To Graph or not to Graph

Evaluating the Predictive Power of Graph Representations for Floorplan Prediction

By: Alisha Khan

Project Plan

1. Embeddings.py: processes floorplan images to extract features and embeddings for use in ML models
2. ML2- Attempt to predict more undefined rooms. Perform feature selection, find best model, hyperparameters tuning etc.
3. Floorplan Modification 2
4. Graph completion model
5.

1

PROJECT MOTIVATION

OBJECTIVE

The objective of this project is to enhance the accuracy of predicting the internal configurations of buildings, specifically the types of rooms, by utilizing solely external information, such as the placement of walls, doors, and windows.

WHY IS THIS USEFUL

This helps improve the safety of firefighters, special forces, and SWAT teams. There are many ways this is beneficial in high-risk operations.

- Improved situational awareness
- Enhanced response time
- Better resource allocation

HOW CAN DATA SCIENCE HELP?

We can try out different classification models to see what yields the best results.

I will be comparing two approaches, non-graph based classification models (random forest classifier, logistic regression, KNN, etc) vs a Graph Neural Network.

MY HYPOTHESIS

A graph-based approach to predicting room types in a floor plan would outperform a non-graph classification model in terms of accuracy and generalization ability. This is because the graph-based approach can effectively capture the spatial relationships between rooms and leverage powerful graph algorithms to classify rooms based on their connectivity patterns.

2

PROJECT PLAN

PROJECT PLAN

1. Data Cleaning and Modification
2. Feature Extraction
3. GNN Model
4. Regular Models
5. Comparison

3

DATA CLEANING AND MODIFICATION

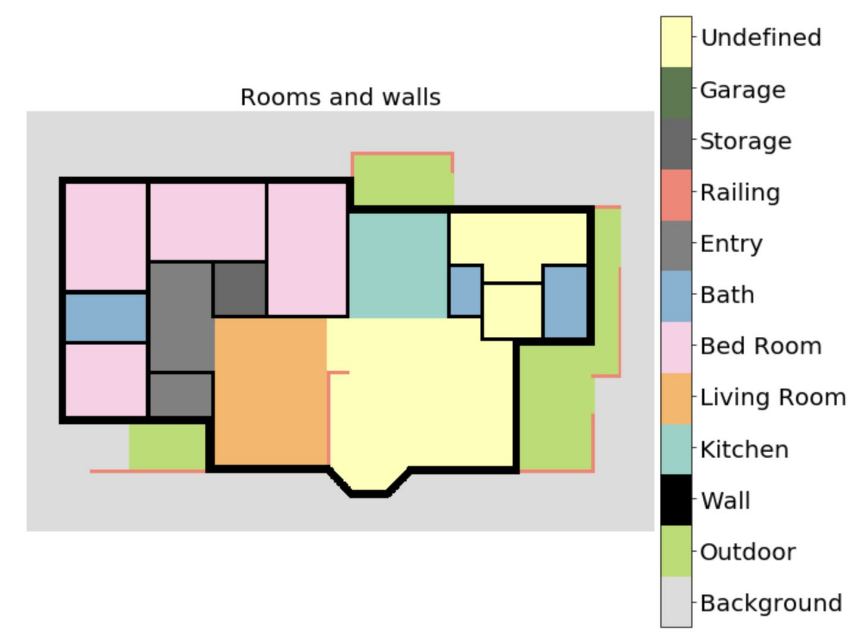
Dataset

The Cubicasa 5k dataset contains 5,000 floorplans from various building types, annotated with both room type labels (bedroom, kitchen, living room, etc.) and icon labels (electrical appliance, sauna, closet, etc.). This makes it a valuable resource for training and evaluating room type classification models.

Dataset Format

For each floorplan in the dataset, there are two corresponding 2D numpy arrays (images), one for the room layer and one for the icon layer. Both arrays are of the same size and contain numerical values that represent the room type or icon label for each pixel in the floorplan.

Dataset Format



Data Cleaning Motivation

- We want to train classification models on this dataset, but there are rooms that are undefined
- We won’t train on floor plans that contain those rooms because the undefined room class should never be predicted
- Around 90.5% of floorplans in the dataset contain undefined rooms
- We want to salvage as much of the data as possible

Data Cleaning

- My approach to identifying rooms in floorplans involves the use of OpenCV and shapely libraries
- Once the rooms are identified, we can analyze each room and count the number of icons, such as windows, doors, sinks, and other relevant features

Resulting Dataframe after processing the floorplans

Room Type	Window	Door	Closet	Electrical Applience	Toilet	Sink	Sauna Bench	Fire Place	Bathtub
0	5.0	7.0	2.0	6.0	1.0	2.0	2.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	5.0	7.0	2.0	6.0	1.0	2.0	2.0	0.0	0.0
3	0.0	0.0	0.0	3.0	0.0	1.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
7	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

ML Models

- The previous dataframe will be the input for this ML model that serves the purpose of relabeling unidentified rooms.
-

Model	Accuracy
Multinomial Naive Bayes	55.8%
Random Forest	58.6%
K Nearest Neighbors	57.5%
Multiclass Logistic Regression	56.5%

Getting Ready for the Graph

- Extract features from the room and icon images for each floorplan using OpenCV
- Create graphs for each floorplan using edges and node features
- Save a dataframe with the features for each room to use for the non-graph classification model
-

Introduction to GCN’s

Graph Convolutional Neural Network (GCN)

- Suitable for irregular data structured as graphs
- Effective in handling relational and spatial information
- Comprises of Nodes and Edges
-

GCN Architecture

- GCN Layers
  - Aggregate information from neighboring nodes
- Activation Functions
  - Introduce Non-linearity
- Training and Optimization
  - Loss function: Predictions vs Actual
  - Optimizer: Updates parameters

The big question…

Does the GCN outperform non-graph classification models in this use case?

?

Results

- It outperforms!
- Accuracy: GCN(40.74%), Logistic Regression (31.28%), Random Forest (25.51%)
- It outperformed the other classifiers in terms of precision, recall, and F1-score as well!

Improvements and Future Work

- Hyperparameter optimization
- Alternative architectures
- Additional features
-

Thank you!