Predicting House Prices Using Image Detection

Kaitlyn Chen, Ryan Guo, Jonathan Wang
GenAI Gurus

Introduction

- We developed a model for multi-step time series prediction of the price of houses based on image detection and numerical information
- Useful for people to want alternative sources other than Zillow, Redfin, etc. to estimate house prices
- Helps inform decision making on purchasing houses, home renovation, and real estate

Previous Work

- “Vision-based housing price estimation using interior, exterior & satellite images”
  - https://www.sciencedirect.com/science/article/pii/S2667305322000217
  - Academic study using deep convolutional neural networks (CNNs)
  - Estimated house prices based on attributes such as the interior & exterior of the house, and satellite images
  - Created a model with the strength of being trained on visual information

- Example Solution: https://github.com/tncy67/House-Price-Prediction-via-Computer-Vision
  - Used a CNN to predict house prices from scraped exterior frontal images
  - Average error of 4.3%

Dataset

- Our model is trained on 4 datasets, approximately ~50k exterior and interior home images, and attribute data such as city, # of bedrooms, # of bathrooms, square footage (in feet), and prices (in USD).
  - GitHub Dataset: ~2.2k exterior and interior images
  - Southern California Dataset: ~15.5k exterior images
  - House Rooms Dataset: ~5.3k interior images
  - House Rooms & Street Dataset: ~25k exterior and interior images

- Data Prep: Data cleansing, gathering into one central dataset for training

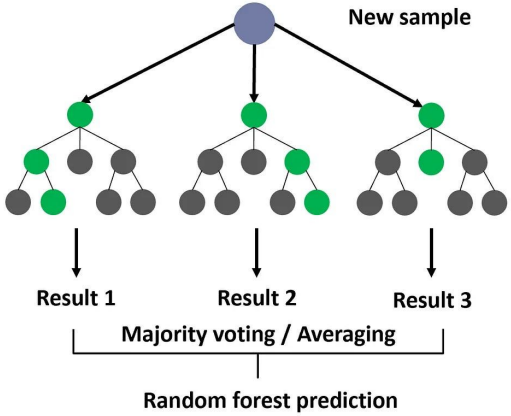
Baseline Model: Random Forest Regressor

Reduces predictive variance by creating multiple estimates of the same question

Inputs: Tabular features - numerical attributes; image features - channel-wise mean values extracted from house images

Feature Fusion: Concatenated tabular data and basic image statistics

Training: Supervised learning using combined feature set



Method

- Combine unsupervised feature learning (larger datasets) with supervised fined tuning (smaller datasets) to maximize utility of available information

- Phase 1: Learning General housing features
  - Analyze larger datasets (RobinReni + MikhailMa) to learn universal patterns such as layouts, architecture style, material

- Phase 2: Fine tuning for house prediction
  - Predict prices using smaller datasets (SoCal + GitHub dataset)
  - Pause early layers to maintain learned features

Evaluation

Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
image_input (InputLayer)	(None, 128, 128, 3)	0	–
conv2d_2 (Conv2D)	(None, 128, 128, 32)	896	image_input[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 128, 128, 32)	128	conv2d_2[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 64, 64, 32)	0	batch_normalizat…
dropout_4 (Dropout)	(None, 64, 64, 32)	0	max_pooling2d_2[…
conv2d_3 (Conv2D)	(None, 64, 64, 64)	18,496	dropout_4[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 64, 64, 64)	256	conv2d_3[0][0]
tabular_input (InputLayer)	(None, 3)	0	–
max_pooling2d_3 (MaxPooling2D)	(None, 32, 32, 64)	0	batch_normalizat…
dense_2 (Dense)	(None, 32)	128	tabular_input[0]…
dropout_5 (Dropout)	(None, 32, 32, 64)	0	max_pooling2d_3[…
batch_normalizatio… (BatchNormalizatio…	(None, 32)	128	dense_2[0][0]
flatten_1 (Flatten)	(None, 65536)	0	dropout_5[0][0]
dropout_6 (Dropout)	(None, 32)	0	batch_normalizat…
concatenate_1 (Concatenate)	(None, 65568)	0	flatten_1[0][0], dropout_6[0][0]
dense_3 (Dense)	(None, 128)	8,392,832	concatenate_1[0]…
dropout_7 (Dropout)	(None, 128)	0	dense_3[0][0]
price_output (Dense)	(None, 1)	129	dropout_7[0][0]

Total params: 8,412,993 (32.09 MB)

Overview of Neural Network Architecture

- Layer Column: Hierarchy of layers from input to output
- Layer types: Conv2, Dense, etc
- Output Shape: Shape of data after passing through the layer
- Param #: Trainable weights the layer learns
- Connected to: Shows how the layers are wired together

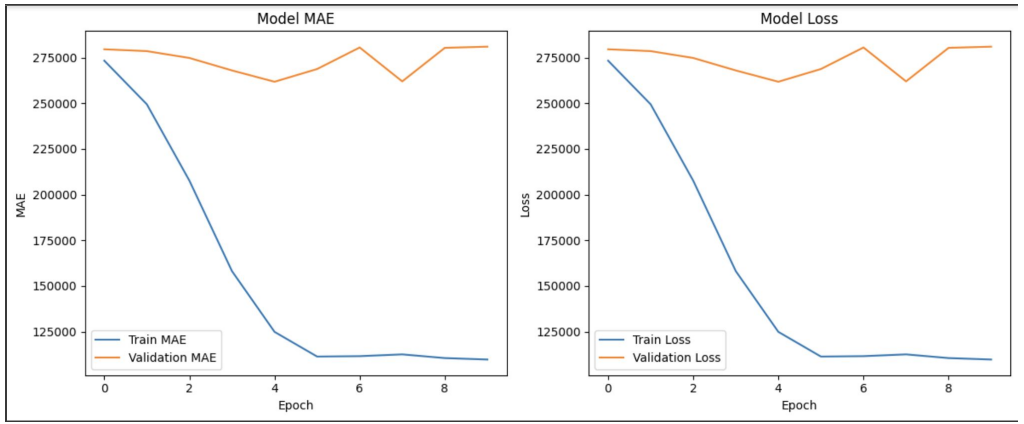
Results

- MAE: 257658.4191
- MSE: 284947.7084

```
Epoch 1/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 54s 2s/step - loss: 283572.8438 - mae: 283572.7812 - mse: 96711835648.0000 - val_loss: 279538.8750 - val_mae: 279538.7812 - val_mse: 954
Epoch 2/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - loss: 253765.0312 - mae: 253764.9844 - mse: 80741859328.0000 - val_loss: 278522.3438 - val_mae: 278522.3125 - val_mse: 948
Epoch 3/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 81s 2s/step - loss: 227586.9688 - mae: 227586.9219 - mse: 67971424256.0000 - val_loss: 274778.4062 - val_mae: 274778.3750 - val_mse: 928
Epoch 4/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 45s 2s/step - loss: 167071.7812 - mae: 167071.7344 - mse: 41629863936.0000 - val_loss: 267950.7188 - val_mae: 267950.6875 - val_mse: 891
Epoch 5/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 43s 2s/step - loss: 128790.3516 - mae: 128790.3125 - mse: 24311468032.0000 - val_loss: 261764.0469 - val_mae: 261763.9844 - val_mse: 858
Epoch 6/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 82s 2s/step - loss: 111077.1484 - mae: 111077.1250 - mse: 16901961728.0000 - val_loss: 268729.8438 - val_mae: 268729.8125 - val_mse: 895
Epoch 7/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 84s 2s/step - loss: 108837.7344 - mae: 108837.6953 - mse: 16151214080.0000 - val_loss: 280510.6875 - val_mae: 280510.6250 - val_mse: 959
Epoch 8/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 81s 2s/step - loss: 113094.7188 - mae: 113094.6641 - mse: 16794559488.0000 - val_loss: 261963.4219 - val_mae: 261963.4062 - val_mse: 859
Epoch 9/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 42s 2s/step - loss: 111523.6250 - mae: 111523.5781 - mse: 16427004928.0000 - val_loss: 280321.4688 - val_mae: 280321.4688 - val_mse: 958
Epoch 10/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 87s 2s/step - loss: 107825.6875 - mae: 107825.6562 - mse: 15725841408.0000 - val_loss: 280964.0000 - val_mae: 280963.9688 - val_mse: 962

Evaluating model...
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 259ms/step

Model Evaluation Metrics:
MAE: 261763.9994
RMSE: 292980.7733
R2: -3.9586
MAPE: 90.6905
```



Discussion

Model Interpretability: Random forest provides some feature importance, but combined features dilute interpretability, tabular features likely dominate due to richer signal compared to rudimentary image stats

Limitations: Computation power, may not generalize well to more diverse datasets or unseen environments, sensitive to image resolution, lighting conditions, and framing

Next Steps: Hyperparameter tuning to improve model performance, use multimodal architectures (e.g., late fusion, joint embeddings)

Thank You!