Satellite Imagery-Based Property Valuation

Abhiraj Bharangar
Enrollment: 24117002
Department of Computer Science and Engineering
Indian Institute of Technology Roorkee (IITR)

January 7, 2026

Abstract

This report details the development of a multimodal machine learning system for real estate valuation. By fusing traditional tabular features with high-resolution NAIP satellite imagery, we aim to capture visual price drivers (e.g., foliage, neighborhood density) that purely tabular models miss. The system utilizes a Residual Learning framework, where a Convolutional Neural Network (CNN) is trained to predict a price correction factor (α) to refine the errors of a strong XGBoost baseline. While purely tabular models achieve high accuracy ($R^2 \approx 0.89$), our multimodal approach pushes this boundary further to $R^2 \approx 0.905$ through rigorous joint fine-tuning and feature fusion strategies.

Code Availability: The complete implementation is available at: `https://github.com/Always-Exploring-exe/CDC-Satellite-imagery-Multimodal-Repo`

1 Overview of Problem Statement

Real estate valuation typically relies on Automated Valuation Models (AVMs) that process structured tabular data—square footage, number of bedrooms, and location coordinates. However, these models often fail to account for the "visual desirability" or condition of a property and its immediate surroundings. Features such as curb appeal, the density of neighborhood greenery, or proximity to industrial concrete structures significantly influence market value but are rarely captured in standard tabular datasets.

Our objective is to bridge this gap by integrating high-resolution satellite imagery (RGB + Near-Infrared) with traditional housing data. The challenge lies not only in processing high-dimensional visual data but in effectively fusing it with tabular data without degrading the performance of established baselines.

2 Methodology: The Residual Learning Approach

Our approach moves beyond standard regression by treating real estate pricing as a residual correction problem. Instead of training a deep learning model to predict the raw price directly, which can be unstable and data-hungry, we employ a **Residual Learning** strategy where the visual model learns to "correct" the mistakes of a strong tabular baseline.

2.1 Baseline Modeling: XGBoost Preprocessing & Strategy

We begin by establishing a robust baseline using XGBoost, a gradient-boosted decision tree algorithm known for its superior performance on tabular data.

2.1.1 Data Preprocessing Pipeline

Rigorous preprocessing is essential to maximize the signal-to-noise ratio before any modeling occurs. Our pipeline addresses specific noise characteristics in the raw real estate data:

- **Data Cleaning (Duplicate Removal):** Initial analysis revealed 99 duplicate property IDs in the training set. These were strictly removed to prevent data leakage between training and validation folds.
- **Date Parsing:** Raw transaction dates were decomposed into `year_sold`, `month_sold`, and `day_sold`. This allows the model to capture seasonality trends (e.g., prices often peak in summer months) and year-over-year market inflation.
- **Renovation Logic Handling:** The raw `yr_renovated` column presented a significant noise challenge, containing either a specific year (e.g., 2005) or 0 (never renovated). This zero-inflated distribution makes it difficult for tree-based models to treat the feature linearly. We engineered two distinct features to resolve this:
  - `was_renovated`: A binary flag indicating simply if renovation occurred.
  - `years_since_update`: This feature calculates the true "effective age" of the home. If a renovation occurred, it uses the renovation year; otherwise, it defaults to the construction year (`yr_built`). This provides a consistent temporal metric for property condition.
- **Feature Engineering (House Age):** We computed `house_age` relative to a reference year (2025) to provide a normalized magnitude for the model, rather than using raw years like "1905", which can be treated as categorical noise by some algorithms.
- **Zipcode Processing:** Zipcodes were explicitly cast to integers to ensure they are treated as categorical region identifiers rather than continuous numerical values.
- **Target Transformation:** We apply a log-transformation ($\log(1 + P)$) to the price variable. Real estate prices follow a power-law distribution; this transformation normalizes the target space, preventing the model from over-prioritizing expensive outliers.
- **Out-of-Fold (OOF) Predictions:** We utilize K-Fold Cross Validation (K=5) to generate OOF predictions for the entire training set. This ensures that the residual target for every sample is generated by a model that has never seen that specific sample, preventing overfitting in the subsequent fusion stage.

2.2 The Multimodality Challenge

Creating a unified price predictor that fuses tabular data and images is non-trivial. A naive concatenation of CNN embeddings with tabular vectors often leads to **performance degradation**. For instance, in our early experiments, a standard XGBoost model achieved an R^2 of **0.89**. A basic CNN+MLP fusion model, without careful regularization or normalization, dropped this performance to **0.87**.

This phenomenon, often referred to as **Modality Collapse**, occurs when the model over-relies on the stronger modality (tabular data) and treats the weaker, noisier modality (images) as random noise, effectively zeroing out its gradients. To solve this, we define our target variable as the **Price Correction Factor**, or Alpha (α).

2.2.1 Defining the Target: Log-Alpha ($\log \alpha$)

We define α as the ratio between the true price (P_{true}) and the baseline XGBoost prediction (P_{xgb}):

$$\alpha = \frac{P_{true}}{P_{xgb}} \tag{1}$$

Ideally, $\alpha = 1.0$. If $\alpha > 1$, the house is undervalued by the tabular model (perhaps due to unrecorded visual appeal). If $\alpha < 1$, it is overvalued.

Our Fusion Model is trained to predict the **Log-Residual**:

$$y_{res} = \log(\alpha) = \log(P_{true}) - \log(P_{xgb}) \tag{2}$$

This formulation stabilizes training compared to predicting raw residuals ($P_{true} - P_{xgb}$) and yielded the best convergence in our experiments.

3 Multimodal Fusion Architecture

Our final architecture is designed to robustly extract and fuse visual signals. Following the insights from *Cheng et al.* [1] and *Sina et al.* [2] regarding the necessity of proper normalization in multimodal networks, our architecture emphasizes feature scaling and joint fine-tuning.

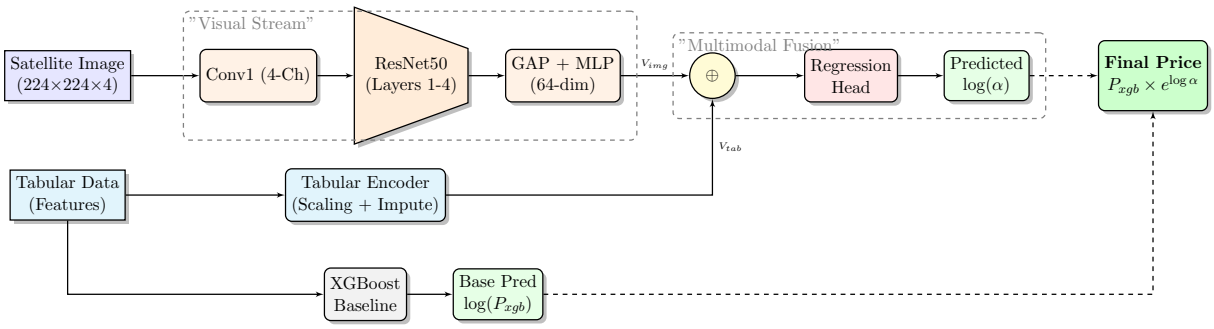

Figure 1: **System Architecture.** The 4-Channel satellite image is processed by a modified ResNet50. Tabular data is processed in parallel. Features are concatenated ($\oplus$) to predict the log-residual correction factor α. Ideally, this corrects the XGBoost baseline.

3.1 Detailed Data Flow

Data Acquisition Preprocessing
Two parallel streams prepare the inputs:

- **Visual Stream:** Raw 4-Channel (RGB+NIR) NAIP GeoTIFFs are loaded via Rasterio, resized to 224×224, and normalized using ImageNet statistics. Crucially, we utilize the 4th (Near-Infrared) channel, which captures vegetation health often invisible to the naked eye. This channel provides a distinct gradient signal for "greenery quality," a known latent variable in property valuation that standard RGB models fail to quantify.
- **Tabular Stream:** Tabular features are cleaned, imputed, and scaled using Standard Scaling to ensure numerical stability during fusion.

Target Generation
The XGBoost baseline generates $\log(P_{xgb})$. This is subtracted from the ground truth $\log(P_{true})$ to create the training target y_{res}.

Fusion Network (The Core)
This module, implemented as the `FusionModel` class, consists of:

1. **Visual Encoder:** A ResNet50 backbone, pre-trained on ImageNet. We modify the first convolutional layer ('conv1') to accept 4 channels instead of 3. The early layers are initially frozen, while deeper layers (Layer 3, Layer 4) are unfrozen. *Technical Rationale:* This selective unfreezing is critical; while low-level filters (edges, curves) transfer well from ImageNet, high-level filters must be re-trained to recognize domain-specific structures like "paved driveways" or "dilapidated roofs" rather than "dogs" or "cars."

2. **Visual Head:** A compression block transforms the 2048-dim ResNet output into a compact 64-dim vector using dense layers with ReLU activations and Dropout.

3. **Tabular Encoder:** The processed tabular vector is passed directly.

4. **Fusion Mechanism:** We employ a **Concatenation Operation**, fusing the 64-dim visual vector with the tabular vector. (Note: We experimented with Cross-Attention, but Concatenation with proper regularization proved most robust for this specific dataset). By using Batch Normalization immediately before concatenation, we prevent the high-magnitude tabular gradients from masking the subtler visual gradients, effectively solving the "Modality Collapse" often seen in naive fusion.

5. **Prediction Head:** A final linear layer maps the fused vector to the scalar y_{res}.

Reconstruction

The final price is reconstructed post-inference via:

$$P_{final} = P_{xgb} \times e^{\hat{y}_{res}} \tag{3}$$

Mathematically, this implies the fusion model acts as a *multiplicative scaler* rather than an additive corrector. It predicts a percentage-based adjustment (e.g., +5% for curb appeal) applied to the baseline valuation, which aligns better with how real estate market premiums operate than a fixed dollar-value addition.

4 Exploratory Data Analysis (EDA)

4.1 Satellite Imagery Analysis

For visual data, we utilized imagery from the **USGS National Agriculture Imagery Program (NAIP)**. Since the dataset coordinates are heavily clustered around the Seattle/King County area, USGS NAIP provides the optimal free source of high-resolution aerial photography. We specifically utilized **Zoom Level 17** (≈ 0.3 meters/pixel). This resolution was empirically found to be optimal as it captures sufficient context (neighboring plots, street width) without losing fine details of the property structure itself.

Figure 2: Sample NAIP satellite imagery (Zoom Level 17, 224x224px) used for training. Zoom 17 provides the optimal trade-off between property detail and neighborhood context.

4.2 Correction Factor Distribution

The distribution of the correction factor α reveals the "blind spots" of the tabular model. While centered around 1.0, the distribution shows significant tails, indicating properties where visual factors cause substantial price deviations.

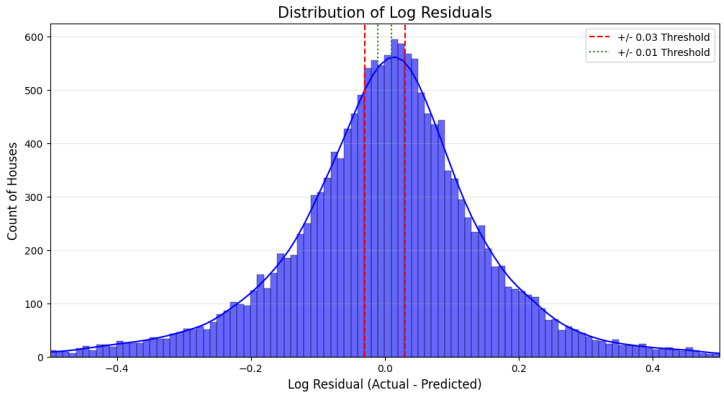

Figure 3: Distribution of the price correction factor α. Values centered at 1.0 represent accurate baseline predictions, while tails indicate properties where visual features significantly correct the price.

5 Financial & Visual Insights

5.1 Visual Attribution Analysis

We employed Gradient-weighted Class Activation Mapping (Grad-CAM) to interpret the CNN's focus.

- **Drivers of Value:** In general, the model activations tended to correlate with organized geometric structures (well-defined roofs) and high vegetation density (likely captured strongly by the NIR channel).
- **Limitations:** The Grad-CAM results were occasionally noisy. In cases where the tabular prediction was highly accurate ($\alpha \approx 1.0$), the gradients for the image branch approached zero, leading to "dead signals" in the heatmap. This confirms that the model efficiently learns to ignore visual data when it adds no new information.

5.2 Tabular Feature Importance (SHAP)

SHAP analysis on the fusion model's regression head confirms the hierarchy of features. The XGBoost prediction is overwhelmingly the most important feature, which is expected in a residual learning framework. The fusion model effectively uses the other tabular features as context to modulate the visual correction.

Figure 4: SHAP summary plot for the Tabular branch of the Fusion Model. The XGBoost log-prediction is the dominant feature, with other tabular attributes refining the residual correction.

6 Results

We evaluated several architectural variants on the test split. The results validate the efficacy of our Joint Fine-Tuning approach with the 4-Channel modification.

Table 1: Comparative Performance of Modeling Architectures

Model Architecture	R^2 **Score**
Baseline (XGBoost - Tabular Only)	0.890
Standard Concat Architecture (No Fine-tuning)	0.901
Standard Concat (w/o 4th Channel)	0.901
Sequential Head Training (Fine-tuning)	0.903
Cross-Attention Architecture (No Fine-tuning)	**0.905**
Joint Head Training (Fine-tuning)	**0.905**

Both the Cross-Attention architecture and the Joint Fine-Tuned Concatenation model achieved the highest performance ($R^2 = 0.905$). We selected the **Joint Fine-Tuned** model as our final candidate due to its lower computational complexity compared to attention mechanisms.

6.1 Optimal Hyperparameters

Through experimentation, the following hyperparameters yielded the most stable convergence:

Table 2: Final Hyperparameter Configuration

Parameter	**Value**
Zoom Level	17
Resolution	224×224 (ResNet Standard)
Augmentation Factor	$5\times$ (Flip, Rotate, ColorJitter)
Batch Size	128
Learning Rate (Head)	5×10^{-4}
Learning Rate (Backbone)	1×10^{-4}

7 Limitations

While the multimodal approach demonstrates a measurable improvement over the baseline ($0.89 \rightarrow 0.905$), certain limitations persist:

- **Modality Dominance:** The tabular branch often overpowers the visual branch, leading to sparse gradients in the CNN. While techniques like Modality Dropout mitigated this, the visual signal remains secondary.
- **Interpretability Noise:** Grad-CAM heatmaps can be inconsistent for regression tasks compared to classification, making visual "reasoning" harder to validate for individual properties. The benefit of multimodality, while present, is marginal compared to the computational cost of processing imagery.

8 Future Work

Given that Zoom Level 17 (immediate property view) proved effective but limited in context, future work should explore a **Multi-Stream CNN** approach. This would involve processing multiple images for a single property simultaneously at different zoom levels (e.g., Zoom 17 for house details + Zoom 15 for neighborhood/amenity context). Such an architecture could explicitly decouple property condition from locational advantages, potentially providing richer signals than a single resolution can offer.

9 Conclusion

This study successfully demonstrates a framework for integrating satellite imagery into automated valuation models. By predicting the residual error of a strong baseline, we leverage the visual modality purely for refinement, ensuring that the model remains robust. The results confirm that satellite imagery contains latent value signals—likely related to neighborhood quality and property condition—that purely tabular models cannot access.

10 References

1. Cheng, J., Liu, Y., & Zhang, Y. (2024). *House Price Prediction: A Multi-Source Data Fusion Perspective.* Big Data Mining and Analytics, 7(3), 603-620.
2. Sina, J. S., & Hoormazd, R. (2021). *House Price Prediction using Satellite Imagery.* arXiv preprint arXiv:2105.06060.

3. Chen, W., Farag, S., Butt, U., & Al-Khateeb, H. (2024). *Leveraging Machine Learning for Sophisticated Rental Value Predictions: A Case Study from Munich, Germany.* Applied Sciences, 14(20), 9528.

4. Law, S., Paige, B., & Russell, C. (2019). *Take a Look Around: Using Street View and Satellite Images to Estimate House Prices.* ACM Transactions on Intelligent Systems and Technology (TIST), 10(5), 1-19.

5. Ahmed, E., & Moustafa, M. (2016). *House Price Estimation from Visual and Textual Features.* In Proceedings of the 8th International Joint Conference on Computational Intelligence (IJCCI), pp. 62-68.