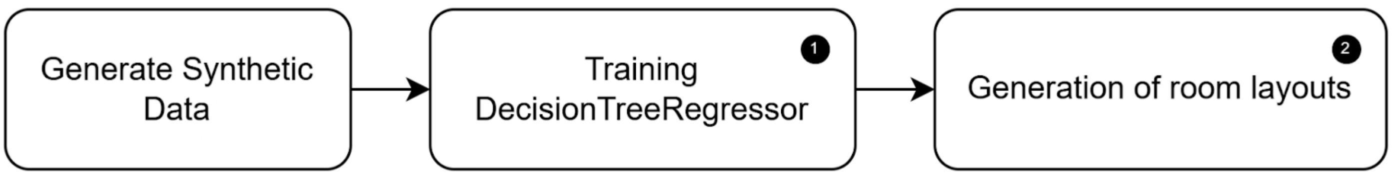
Assignment Report

Harshini Borugadda

AI-Based Room Layout Generation

Approach

To create realistic room layouts in a space-efficient way. Used a method called randomized recursive partitioning along with a decision tree approach. First, it divides the plot size into rooms randomly while keeping a balanced aspect ratio. To give more layouts in this we're randomly choosing to split either horizontally or vertically. Then, train a Decision Tree Regressor to learn how to partition rooms. This trained model can predict the size and location of rooms based on the input, such as the plot size and number of rooms desired. Finally, a visualization function displays the layout, maximizing the use of available space.



1. The DecisionTreeRegressor learns to generate a range of values for the x and y coordinate of the room (partition) as well as the width and height

2. The layout is generated in JSON and Image Format

1. Generate synthetic dataset

To create a synthetic dataset, we start by randomly selecting the plot's width, height, and number of rooms. Next, we divide the plot into smaller sections while ensuring that the aspect ratio remains balanced. We randomly choose whether to split the plot horizontally or vertically to add variety to the layouts. Once we have our dataset of room partitions, we use it to train a decision tree model dataset of room partitions, we use it to train a decision tree model.

2. Building the DecisionTreeRegression

We use a DecisionTreeRegressor, a machine learning algorithm that helps us identify patterns by breaking down numerical data. It's great for problems like room partitioning. In case input features we have plot width, plot depth, and the number of rooms, while the output consists of room partitions (x, y, width, height).
By using synthetic data for training, we enable the model to learn how different plot sizes and room counts affect partitioning, allowing us to predict realistic layouts for new inputs.

3. Training and Evaluation

In order to teach the model how to divide a given plot into rooms with the proper proportions, the DecisionTreeRegressor is fitted to synthetic data. Mean Absolute Error (MAE), which calculates the average discrepancy between the expected and actual room dimensions, is used to assess the model after training.

Analysis of model

Model Performance analysis done is using the Mean Absolute Error because this is best suited for the Regression purposes. The average MAE Score we got is 6.36 which is quite good for a model which is trained on synthetic data.

Challenges

1. Lack of available dataset

The absence of a real-world dataset for room partitioning is a significant obstacle to this solution, making it challenging to train the model on a variety of realistic layouts. A small or artificial dataset might not include all potential variations because floor plans differ greatly depending on architectural standards, cultural preferences, and space utilization requirements. This may result in a model that does well on the generated data but has trouble generalizing to the real world. Furthermore, it is more difficult to verify whether the generated layouts are actually optimal or merely statistically reasonable in the absence of actual examples. The accuracy and usefulness of the model could be greatly increased by adding actual architectural plans or professionally created layouts to the dataset.

2. Time constraints

To generate realistic room layouts, a model such as DCGAN (Deep Convolutional Generative Adversarial Network) would need a lot of data, a lot of processing power, and a long training period. It is also more difficult to fine-tune GANs due to issues like mode collapse and unstable training. Rather, we employ a Decision Tree Regressor, which efficiently creates structured room layouts while being significantly faster, more interpretable, and requiring less data. Without the complexity of deep learning, this method solves the room partitioning problem more practically by striking a balance between speed and accuracy.

Potential Improvements

1. Use of DCGAN

By identifying patterns in actual floor plans, a DCGAN can learn complex spatial relationships and produce more varied and realistic room layouts, which could be a major improvement. DCGANs can produce imaginative and organic designs that more closely resemble human-designed layouts than decision trees, which adhere to rigid rule-based splits. Furthermore, a DCGAN could generalize better to a variety of architectural styles with sufficient training data, making it a more adaptable and potent method for this task.

2. Adding of more features to model complex layouts

The layout generation can be expanded beyond basic rectangles to any arbitrary shape by identifying additional features, such as irregular plot boundaries, entry points, structural constraints, and zoning regulations. The model can learn how to divide complex spaces while adhering to the specified constraints by integrating these features. The model would be more flexible to real-world architectural designs if it included features like wall orientations, polygonal representations, and accessibility requirements. This would allow the model to generate layouts that fit complex plots.