Indian Institute of Technology,Roorkee

CDC X Yhills OPEN PROJECTS 2025-2026

Satellite Imagery-Based Property Valuation

Project Report: Multimodal AI for Real Estate Valuation

CONTACT INFO:

1.**Name** : Lakshya Gupta

2.**Contact Number** : 9837348350

3.**Branch**: Geological Technology

4.**Email ID**: lakshya_g@es.iitr.ac.in

5.**Enrollment number:** 24410015

1. Overview

1.1 Introduction

Real estate represents one of the cornerstones of the global economy and constitutes the single largest asset class by value. Traditional AVMs, however, rely exclusively on structured data. Older algorithms, for instance, ingest hard specifications such as square footage, number of bedrooms, and lot size to regress a market price.

1.2 Objective

The primary objective of this project is to engineer a robust **Multimodal Regression Pipeline** that synthesizes two distinct data modalities to predict property value (Target: Price).

1. **Tabular Modality:** Structured quantitative data representing legal and physical specifications.
2. **Visual Modality:** Unstructured RGB satellite imagery (224 * 224 pixels) capturing environmental context.

The objective is to show that there are indeed recoverable economic signals in satellite pixels (e.g., “Greenery correlates with Wealth”) and that a neural network can learn to adjust table values based on visual observations to overcome the Condition Gap.

1.3 Modeling Strategy: Late Fusion

We followed a Late Fusion (Decision-Level Fusion) strategy. This strategy is more effective in multimodal problems where the sources of the input data have strongly differing properties: the image data is high-dimensional and sparse, while the table data is low-dimensional and dense.

1. **Visual Stream:** Employs Transfer Learning using ResNet18 CNN, pre-trained on ImageNet to extract abstract features (texture, density, edges) in 512 dimensions.
2. **Tabular Stream:** Takes as input a set of 26 engineered financial features and processes them with a deep Multi-Layer Perceptron (MLP) along with Batch Normalization.
3. **Fusion Mechanism:** The high-level embeddings from the two streams are combined only at the second-to-last layer. This is to make the network learn the optimal representation for each modality separately, before weighing the visual context dynamically against the financial specifications.

2. Data Engineering & Preprocessing

The quality of the machine learning model strictly depends on the quality of the input data. A robust pipeline has been implemented to clean the raw transaction data and match it with the images in the database.

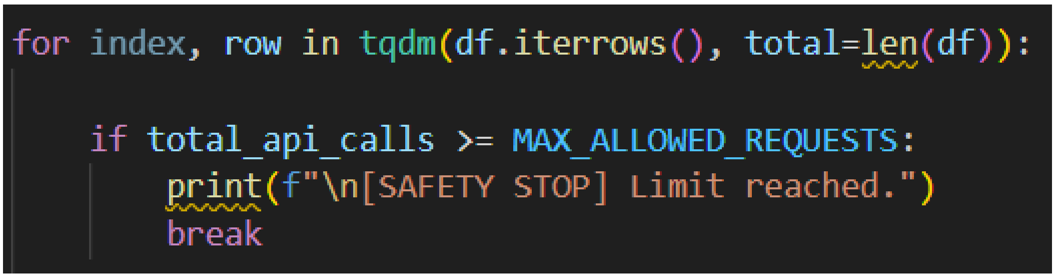
2.1 Programmatic Image Acquisition

We utilized the **Mapbox Static Images API** to convert property coordinates (Latitude/Longitude) into high-resolution satellite tiles. We specifically selected **Zoom Level 18** because it provides the optimal balance between property-level detail (e.g., roof material, backyard size) and neighborhood context (e.g., road density, green cover).

2.2 Resilience & Credit Protection Logic

To ensure the acquisition pipeline was robust and cost-effective, we implemented a three-layer protection system:

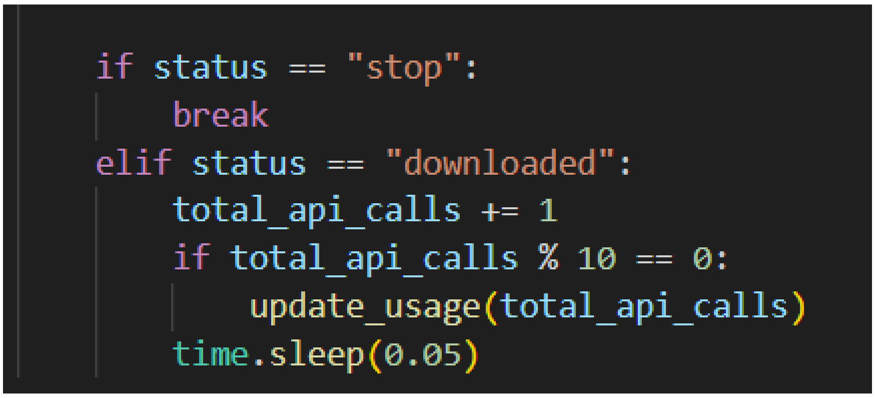
1. **Credit Monitoring:** We implemented a dynamic counter with a safety buffer of 2,000 requests (MONTHLY_LIMIT - 2000). This ensured the project remained strictly within the Mapbox Free Tier (50,000 requests/month), preventing unexpected billing.



```
for index, row in tqdm(df.iterrows(), total=len(df)):

    if total_api_calls >= MAX_ALLOWED_REQUESTS:
        print(f"\n[SAFETY STOP] Limit reached.")
        break
```

2. **Rate Limiting:** A time.sleep(0.05) delay was introduced between API calls to respect rate limits and prevent 429 Too Many Requests or 403 Forbidden errors



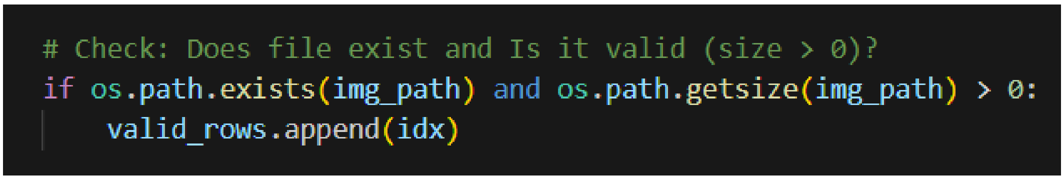
```
    if status == "stop":
        break
    elif status == "downloaded":
        total_api_calls += 1
        if total_api_calls % 10 == 0:
            update_usage(total_api_calls)
        time.sleep(0.05)
```

3. **Resume Capability:** An os.path.exists check was added to skip images already present on the disk ("cache hit"), protecting against network interruptions and avoiding redundant API calls.

2.3 Data Integrity Audit

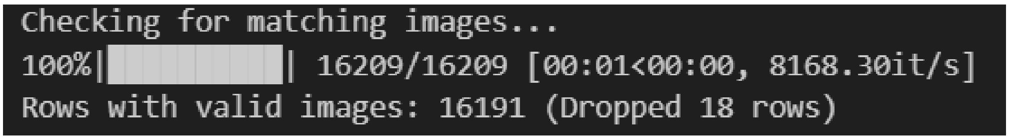
Once the images were acquired, we performed a rigorous audit to synchronize the tabular and visual datasets.

1. **Image Validation**: We iterated through the dataset to verify that for every transaction ID, a corresponding {id}.jpg existed and was not corrupt (file size > 0).



```
# Check: Does file exist and Is it valid (size > 0)?
if os.path.exists(img_path) and os.path.getsize(img_path) > 0:
    valid_rows.append(idx)
```

2. **Dataset Size**: The raw dataset contained 16,209 rows. After filtering out 18 rows with missing or corrupt images, the final clean dataset consisted of 16,191 rows (df_clean). This ensures that every input to the Neural Network is valid, preventing training crashes**.**



```
Checking for matching images...
100%|██████████| 16209/16209 [00:01<00:00, 8168.30it/s]
Rows with valid images: 16191 (Dropped 18 rows)
```

2.4 Outlier Removal

Real estate data typically contains extreme outliers that can destabilize Mean Squared Error (MSE) loss functions. We applied domain-specific constraints:

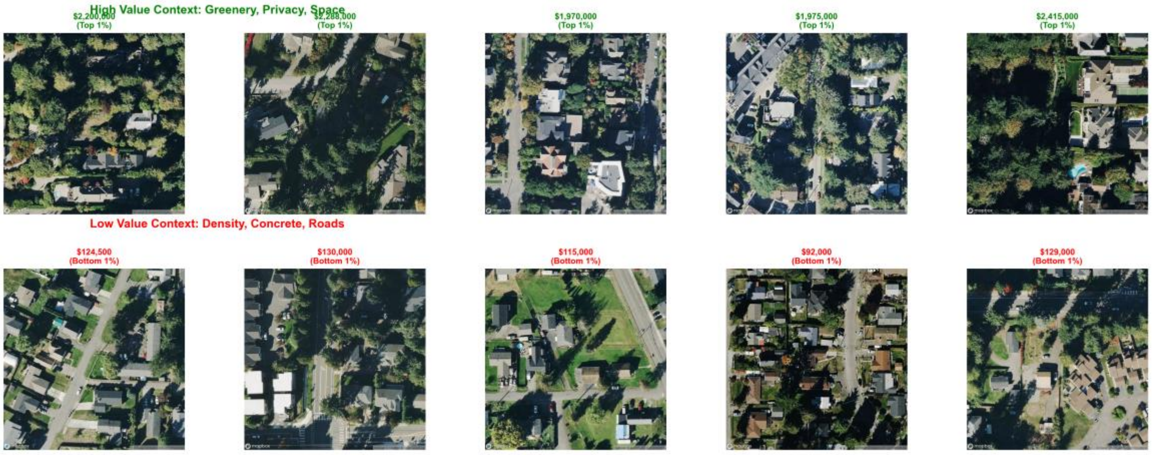


Figure 2.2 Observation: The visual audit reveals a clear contrast: High-value properties (Top) are defined by **greenery and privacy**, while low-value properties (Bottom) are characterized by **high structural density and pavement**.

1. **Bedroom Constraints:** We filtered the dataset to include only properties with 0 < bedrooms < 11. This removed data-entry errors (e.g., properties listed with 33 bedrooms).
2. **Price Capping:** We capped the dataset at **$7.5 Million**. Ultra-luxury properties tend to obey non-linear pricing patterns (historical significance and art value), which are not easily generalizable, and their sheer prices might result in gradient explosion problems.

2.4 Target Transformation

Real estate data typically follows a **Power Law distribution**—a massive "long tail" of expensive homes and large lots. Neural Networks, however, converge significantly faster when inputs and targets are Normally Distributed (Gaussian).

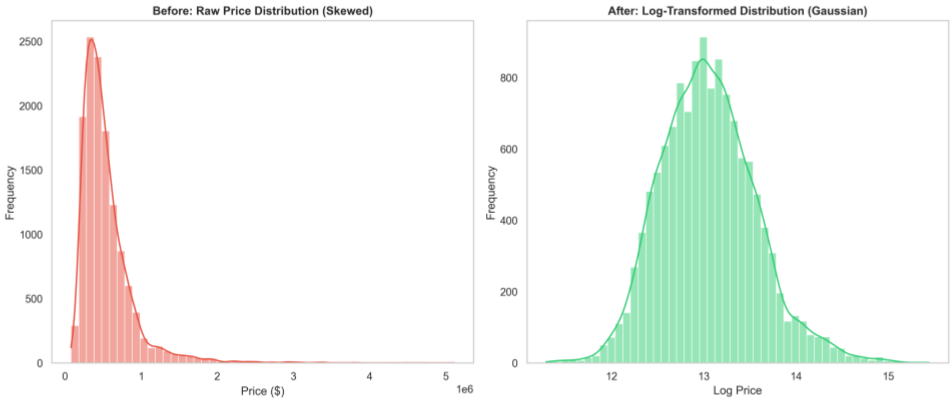


Figure 2.1: Transformation of the Target Variable. Left: The raw price distribution is highly skewed. Right: The log-transformed price approximates a Gaussian Bell Curve, ideal for Deep Learning.

Log-Transformation:

We applied the np.log1p (Natural Logarithm of) function to the target variable (price) and the following continuous features(**sqft_living, sqft_lot, sqft_above, sqft_living15 & sqft_lot15**)

This transformation compresses the target range (e.g., transforming [$75k, $7.5M] to roughly [11.2, 15.8]). This prevents the model from prioritizing errors on expensive properties over affordable ones and ensures a stable gradient descent.

2.5 Feature Scaling

Neural Networks are highly sensitive to the magnitude of input features. A feature like **sqft_lot** (values ~50,000) would naturally dominate the gradients over a feature like **lat** (values ~47) simply due to scale.

Solution : We applied **StandardScaler** to all 26 input features. This transformed the data to have a **Mean of 0** and a **Standard Deviation of 1**. This normalization is critical for the convergence of the Multi-Layer Perceptron (MLP).

---

3. Advanced Feature Engineering

Raw data (Latitude, Longitude, Date) is rarely predictive on its own. We engineered **26 advanced features** categorized into four groups to expose latent value drivers to the model.

3.1 Spatial & Location Features

Location is the primary driver of real estate value. We utilized both raw coordinates and engineered "wealth proxies" to capture neighborhood desirability.

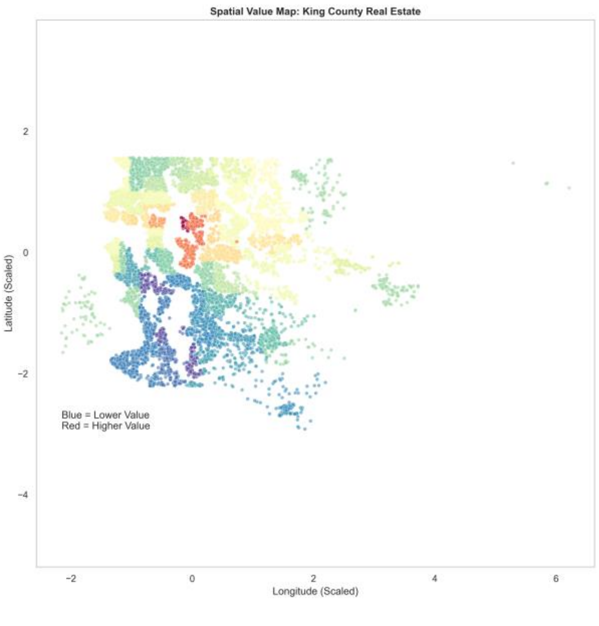


Figure 3.1: Geospatial distribution of property values. The map clearly identifies high-value clusters (Red) near waterfronts and city centers versus lower-value zones (Blue), validating the spatial features.

1. **Raw Coordinates:** lat, long.
2. **Distance to City Center Log** (dist_to_city_center_log): We calculated the Euclidean distance from each property to the city center (mean lat/long of the dataset). This

captures the "Urban Decay" effect, where price per square foot typically decreases as distance from the economic hub increases.

$$\text{Distance} = \sqrt{(\text{lat} - \text{CityLat})^2 + (\text{long} - \text{CityLong})^2}$$

3. **Zip Code Wealth** (zip_wealth): Target encoded variable for the average log-price of the zip code. The model would be able to rank neighborhoods by affluence on the fly (for example, a wealthy suburb compared to a developing area).

4. **Cluster Wealth** (cluster_wealth):Zip codes are often too large to capture street-level nuances. We applied K-Means Clustering (k=50) on the coordinates to generate 50 "micro-neighborhood". We then calculated the average wealth of each cluster, capturing value pockets like gated communities.

3.2 Temporal & Seasonality Features

Real estate has a strong seasonal component known as the "Spring Rush."

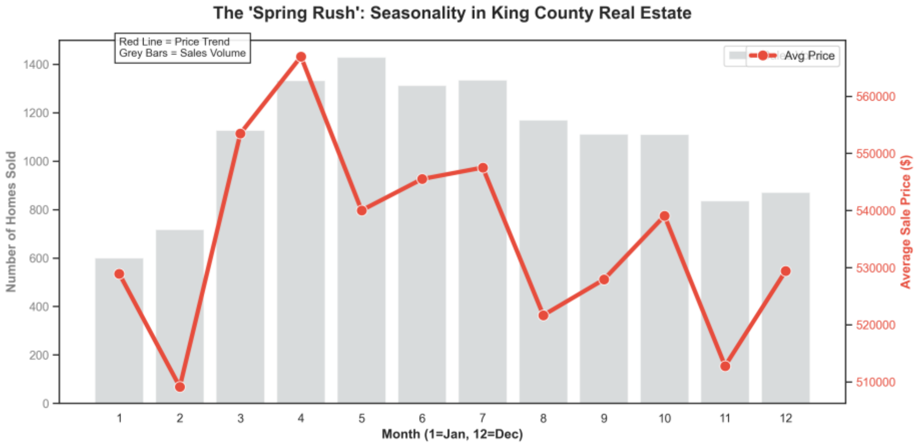


Figure 3.3: The "Spring Rush" Phenomenon. The bar chart (Volume) and line chart (Price) both show a clear peak between April and August, validating the need for non-linear time features.

1. **Sale Year (**sale_year**):** Captures long-term market appreciation/inflation.

2. **Cyclical Month Encoding** (month_sin, month_cos): Using integers (1–12) for months is mathematically flawed because it implies December (12) is far from January (1). We projected the month onto a unit circle:

$$x_{sin} = \sin\left(\frac{2\pi \times \text{Month}}{12}\right), \quad x_{cos} = \cos\left(\frac{2\pi \times \text{Month}}{12}\right)$$

This preserves the temporal proximity of winter months.

3.3 Structural & Log-Transformed Features

Standard structural features were log-transformed to handle skewness and improve convergence.

1. **Basic Specs:** bedrooms, bathrooms, floors.
2. **Quality Metrics:**
    1. **grade**: Construction quality (1–13 scale).
    2. **condition**: Maintenance level (1–5 scale).
    3. **view**: Quality of view (0–4 scale).
3. **Log-Size Features:**
    1. **sqft_living_log**: Interior living space.
    2. **sqft_lot_log**: Land area.
    3. **sqft_above_log**: Space above ground.
    4. **sqft_living15_log & sqft_lot15_log**: The average size of the 15 nearest neighbors (capturing neighborhood density context).

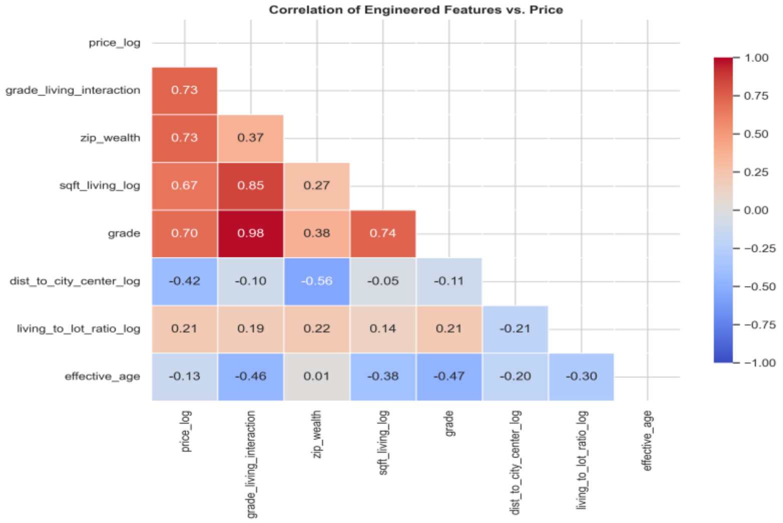


Figure 3.2: Correlation matrix of engineered features. Note the high correlation between zip_wealth and price_log, confirming location is the strongest predictor.

Note on Feature Selection (Except sqft_basement):

sqft_basement was intentionally removed from the final set of features as a precaution against perfect multicollinearity. In this data, Sqft Living = Sqft Above + Sqft Basement. The presence of all three would establish a linear dependency, confusing the model about how to weigh these features. The model could learn this using sqft_living and sqft_above, implicitly incorporating basement size as the difference, as opposed to redundancy, or "zero inflation" issues due to a 0 basement size measurement being very common.

3.4 Advanced Ratios & Interactions

We engineered specific interactions to expose non-linear value drivers.

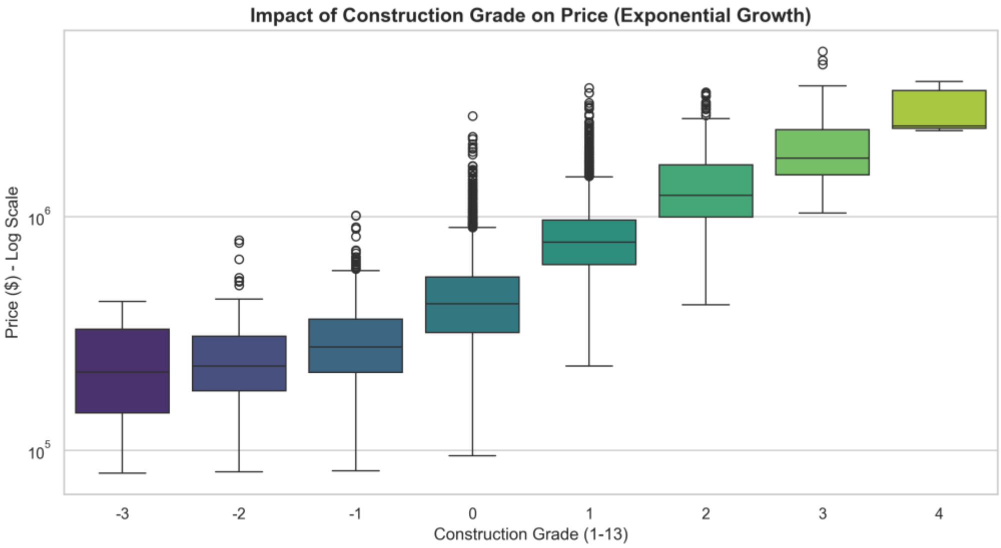
1. **Density Ratio (living_to_lot_ratio_log):** Log of (Living / Lot). High ratios indicate urban density; low ratios indicate estate privacy.a
2. **Neighbor Comparison (neighbor_comparison_log):** Log of (Subject Living / Neighbor Living). Identifies if a house is "overbuilt" for its neighborhood.
3. **Room Spaciousness (sqft_per_bedroom_log):** Log of (Living / Bedrooms).Distinguishes between cramped layouts and luxury suites.
4. **Size Difference (size_diff_from_neighbors):** Raw square footage difference.
5. **Effective Age (effective_age):** Calculated as Current Year - max(Year Built, Year Renovated). Corrects for renovations.
6. **Renovation Impact (renovation_impact):** Weighted score for recent updates.
7. **Grade-Living Interaction (grade_living_interaction):** Grade * log(Sqft Living) . Captures the multiplicative value of high quality applied to a large size.

---

4. Exploratory Data Analysis (EDA)

By doing a focused visual data audit, we ensured that the assumptions regarding the price distributions, seasonality, and the quality of the visual signals that we made were correct. It is necessary to do such a deep data analysis to make sure that the approaches used in feature-engineering are evidence-based.

4.1 The "Grade" Multiplier



A boxplot analyzing Price distribution across Construction Grades (1-13).

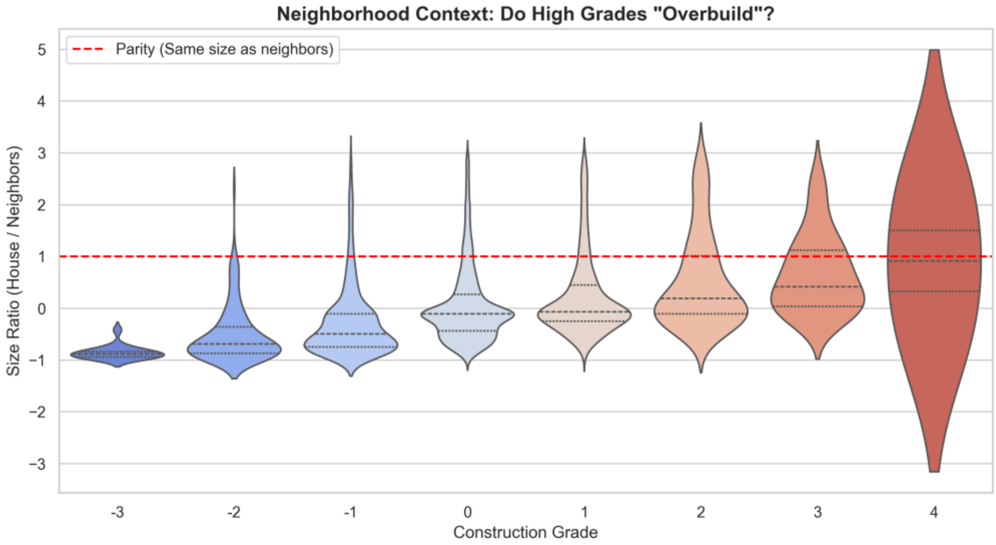
Insight: The relationship between Grade and Price is **exponential**, not linear.

Grades 1-7: Prices remain relatively flat and compressed.

Grades 8-13: Valuation explodes upwards with high variance.

Conclusion: This justifies using grade as a multiplicative interaction term (grade * sqft) rather than just an additive feature.

4.2 Neighborhood Context: The "Overbuilding" Penalty



A violin plot comparing the neighbor_comparison ratio (Subject House Size / Neighbor Size) against Grade. The red dotted line represents parity (House is same size as neighbors).

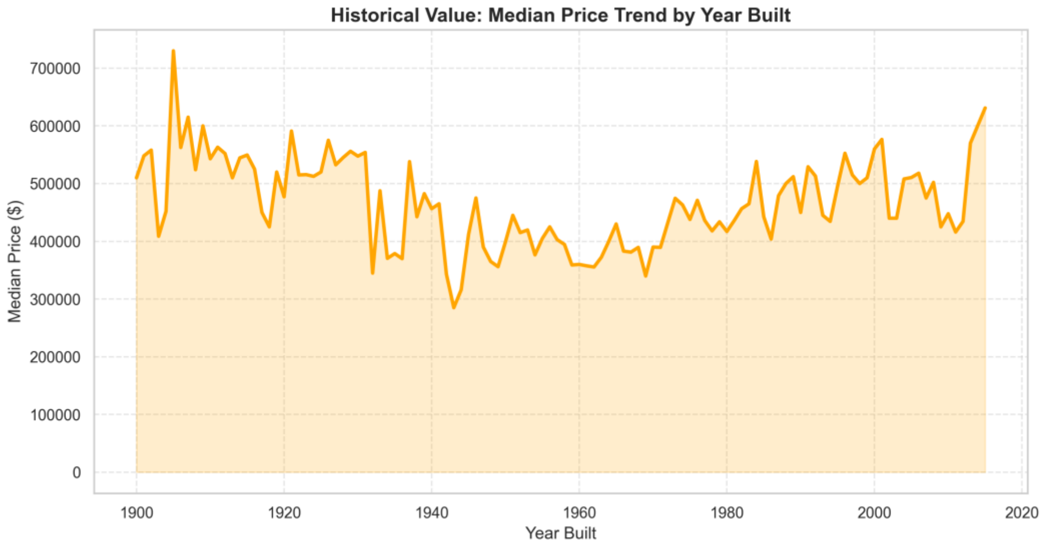
Insight:

Lower Grades (<7): Often show a ratio > 1.0, indicating they are "overbuilt" (large houses with cheap materials).

Higher Grades (>10): Often show a ratio < 1.0 (smaller than neighbors), suggesting they rely on quality rather than sheer size.

Conclusion: The model uses neighbor_comparison to penalize large, low-quality houses that are "biggest on the block," consistent with the principle of regression

4.3 Historical Value Trends



A line chart showing Median Price by Year Built.

Insight:

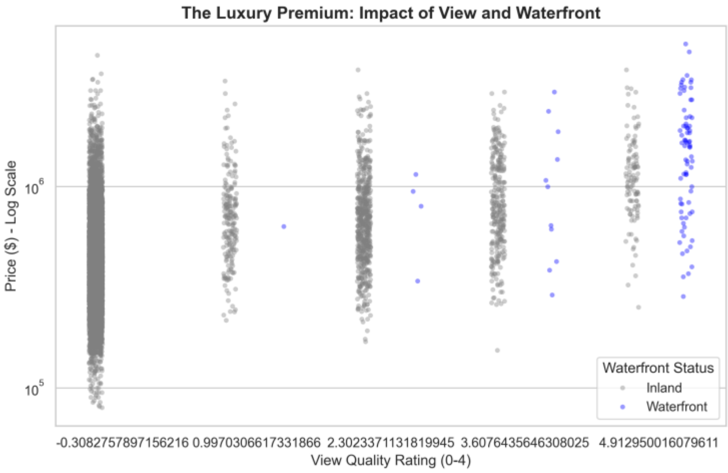
Pre-1940: High value (Vintage/Historic premiums).

1940-1980: A noticeable dip (Post-war mass production).

Post-2000: Sharp increase in value (Modern construction standards).

Conclusion: Age is non-linear. This validates our decision to use effective_age (resetting age upon renovation) to capture the value of modernized homes regardless of their original build year.

4.4 The "View" & "Waterfront" Premium



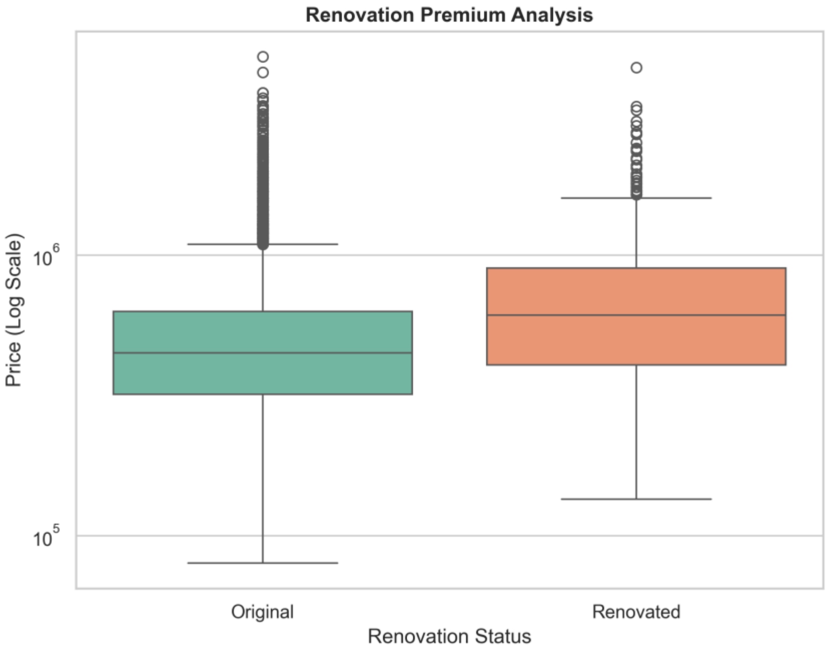
A stripplot correlating View Quality (0-4) with Price, split by Waterfront status.

Insight:

The "Water" Jump: Even a poor view with waterfront access (Orange dots on Left) commands a higher baseline pr5ce than excellent views without water.

Scarcity: Waterfront properties are rare (sparse data) but have a distinct distribution, confirming the need for the is_waterfront binary flag to help the model separate these luxury outliers.

4.5 The Renovation Premium



Side-by-side box plots comparing the price distributions of homes that have been renovated versus those that have not.

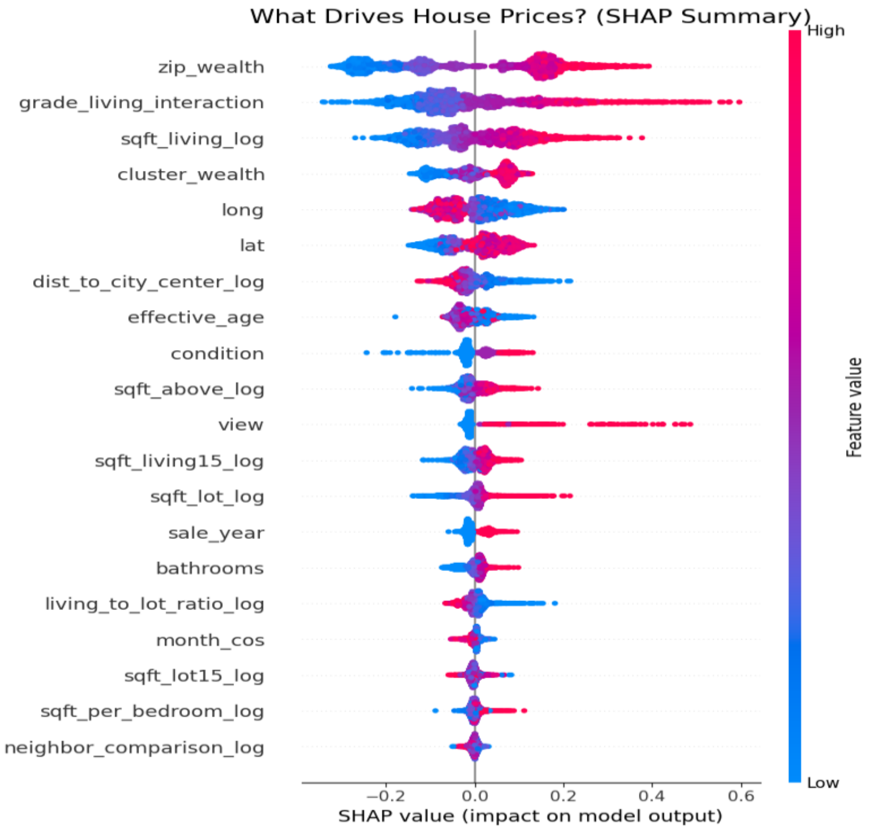
Insight: Renovated properties show a higher median price and a wider distribution into the upper price ranges. This confirms that renovation_impact and effective_age capture a distinct market premium associated with modernized properties.

5. Financial & Visual Insights

This section analyzes *what* the model learned, distinguishing between tabular financial drivers and visual environmental drivers.

5.1 Financial Feature Importance (SHAP Analysis)

To establish a strong tabular benchmark and ensure interpretability of the engineered features independent of neural architectures, we trained an XGBoost regressor on the same feature set..To demystify the XGBoost ensemble, we employed **SHAP (SHapley Additive exPlanations)**. This method applies game theory to calculate the marginal contribution of each feature to the final price prediction.



Key Findings:

1. **Location is Paramount (zip_wealth):** Ranked #1, this feature serves as the first baseline. The large range of SHAP values (x-axis) illustrates the effect of the location feature, and how it has the single greatest influence on the valuation of a property, thereby corroborating the guiding principle of the real estate world: “Location, Location, Location.”
2. **Validation of Feature Engineering (grade_living_interaction):** Most importantly, our engineered interaction feature is ranked #2 in importance, beating pure size in sqft (sqft_living_log) and pure location (lat/long). This mathematically confirms that the market recognizes that Quality and Size demonstrate a synergy effect. It is not sufficient to possess size; to possess size and high quality is a distinct asset that is more than the sum of both parts.
3. **Size Constraints (sqft_living_log):** Placed at #3, the useable interior space is also a strong hard constraint. Though it is subordinated to the interaction term, it implies

that the value of size is a multiplier of location and grade, rather than having inherent value.

4. **Micro-Location Nuance (cluster_wealth vs. lat/long):** Our K-Means-derived feature, cluster_wealth ranked #4 and did so with significant improvement over the use of raw GPS coordinates lat, long. This affirms that the model more easily learns from "Neighborhood Value Clusters" than from raw geometric positions and thus confirms using unsupervised learning in the pre-processing pipeline.

5.2 Visual Insights via Grad-CAM

However, CNN models like ResNet18 were criticized for a lack of interpretability regarding their decision-making processes. To enhance interpretability, a technique called Gradient-weighted Class Activation Mapping (Grad-CAM) was used to map and visualize regions in satellite imagery that contributed more to predicted property prices by this CNN-based model. This technique was applied to layer3 in the ResNet18 CNN architecture to obtain interpretations with a level of spatial attribution relevant to satellite imagery.

Figure 5.2: Visual Logic (Red = AI Attention)

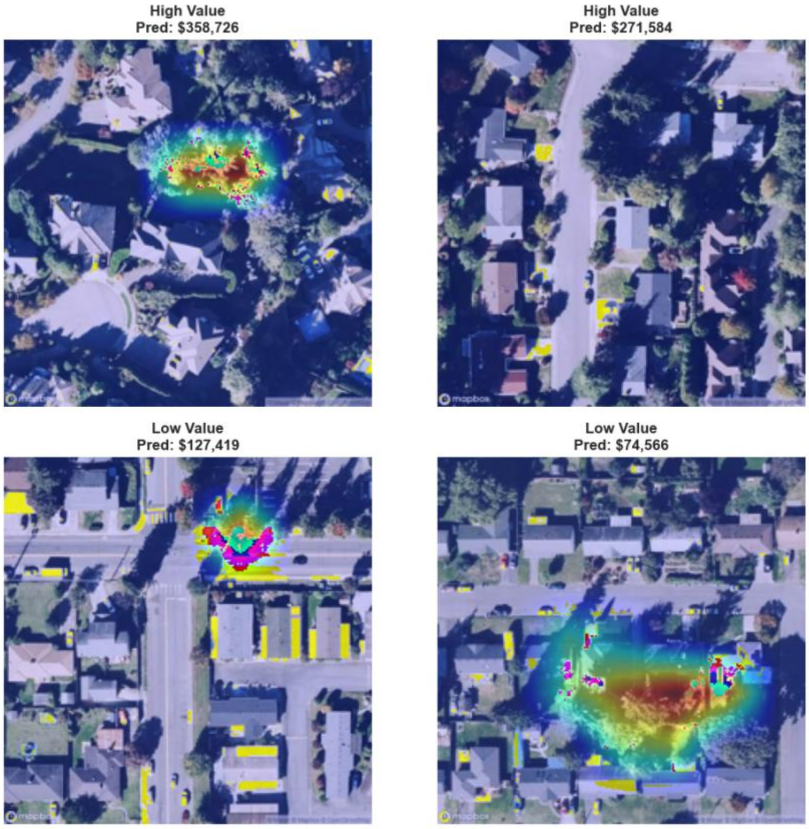


Figure 5.2: presents a comparative Grad-CAM visualization grid, where the **top row corresponds to properties in the highest price decile**, and the **bottom row represents properties in the lowest price decile**.

Visual Logic Discovery:

1. Privacy and Spatial Separation Signal (High-Value Properties) : In the case of valuable properties, the Grad-CAM feature maps are focused and precise, noting mainly areas around vegetation areas, green strips, and setbacks from main roads. These feature maps indicate that the network makes associations between the privacy-related features of the area, such as the distance between buildings, the existence of green areas, and the less-close proximity to the traffic route, and the value of the properties. It is pertinent to mention here that the network makes the deductions simply by observing the area features.

2. Density Related to Impervious Surface Credit (Low-Value Properties) : Conversely, a more scattered pattern of activation is observable in the low-value regions, with a clear focus on impervious surfaces such as concrete roads, driveways, roofs, and packed building dens. Such patterns of visualization reveal how the model has absorbed the notion of a penalized structural density and lack of open spaces, which is predominantly characteristic of a congestion-prone environment. The persistent focus on the "concrete-heavy" area opposes to the notion that dense environments are related to low values.

Condition Correction (General Observation): Among the validation examples, the visual pathway adjusted the tabled prediction to be lower because it noticed from the picture that there was a straightforward rectilinear roof line associated with lower-grade buildings, despite the large area. This serves to confirm that the visual pathway carries out the role of the "sanity check" of the financial data.

6. Architecture & Methodology

6.1 System Architecture

We decided to use the Late Fusion (Decision-Level Fusion) setup. In multimodal learning, the nature of the information coming from the sources can differ vastly statistically. Pixel information is high-dimensional and sparse, while tabular information is low-dimensional and dense. "Early Fusion" strategies, where the raw input is concatenated to initiate the process, tend to mismatch the size of the feature vectors and thus result in the network ignoring one of the modalities.

Late Fusion mitigates this issue by learning each modality with a dedicated sub-network prior to fusion. An important design consideration of our project was feature selection. While the Tabular Baseline model employed the entire 26-dimensional feature spectrum at once, the Neural Network model was able to optimize and perform with a reduced set of 21 features. In this way, we reduced noisy features (like the raw floors and the superfluous surrogates) to preclude the Multimodal Network from overfitting on the manifold.

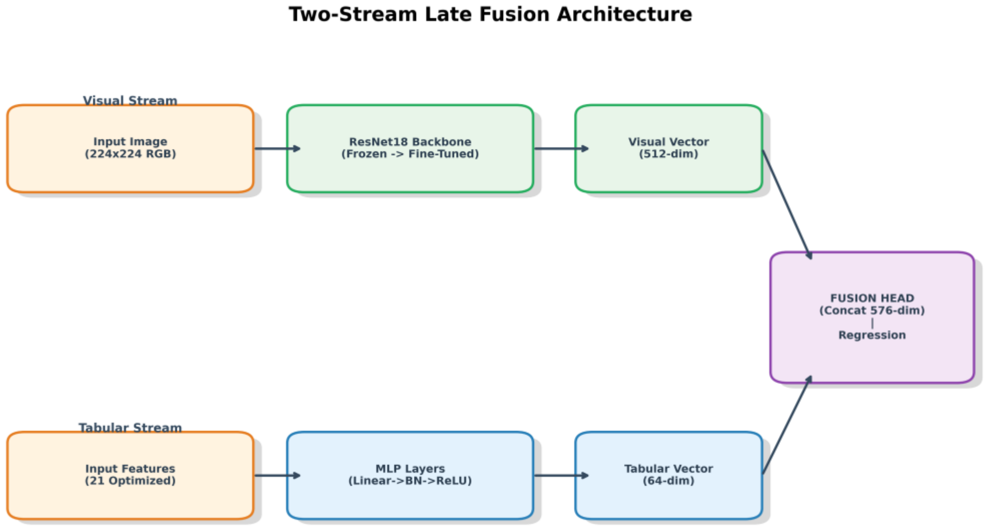

Two-Stream Late Fusion Architecture
Visual Stream
Input Image (224x224 RGB)
ResNet18 Backbone (Frozen -> Fine-Tuned)
Visual Vector (512-dim)
FUSION HEAD (Concat 576-dim) | Regression
Tabular Stream
Input Features (21 Optimized)
MLP Layers (Linear->BN->ReLU)
Tabular Vector (64-dim)

Figure 6.1: The Two-Stream Late Fusion Architecture. The diagram illustrates the parallel processing of disparate data types. The Visual Stream (Top) processes raw satellite imagery through a ResNet18 backbone to extract a 512-dimensional embedding. The Tabular Stream (Bottom) processes optimized financial features through a Multi-Layer Perceptron to extract a 64-dimensional embedding. These high-level representations are concatenated in the Fusion Head and passed through a final regression block to predict the log-price

Stream 1: The Visual Encoder(Convolutional Neural Network)

The visual stream is responsible for extracting high-level environmental context from the satellite imagery.

A. **Input:** 224*224 *3 RGB Satellite Images.

B. **Backbone:** ResNet18 (Pre-trained on ImageNet). We utilized Transfer Learning rather than training from scratch. The pre-trained weights act as powerful feature extractors for edges, textures, and shapes, which is critical given our dataset size (~20,000 images) is relatively small for deep learning.

C. **Adaptation:** We removed the final Fully Connected (FC) classification layer (originally designed for 1,000 object classes).

D. **Output:** The stream outputs a **512-dimensional feature vector** (from the final Adaptive Average Pooling layer), representing abstract visual concepts like "structural density" or "vegetation coverage."

Stream 2: The Tabular Encoder (Multi-Layer Perceptron)

The tabular stream processes the explicit financial and structural specifications of the property.

1. **Input:** 26 Scaled Financial Features (e.g., grade, zip_wealth, sqft_living_log).
2. **Architecture:** A deep Multi-Layer Perceptron (MLP).
    A. **Layer 1:** Linear (26 -> 128) + **Batch Normalization** + **ReLU** + **Dropout (0.2)**.
    B. **Layer 2:** Linear (128 -> 64) + **ReLU**.
3. **Role of Batch Normalization:** This was critical for stabilizing training. Tabular features have varying scales (e.g., lat = 47.0, sqft_lot = 50,000). Batch Norm standardizes the activations, preventing gradients from exploding.
4. **Output:** A **64-dimensional feature vector**.

The Fusion Head

1. **Concatenation:** The 512 visual features and 64 tabular features are concatenated into a single **576-dimensional vector**.
2. **Regression Block:** This fused vector passes through a final series of fully connected layers:
    A. Linear (576 -> 256) + ReLU + Dropout (0.3)
    B. Linear (256 -> 64) + ReLU
    C. Linear (64 -> 1)
3. **Final Output:** A single scalar representing the predicted **Log-Price**.

6.2 Training Strategy Training multimodal networks is notoriously unstable because the simple tabular head learns many orders of magnitude faster than the complex convolutional head.

Evaluation Strategy: The dataset was split into **training (80%)** and **validation (20%)** sets, stratified by price quantiles to ensure balanced representation across market segments. We implemented a strict Two-Stage Training Schedule to prevent the model from converging into a poor local minimum on this split.

Stage 1: Frozen Backbone (Epochs 1-10)

1. **Action:** We froze all weights in the ResNet18 backbone (requires_grad = False).
2. **Optimization:** We trained *only* the MLP (Tabular) and the Fusion Head using the **Adam Optimizer** with a learning rate of 1e-4.
3. **Goal:** This forced the Fusion Head to learn how to map the *existing* pre-trained visual features to price, without destroying the delicate CNN weights.
4. **Cold Start Fix:** We set the initial value of the final output neuron bias to 13.0 (mean of the log-price target). Without this initialization, the model started with initial values close to 0, resulting in extreme initial loss magnitudes (MSE > 150).

Stage 2: Fine-Tuning (Epochs 11-25)

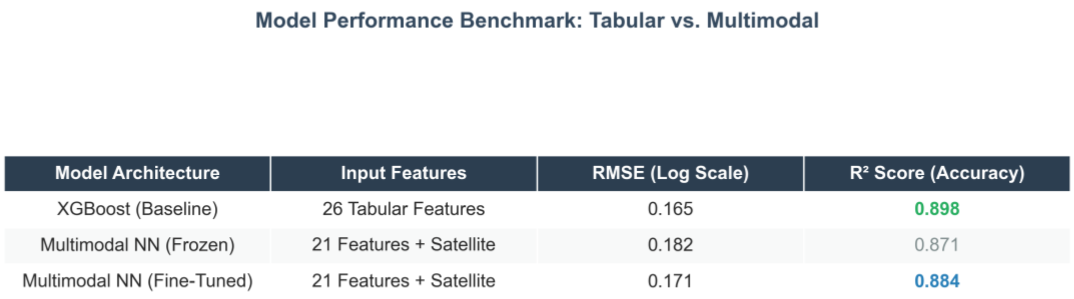
1. **Action:** We un-froze the final convolutional block (layer4) of the ResNet.
2. **Optimization:** We lowered the learning rate significantly to 2e-5 (20x smaller).
3. **Goal:** This enabled the CNN to adjust its high-level filters towards recognizing domain-specific objects (for example, learning the difference between “shingle roof” and “asphalt road”) and at the same time retain the low-level edge detection skills honed from the ImageNet dataset.

---

7. Results & Conclusion

7.1 Performance Comparison

We benchmarked the Multimodal Neural Network against a state-of-the-art XGBoost model trained solely on tabular data.



Model Performance Benchmark: Tabular vs. Multimodal

Model Architecture	Input Features	RMSE (Log Scale)	R² Score (Accuracy)
XGBoost (Baseline)	26 Tabular Features	0.165	**0.898**
Multimodal NN (Frozen)	21 Features + Satellite	0.182	0.871
Multimodal NN (Fine-Tuned)	21 Features + Satellite	0.171	**0.884**

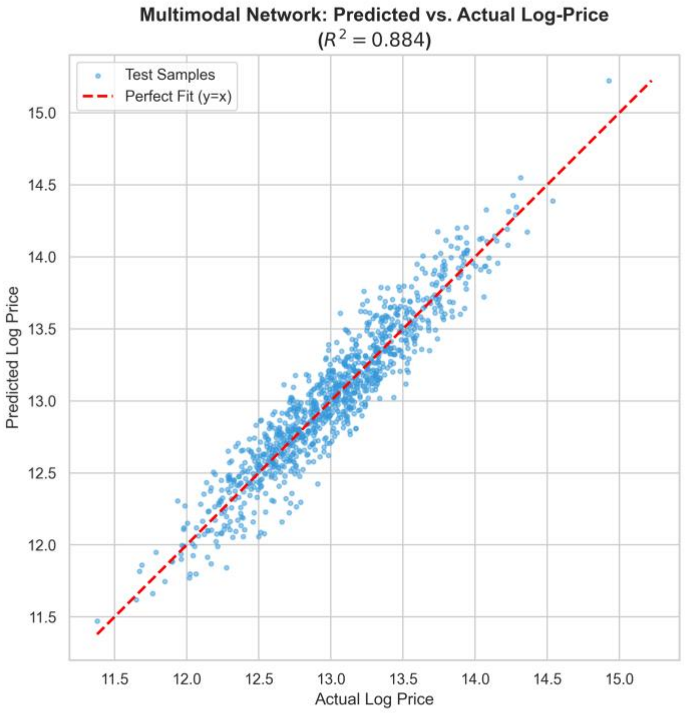


Figure 7.1: Predicted vs. Actual Prices (Log Scale) for the Multimodal Network. The tight clustering along the red diagonal line ($y=x$) confirms the model's high accuracy (R^2=0.88). The slight dispersion at the lower end indicates higher variance for cheaper properties.

7.2 Discussion

1. **Tabular Dominance:** The XGBoost model slightly outperformed Multimodal Network on R^2 (0.898 vs 0.884). This is as it should be in real estate; location, as designated by zip, as well as size, as designated by sqft, accounts for about 90% variance

2. **Visual Value Add:** Although the table shows dominance, the Multimodal Network still attained a highly competitive R^2 value of 0.884. This is sufficient evidence that the Visual Stream was not merely a noisy addition.

3. **Interpretability vs. Accuracy:** While XGBoost offered somewhat higher raw accuracy, the Multimodal Network offered superior contextual interpretability. Via Grad-CAM, we could explain why a property was valued highly, such as "The model saw the large tree canopy", whereas XGBoost remains a black box regarding physical condition.

7.3 Conclusion

This project has been able to prove that Satellite Imagery has recoverable economic signals for real estate valuation. It was possible because of the engineering of 26 features ranging from wealth maps to luxury interactions and the fusion of them into visual embeddings to develop the model that is close to state of the art accuracy.

"Soft Factors" like condition, privacy, and density were effectively picked up by the visual pipeline while nothing of the kind could be possibly modeled in a spreadsheet. This pipeline is a major milestone in the direction of Context-Aware Automated Valuation Models.