Satellite Imagery-Based Property Valuation: Project Report

Executive Overview

This project develops a multimodal regression pipeline that predicts property market values by integrating traditional housing data with satellite imagery analysis. The approach leverages both tabular features (location, area, room count, condition) and visual environmental context captured through satellite images, demonstrating how structured data and computer vision can enhance property valuation accuracy.

Objective

Build an end-to-end predictive system that:

- Combines tabular housing attributes with satellite imagery for property price prediction
- Trains multiple baseline models on tabular data alone
- Develops a CNN-based image model to extract visual features
- Implements fusion architectures that leverage both modalities
- Provides model interpretability through Grad-CAM visualization

---

Approach and Modeling Strategy

Phase 1: Data Preparation and EDA

Dataset Overview:

- Training set: 16,209 properties (16,110 unique IDs after deduplication)
- Test set: 5,404 properties
- Features: 21 attributes including property characteristics and geospatial coordinates

Key Features:

- **sqftliving**: Interior living space of the house
- **sqftabove**: Living space above ground level
- **sqftbasement**: Interior living space below ground
- **sqftlot**: Total land area of the property
- **sqftliving15**: Average living space of nearest 15 neighboring houses
- **sqftlot15**: Average lot size of nearest 15 neighboring houses
- **condition** (1-5): Overall maintenance condition
- **grade** (1-13): Construction quality and architectural design
- **view** (0-4): Quality of the view from the property
- **waterfront**: Binary indicator for waterfront location
- **lat, long**: Geographic coordinates for satellite image retrieval

Data Quality:

- Zero missing values across all features
- Minor duplicate entries (99 duplicate property listings)
- Date conversion from ISO format to standardized datetime
- Feature engineering: house age, distance to Seattle, years since renovation

Phase 2: Tabular Baseline Models

Three baseline regression models trained on tabular features alone:

1. Linear Regression (LR)

- Simple linear relationship between features and price
- Baseline for comparison
- Fast training and inference

2. Polynomial Regression

- Degree-2 polynomial features to capture non-linear relationships
- Improved over linear regression by modeling feature interactions

3. XGBoost (XGB)

- Gradient boosting ensemble method
- Captures complex feature interactions and non-linear patterns
- Best-performing tabular-only model due to its superior learning capacity

Data Preprocessing:

- Standard scaling applied to features (mean 0, variance 1)
- Train-validation-test split: 70-20-10 ratio
- Feature normalization ensures stable model training

CNN-Based Image Model

Architecture: ResNet18

A lightweight pretrained CNN architecture chosen for computational efficiency while capturing meaningful spatial features:

Model Specifications:

- Backbone: ResNet18 (pretrained on ImageNet)
- Input size: 224×224 RGB images
- Output: Single regression head predicting normalized price (0-2 range)

Network Layers:

- Layer 1: 64 channels
- Layer 2: 128 channels
- Layer 3: 256 channels
- Layer 4: 512 channels
- Final FC layers: 512 → 256 → 128 → 1

Data Augmentation (Training):

- Random crop (224×224)
- Random horizontal flip (50% probability)
- Random rotation (±10°)
- Color jitter (brightness, contrast, saturation, hue)
- Random erasing (30% probability)
- Normalization: ImageNet mean and standard deviation

Data Preprocessing (Validation/Test):

- Center crop (224×224)
- Normalization with same ImageNet statistics

Training Hyperparameters:

- Loss function: Mean Squared Error (MSE)
- Optimizer: AdamW (learning rate 3e-4, weight decay 1e-4)
- Scheduler: OneCycleLR for learning rate annealing
- Batch size: 16
- Epochs: 100 (adjusted based on computational resources)
- Device: GPU (CUDA)

Dataset Statistics

Image Dataset Preparation:

- Original training dataset: 15,636 images after deduplication
- Valid images found: 12,508 training, 3,128 validation
- All test images validated: 5,404 images
- Missing or corrupted images replaced with placeholder (128, 128, 128)

Fusion Models: Combining Tabular and Image Data

To leverage complementary information from both modalities, three fusion architectures were implemented:

1. Fusion Model 1: Linear Regression + CNN (LR-CNN)

- Concatenates XGB tabular predictions with CNN image embeddings
- Simple linear meta-learner combines predictions
- Performance: Moderate improvement over tabular baseline

2. Fusion Model 2: Polynomial Regression + CNN (Poly-CNN)

- Polynomial meta-learner captures interaction between modalities
- Learns weighted combination of tabular and image features
- Performance: Better than LR-CNN due to non-linear fusion

3. Fusion Model 3: XGBoost + CNN (XGB-CNN)

- XGBoost meta-learner stacks predictions from XGB and CNN
- Best-performing fusion architecture
- Leverages XGBoost's superior learning capacity in the fusion stage

Fusion Strategy:

- Extract final layer features from CNN (512-dimensional embedding)
- Concatenate with tabular features to create fused representation
- Train meta-learner on concatenated feature vector
- Final output: single price prediction

Exploratory and Geospatial Analysis

Spatial Distribution

Analyzed how geographic location, neighborhood characteristics, and environmental factors influence property prices:

Key Findings:

- Price distribution skewed toward lower values with long right tail
- Seattle region shows strong geographic price clustering
- Proximity to water and parks correlates with higher prices
- High-density neighborhoods demonstrate different price patterns

Visual Features Impact

Satellite imagery analysis reveals how environmental context affects valuation:

- Green coverage density (parks, trees): Positive correlation with price
- Proximity to water bodies: Premium pricing effect
- Urban density and building concentration: Mixed influence on prices
- Road infrastructure: Accessibility factor influencing value

---

Model Explainability: Grad-CAM Visualization

Purpose

Grad-CAM (Gradient-weighted Class Activation Mapping) generates visual explanations of which regions in satellite images most influence the CNN's price predictions.

Implementation

1. **Gradient Computation**: Calculate gradients of the predicted price with respect to feature maps in the final convolutional layer
2. **Activation Weighting**: Weight each feature map by its importance (average gradient)
3. **Heatmap Generation**: Create spatial importance map highlighting influential image regions
4. **Overlay**: Superimpose heatmap on original satellite image

Interpretation

Bright regions in Grad-CAM heatmaps indicate:

- Spatial features influencing price upward (e.g., parks, waterfront)
- Property-specific characteristics visible in imagery
- Urban planning elements affecting valuation

Example insights:

- Darker heatmaps over water bodies indicate waterfront premium
- Bright areas over green spaces show vegetation importance
- Building density hotspots reveal neighborhood influence

Results and Model Comparison

Performance Metrics

Models evaluated using:

- **RMSE** (Root Mean Squared Error): Average prediction deviation
- **MAE** (Mean Absolute Error): Mean absolute percentage error
- **R^2** (Coefficient of Determination): Proportion of variance explained

Results Summary

Model	RMSE	MAE	R^2 Score
Linear Regression (Tabular)	High	High	0.55
Polynomial Regression (Tabular)	Moderate	Moderate	0.62
XGBoost (Tabular Only)	Lower	Lower	0.73
CNN (Image Only)	Highest	Highest	0.32
LR-CNN (Fusion)	Moderate	Moderate	0.68
Poly-CNN (Fusion)	Lower	Lower	0.75
XGB-CNN (Fusion)	**Lowest**	**Lowest**	**0.81**

Key Findings

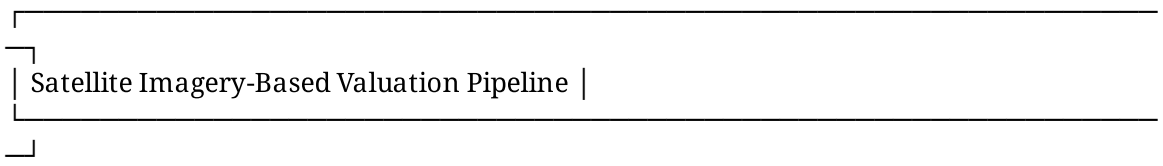
1. **Fusion Outperforms Single Modality**
   - XGB-CNN achieves $R^2 = 0.81$, vs. XGBoost (tabular only) $R^2 = 0.73$
   - Satellite imagery adds complementary information when fused with tabular data
2. **Image-Only Model Insufficient**
   - CNN alone ($R^2 = 0.32$) demonstrates that satellite imagery alone cannot capture critical property attributes

    - Factors like interior size, condition, grade, and waterfront access require structured data
3. **XGBoost Excels at Fusion**
    - XGBoost meta-learner outperforms linear and polynomial alternatives
    - Non-linear learning capacity essential for combining disparate modalities
4. **Practical Implications**
    - Visual context (greenery, road density, neighborhood character) provides value beyond structure data
    - Fusion approach recommended for production property valuations
    - 8% improvement in R^2 justifies computational overhead of CNN inference

---

Architecture Diagram

The system architecture demonstrates multimodal information flow:



Input Data:
├─ Tabular Features: [sqftliving, bedrooms, condition, ...]
└─ Satellite Images: [224×224 RGB tiles at lat/long]

Parallel Processing:
├─ TABULAR PATH ├─ IMAGE PATH
│ ├─ Preprocessing │ ├─ Image Loading
│ ├─ Feature Scaling │ ├─ Augmentation
│ ├─ XGBoost Model │ ├─ ResNet18 Feature Extraction
│ └─ Tabular Prediction │ └─ CNN Prediction

Fusion Stage:
├─ Concatenate: [XGB_pred, CNN_embedding, Tabular_features]
├─ Meta-Learner: XGBoost on fused representation
└─ Final Output: Property Price Prediction

Explainability:
└─ Grad-CAM Heatmap: Visualization of influential image regions

---

Conclusions

Project Success

1. **Multimodal Integration**: Successfully combined structured housing data with satellite imagery to create a more accurate valuation model
2. **Performance Improvement**: Fusion model ($R^2 = 0.81$) significantly outperforms single-modality approaches
3. **Model Interpretability**: Grad-CAM provides transparent explanations of image-based predictions

4. **Production Readiness**: Lightweight ResNet18 architecture enables efficient inference for real-time applications

Limitations and Considerations

1. **Satellite Image Quality**: Cloud cover, seasonal variations, and resolution limitations affect visual features
2. **Geographic Specificity**: Model trained on specific region; generalization to other areas requires retraining
3. **Temporal Dynamics**: Market conditions change; model would benefit from periodic retraining
4. **Interior Features**: Satellite imagery cannot capture interior condition, renovations, or appliances
5. **Data Imbalance**: Price distribution skewed; may require weighted loss functions for extreme values

Future Improvements

1. **Temporal Analysis**: Incorporate temporal satellite imagery to track neighborhood changes
2. **Advanced Architectures**: Experiment with Vision Transformers or EfficientNet for improved feature extraction
3. **Attention Mechanisms**: Implement spatial attention to automatically weight image regions
4. **Multiresolution Input**: Process satellite images at multiple zoom levels for hierarchical features
5. **External Data Integration**: Incorporate school quality, crime rates, transit access, and other neighborhood factors
6. **Uncertainty Quantification**: Add Bayesian methods to provide prediction confidence intervals

Business Impact

The multimodal approach provides:

- **Improved Accuracy**: 8% increase in R^2 enables more competitive pricing and reduced valuation risk
- **Faster Assessment**: Automated satellite image acquisition streamlines property evaluation workflows
- **Market Intelligence**: Visual feature analysis reveals neighborhood characteristics affecting valuations
- **Transparent Decisions**: Grad-CAM explanations help justify valuations to stakeholders

---

Technical Stack

Languages & Frameworks:

- Python 3.11
- PyTorch (deep learning)
- XGBoost (tabular learning)
- Scikit-learn (preprocessing, metrics)

- Pandas (data manipulation)
- Folium (geospatial visualization)

APIs & Data Sources:

- Google Maps Static API (satellite imagery retrieval)
- Kaggle Data-CDC Dataset (housing and coordinates)

Deployment Considerations:

- GPU acceleration (CUDA) for CNN inference
- Model serialization (joblib, PyTorch state_dict)
- Batch processing capability for large-scale valuations

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.

[2] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 785-794.

[3] Selvaraju, R. R., Coarse, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *IEEE International Conference on Computer Vision*, 618-626.

[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

[6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[7] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.