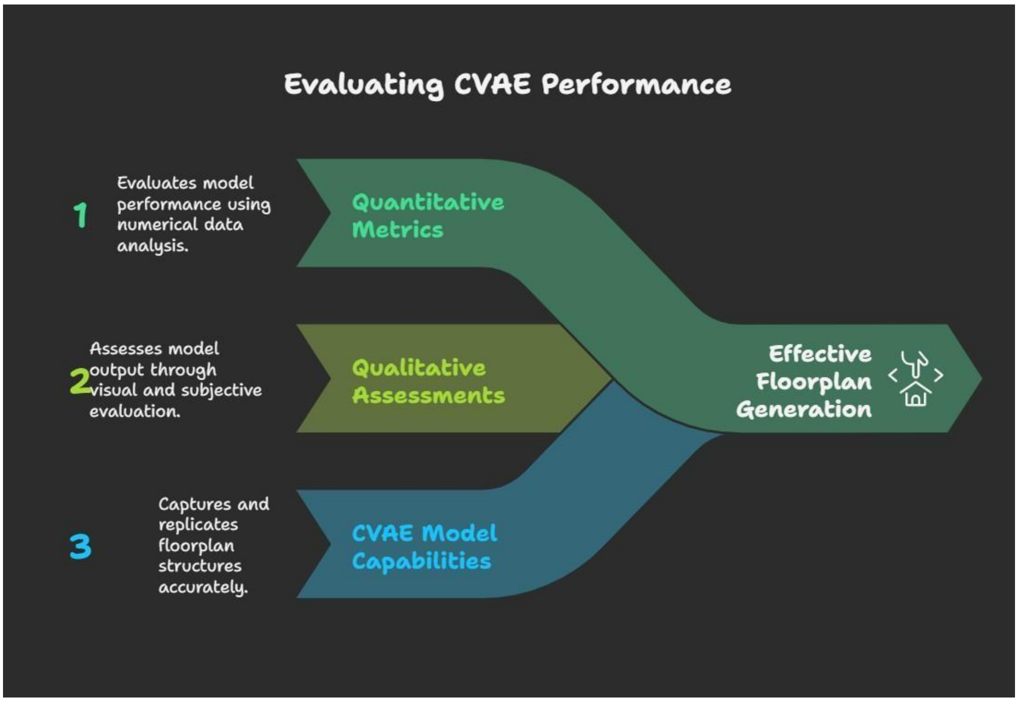


Fig 9.2: Performance measuring parameters

- **Training Results:**

  At the end of 50 epochs, the model's training loss stabilized at **8651.1139** and the validation loss at **8677.2230**, suggesting that the Conditional Variational Autoencoder (CVAE) converged without significant overfitting. The final checkpoint was successfully saved to the model_output directory as vae_model.pt.

- **Key Observations**
  - ***Close Train & Validation Losses:***

    The minimal gap between training and validation losses indicates consistent learning across the dataset, hinting at good generalization to unseen data.
  - ***Stable Convergence:***

    The smooth progression of losses through all epochs confirms that the model has reached a stable point in its optimization, ready for further evaluation or real-world application.

- ***Foundation for Future Improvements:***

  While these losses alone do not directly translate to an accuracy metric, the convergence provides a solid basis for subsequent assessments of floorplan quality and potential refinements to the model architecture or training pipeline.

Hybrid Integration Potential

The current FLOORVERSE system leverages a Conditional Variational Autoencoder (CVAE) to generate floorplan designs efficiently based on user-defined parameters. However, the full potential of automated architectural design can be realized by integrating complementary models into a hybrid framework. Such an approach would combine the generative strengths of the CVAE with additional methodologies to refine and optimize the floorplan outputs further.

Key Aspects of Hybrid Integration:

1. ***Enhanced Structural Modeling:***

   A hybrid system could integrate graph-based models—such as Graph Neural Networks (GNNs)—to capture the inherent spatial relationships between different architectural elements more explicitly. While the CVAE effectively generates initial floorplan layouts, a GNN can further refine these designs by modeling rooms as nodes and their connections as edges, thereby enforcing realistic adjacency and connectivity constraints.

2. ***Multi-Objective Optimization:***

   Integrating quantum-powered components, such as Variational Quantum Classifiers (VQCs), can enable advanced multi-objective optimization. This integration would allow the system to simultaneously optimize for functional requirements (e.g., efficient space utilization, logical room placements) and aesthetic considerations (e.g., natural lighting, overall flow), thus enhancing the quality of the generated designs.

3. ***Sequential Refinement Pipeline:***

   A possible approach to hybrid integration is to adopt a sequential refinement pipeline. In this scenario, the CVAE first produces a base layout that captures the general structure of the floorplan. Subsequently, additional modules—such as a GNN for spatial consistency and a quantum optimizer for multi-objective tuning—could iteratively refine the layout. This step-by-step process would yield designs that are not only visually coherent but also meet intricate functional and aesthetic criteria.

4. ***User-Centric Customization:***

   Hybrid integration offers the opportunity to embed more sophisticated user constraints directly into the design process. By combining the flexibility of generative models with the precision of optimization techniques, the system can better accommodate detailed user preferences and produce personalized designs that are both practical and visually appealing.

10. INDIVIDUAL CONTRIBUTION

Table 10 : Individual Contribution

Name	Reg No.	Contribution
Ansh Prakash	21BCE11618	• Served as the **Project Lead,** overseeing the overall workflow and ensuring timely completion of tasks. • Proposed and led the development of the Conditional Variational Autoencoder (CVAE) model, steering the project toward an efficient and scalable approach for floorplan generation. • Supervised system development and coordinated research efforts, contributing significantly to the project's documentation and paper.
Snigdha Pandey	21BCE11425	• Helped with the development of the Conditional Variational Autoencoder (CVAE) model, integrating it with the user interface for real-time floorplan generation. • Managed data preprocessing, model optimization, and thorough documentation of project milestones. • Previously spearheaded the Variational Quantum Classifier (VQC) model, achieving notable accuracy and providing insights for future quantum integration.
Naveen Kumar Singh	21BCE11527	• Led front-end development, focusing on creating an intuitive, user-friendly interface with interactive features. • Ensured smooth functionality, assisted with model testing, and assisted with

		model testing and the documentation process.
Esther George Sam	21BCE10815	• Contributed extensively to project documentation and the development of the **Graph Neural Network (GNN)** model in earlier phases. • Conducted research on **diffusion models** and **Generative AI** methodologies, informing advanced design strategies for automated floorplan generation.
Munish Kumar	21BCE10485	• Supported documentation efforts and collaborated on creating presentation materials • Verified dataset integrity and participated in model validation, ensuring consistent performance.

11. CONCLUSION

The FLOORVERSE project represents a significant advancement in the field of automated architectural design. By harnessing the power of a Conditional Variational Autoencoder (CVAE) integrated with a Flask-based backend and a responsive, user-friendly front-end, the system successfully generates realistic and customizable floorplans based on user-defined criteria. This approach not only simplifies the traditionally complex and resource-intensive process of floorplan design but also democratizes access to personalized home planning tools.

Through meticulous data preprocessing, robust model training, and comprehensive evaluation, the project has demonstrated the feasibility of using deep generative models to capture intricate spatial relationships inherent in architectural layouts. The modular design of the system ensures scalability and sets the foundation for future enhancements, including the potential integration of hybrid models and advanced optimization techniques.

Overall, FLOORVERSE has achieved its core objectives of reducing design time, lowering costs, and empowering non-professional users to explore diverse design possibilities. The insights gained and the robust framework established in this project pave the way for ongoing improvements and the eventual transition from a research prototype to a fully operational design tool.

12. FUTURE WORK

As FLOORVERSE evolves, several enhancements and new features can be implemented to further elevate its capabilities and user experience:

a. Hybrid Model Integration:
   Future iterations could integrate additional generative models, such as Graph Neural Networks (GNNs) or Variational Quantum Classifiers (VQCs), with the current CVAE framework. This hybrid approach would combine the strengths of different architectures to optimize both the structural coherence and aesthetic quality of generated floorplans.

b. Enhanced User Interaction:
   The current front-end provides an intuitive landing page for parameter selection. Future improvements may include:

   i. Real-time editing tools that allow users to adjust floorplan elements dynamically.

ii. An advanced dashboard that visualizes efficiency metrics and design comparisons side-by-side.

iii. Interactive features for detailed customization, such as furniture layout optimization, which would help users visualize interior design arrangements.

c. Scalability and Performance Optimization:

With the RPlan dataset comprising over 80,000 images, further optimizations in data loading, preprocessing, and model inference will be essential to maintain responsiveness in real-time applications. This may involve:

i. Leveraging distributed computing or cloud-based solutions.

ii. Optimizing data pipelines to reduce latency.

iii. Incorporating hardware accelerations such as GPU clusters for training and inference.

d. Robust Evaluation Metrics:

The current evaluation mechanism provides basic efficiency metrics and visual comparisons. Future work could develop more sophisticated quantitative metrics for assessing floorplan quality, including:

i. Detailed spatial efficiency measures.

ii. User satisfaction indices based on empirical feedback.

iii. Automated assessments of functional and aesthetic design aspects.

e. Dataset Expansion and Augmentation:

Enriching the dataset with more diverse architectural styles, irregular room shapes, and multi-story designs could improve the model's generalization and versatility. Further, employing advanced data augmentation techniques could simulate a broader range of real-world scenarios, enhancing the robustness of generated outputs.

f. Integration of Emerging Technologies:

As quantum computing and advanced AI techniques mature, FLOORVERSE can explore the integration of:

i. Quantum-powered optimization techniques to refine layout designs under multiple constraints.

ii. Generative Adversarial Networks (GANs) or other state-of-the-art generative models to enhance creativity and design diversity

By scaling the hybrid model, incorporating generative AI, and improving the training dataset, FLOORVERSE aims to deliver a comprehensive, reliable, and creative residential design solution. These developments will ensure that the platform remains scalable, user-friendly, and capable of meeting the diverse needs of its users.

13. REFERENCES

1. Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver Van Kaick, Haozhang, Hui Huang (2020). *Graph2Plan: Learning Floorplan Generation from Layout Graphs*. In *ACM Transactions on Graphics*.
2. Divya Sharma, Nitin Gupta, Chiranjoy Chattopadhyay, Sameep Mehta (2017). *DANIEL: A Deep Architecture for Automatic Analysis and Retrieval of Building Floor Plans*.
3. Wen Ming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, Ligang Liu (2019). *Data-driven Interior Plan Generation for Residential Buildings*.
4. Sheraz Ahmed, Andreas Dengel, Marcus Liwicki, Markus Weber (2011). *Improved Automatic Analysis of Architectural Floor Plans*.
5. Johannes Bayer, Syed Saqib Bukhari, Andreas Dengel (2017). *Floor Plan Generation and Auto Completion Based on Recurrent Neural Networks*.
6. Hyunjung Kim, Seongyong Kim, Kiyun Yu (2021). *Automatic Extraction of Indoor Spatial Information from Floor Plan Image: A Patch-Based Deep Learning Methodology*.
7. Hyunjung Kim (2021). *Evaluation of Deep Learning-Based Automatic Floor Plan Analysis Technology: An AHP-Based Assessment*.
8. Hanan Tanasra, Tamar Rott Shaham, Tomer Michaeli, Guy Austern, Shany Barath (2023). *Automation in Interior Space Planning: Utilizing Conditional Generative Adversarial Network Models to Create Furniture Layouts*.
9. Adam Fitriawijaya, Taysheng Jeng (2024). *Integrating Multimodal Generative AI and Blockchain for Enhancing Generative Design in the Early Phase of Architectural Design Process*.
10. Mitali Chavan, Nidhi Menon, Ritika Kumar, Shivani Rana(2017) *Floor Layout Planning Using Artificial Intelligence Technique.*
11. Chaillou, S. Space Layouts & GANs. Artificial Intelligence and Architecture. 2020.
12. Exploring the Power of Generative Adversarial Networks https://www.presidio.com/exploring-the-power-of-generative-adversarial-networks-gans-with-azure/
13. Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, Jure Leskovec ,2018. *GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models.*
14. Balraj Vaidya, Padmakar Pimpale, Guruprasad Khartadkar ; *Generative Floor Plan Design Using Deep Learning: An AI-Powered Approach.*
15. Hun Lim , *Automatic Generation of AI-powered Architectural Floor Plans using Grid Data*
16. Mohammad Amin Shabani, Sepidehsadat Hosseini, Yasutaka Furukawa, *HouseDiffusion: Vector Floorplan Generation via a Diffusion Model With Discrete and Continuous Denoising*

17. Mohamed R. Ibrahim, Josef Musil, and Irene Gallou, *Generating floorplans for various building functionalities via latent diffusion model*
18. Chaillou, S., & Spacemaker, A. D. S. (2020). *Space Layouts & GANs*
19. Junming Chen , Zichun Shao , and BinHu , *Generating Interior Design from Text: A New Diffusion Model-Based Method for Efficient Creative Design*
20. Xiao Hu, Hao Zheng, Dayi Lai , *Performance Prediction of AI-generated Architectural Layout Design: Using Daylight Performance of Residential Floorplans as an Example*
21. Azizi, V., Usman, M., Zhou, H., Faloutsos, P., & Kapadia, M.. (2020). *Graph-Based Generative Representation Learning of Semantically and Behaviorally Augmented Floorplans.*
22. Paudel, A., Dhakal, R., & Bhattarai, S.. (2021). *Room Classification on Floor Plan Graphs using Graph Neural Networks.*
23. Carta, S. (2021). *Self-Organizing Floor Plans.* Harvard Data Science Review.
24. Li, C., Zhang, T., Du, X., Zhang, Y., & Xie, H.. (2024). *Generative AI models for different steps in architectural design: A literature review*. Frontiers of Architectural Research.
25. Leng, S., Zhou, Y., Dupty, M. H., Lee, W. S., Joyce, S. C., & Lu, J. W. Z. (2023). *Tell2Design: A Dataset for Language-Guided Floor Plan Generation.*
26. RPLAN Dataset. Residential floorplans dataset for training machine learning models.