ANN IMAGE SEARCH WITH FINE TUNED EMBEDDING SPACE USING ACTIVE LEARNING METHODS

Naomi Derel & Gili Cohen & Renana Shachak

naomi.derel, gili.cohen, renanas @campus.technion.ac.il

31.10.2024

ABSTRACT

This project addresses the challenge of efficiently retrieving visually similar house images for real-estate search tasks. We propose a novel pipeline that creates a vector database of image embeddings fine-tuned through Siamese neural networks, and applies Approximate Nearest Neighbor (ANN) search techniques to retrieve visually similar results from the database. This is achieved by creating a unique dataset of similarity-labeled image pairs, using active learning to iteratively refine the labeled data in this resource-intensive domain. This approach shows a novel combination of methodologies for improving user experience by delivering fast and accurate image-based search results, in real estate and other fields where finding visually similar images is crucial. Results indicate promising retention of key search metrics alongside enhanced efficiency, highlighting potential for further advancements and applications. [1]

1 Introduction

Efficient retrieval of relevant items from large datasets is a fundamental challenge in information retrieval (IR), especially as modern datasets grow in complexity. This has driven interest in approximate nearest neighbor (ANN) algorithms, which offer a balance between speed and accuracy by trading off some precision for faster query times [1].

In this work, we focus on image similarity search within the real estate domain, where the user's information need includes visual design properties. Traditional real estate search engines easily filter results based on structured data, such as price or location, but it is difficult to accurately account for visual preferences. Our goal is to develop a retrieval pipeline that quickly finds visually similar house images based on a query image, thereby improving the ability of such search engines to answer the information need.

We begin by leveraging a pre-trained vision model, CLIP [2], to embed house images into a high-dimensional vector space. To improve both the efficiency and precision of the ANN search, we fine-tune these embeddings using a Siamese neural network [3], a supervised training method designed to bring embeddings closer or further apart according to their similarity. This network helps reduce the dimensionality of the embedding space while optimizing it to better align with the user's visual preferences.

To support the training of our model, we started with an existing dataset of house images labeled by exterior style: modern, rustic, and farmhouse. We expanded this dataset by constructing pairs of images, each accompanied by a similarity label indicating whether the images share the same style. This expanded dataset was further enhanced using active learning techniques, allowing us to iteratively improve the model and focus on the most uncertain or informative image pairs - as previously done in various vision tasks [4, 5].

The final result is an automated pipeline capable of retrieving top visually similar house images based on a query image, with ANN. The performance of our system is evaluated using task-specific metrics for approximation quality, run time improvement, and retrieval quality, in addition to concrete examples and possible future improvements.

Our Contribution is twofold: first, we present an expanded dataset of house image pairs annotated with similarity labels, facilitating more precise model training for future visual search tasks. Second, we propose a novel combination image retrieval pipeline that utilizes a fine-tuned embedding space. It is distinct from previous state-of-the-art ANN

[1] Code is available at `https://github.com/NaomiDerel/Image_Search`

methods due to the unique embedding space [6], while also focusing on the efficiency of retrieval by implementing ANN, distinct from previous image retrieval efforts using Siamese networks [7].

2 Data

2.1 Original Dataset & Preprocessing

Our project utilizes the "House Type/Style Detection" dataset from Kaggle [8], comprising 867 house images categorized into three architectural styles: farmhouse (428 images), modern (289 images), and rustic (150 images). For our similarity search task, we performed standard preprocessing steps, including resizing all images to a 128x128 pixel resolution.

2.2 Paired Image Similarity Dataset

Creating a similarity model requires pairs of images with similarity labels. Since labeling all possible pairs (867^2) would be infeasible, we developed a strategic sampling approach. We first sampled 150 images from each category to ensure balanced representation and generated all possible pairs, creating a shuffled dataset of 450^2 samples. To simplify the labeling process, we automatically assigned a dissimilarity label (0) to any pair of images from different categories - a reasonable assumption given our definition of similarity being based of style. This reduced the manual labeling task to $3 * 150^2$ potential pairs, from which we sampled subsets for human annotation by our team members.

The objective of the task was defined as follows: The users information need is finding houses with similar aesthetic characteristics, utilities, and environments to the given query image. To ensure our similarity labels reflected this goal consistently across our three human annotators, we established clear labeling guidelines.

Manual labeling rules centered around a standardized questionnaire (Appendix A) containing 9 questions about specific features: house size, color scheme, building materials, distinctive stylistic or functional elements, surrounding environment, and overall aesthetic impression. Annotators were explicitly instructed to ignore non-architectural features such as image quality, lighting, etc., as these do not relate to architectural similarity.

The similarity scoring used a scale of 0 to 3, where 0 was reserved for cross-category pairs. Human annotators assigned scores from 1 to 3 based on the proportion of positive question responses and their expert judgment. Given the nuanced nature of architectural similarity assessment, this process could not be effectively automated - which made our process highly costly, and presented a real world scenario for active learning where we need to maximize the utility of limited labeling resources.

Active Learning was implemented through iterative rounds of tagging and model training to optimize our labeling resources. After an initial data pool, each subsequent round selected both untagged samples 0 samples, using a model-specific least confidence calculation, with data split into train (80%) and evaluation (20%) sets each round.

The final similarity dataset, our first contribution, was developed over 4 rounds containing 2368 fully labeled samples (scores 0-3). The dataset includes 1654 training samples and 414 evaluation samples, allocated based on the performance of the model after the initial round. Furthermore, 300 test samples were independently selected and kept separate from all training procedures.

Usage in our embedding model was reduced to binary labels, treating only score 3 as positive similarity. This decision prioritized quality over quantity in search results, aligning with our goal of providing users with highly relevant matches.

3 Models

Our pipeline includes two main models. First, the embedding model, to create a database of embedded image vectors in a vector space, where Euclidean distance between vectors represents their estimated similarity. Second, the retrieval model, to extract the closest k images based on a query image with an ANN architecture.

3.1 Embedding Model

Siamese networks are specialized neural architectures used to learn similarity metrics between pairs of inputs. These networks consist of two identical neural subnets that share weights and parameters, and are connected by a distance function [3]. This architecture is particularly well-suited for our task, as it enables the network to map images into a feature space where similar images are positioned closer together, while dissimilar images are placed farther apart. This approach is consistent with other applications in the literature, where Siamese networks have been used to evaluate the similarity of complex inputs, such as songs [9].

3.1.1 Siamese Network

Our Siamese network implementation utilizes CLIP (Contrastive Language-Image Pre-training) [2] as the backbone for initial feature extraction. CLIP, which has been pre-trained on 400 million image-text pairs, provides robust visual representations that capture both low-level image features and high-level semantic information. We specifically use the ViT-B/32 variant of CLIP, which generates 512-dimensional feature vectors.

Network Architecture: The architecture takes two image inputs that are transformed into 512-dimensional CLIP embeddings. Each embedding is then processed separately by a fully connected two-layer network with ReLU activation functions, which reduces the embeddings to 128 dimensions. This design allows for fine-tuning of the CLIP-generated features, which already include learned representations of the images and convolutional layers.

Contrastive Loss Function: To learn the similarity between the two 128-dimensional embeddings produced by the network, we use the contrastive loss function [10]. Unlike traditional loss functions, contrastive loss takes two input vectors, x_1 and x_2, along with a label y indicating their similarity (1 for similar pairs, 0 for dissimilar pairs). The distance function $D(x_1, x_2)$ is defined as the Euclidean distance, and we use a margin parameter $m > 0$, set to 1 [11]. The loss is computed as:

$$L(x_1, x_2, y) = y \cdot D(x_1, x_2)^2 + (1 - y) \cdot \max(0, m - D(x_1, x_2))^2 \tag{1}$$

This formulation minimizes the distance between similar pairs ($y = 1$) while penalizing dissimilar pairs ($y = 0$) if their distance falls below the margin, encouraging them to be pushed farther apart.

Training Procedure: The network is trained using the Adam optimizer [12] with a learning rate of 0.01, combined with data augmentation techniques such as random resized cropping, horizontal flipping, and color jittering (each with a 30% probability of activation) in earlier rounds. Training continues in rounds, each comprising approximately 100 epochs with additional data samples added per round. The model continues to train from the saved weights of the previous checkpoint, allowing progressive improvement in its predictive capabilities.

3.1.2 Uncertainty Active Learning Algorithm

After each training round, the model selects new samples up to a predefined budget, using an uncertainty-based sampling algorithm, a popular approach to active learning [13]. In this case, the model defines the confidence score as the absolute difference between the euclidean distance of the image embeddings, and the defined margin. Specifically, pairs with distances close to the margin (small confidence scores) represent high uncertainty cases where the model struggles to definitively classify them as similar or dissimilar. This approach is inspired by previous work, applying an absolute value on a confidence margin from a Perceptron model [14]. It aligns with the intuition that borderline cases near the margin boundary require more attention. By selecting samples with the lowest confidence scores, the model focuses its learning on the most ambiguous cases.

Due to resource constraints, the algorithm searches for high-uncertainty samples within a smaller subset of the entire data pool. However, due to the large size of the data pool compared to the tagged data, the selected samples still remain within 0.1 of the margin. To maintain balance in the training data and manage the number of human annotations, the algorithm chooses unlabeled data points with high uncertainty up to the budget, along with three times as many high-uncertainty automatically labeled dissimilar points.

3.2 Retrieval Model

Approximate Nearest Neighbors (ANN) is a common approach to improve the efficiency of exact nearest neighbor search, particularly valuable for high-dimensional vector spaces like image embeddings [1]. By accepting a small compromise in accuracy, ANN methods can dramatically improve search speeds compared to exhaustive search. It has often been used in image search functions and engines due to these desired qualities, where few matches are required and speed is of importance [6].

We utilize the FAISS (Facebook AI Similarity Search) library [15], which provides efficient implementations of various ANN algorithms. For our image retrieval system, we explored a few indexing approaches:

- FlatL2: performs exact Euclidean distance computation, as a baseline to other indexes.
- LSH (Locality-Sensitive Hashing): projects vectors onto random hyperplanes to create binary hash codes.
- HNSW (Hierarchical Navigable Small World graphs): organizes high-dimensional vectors in a graph structure, where each node represents a vector, and the edges represent the approximate distance between the vectors.

These methods were selected based on the properties of our fine-tuned embedding space, where Euclidean distance was designed to effectively captures meaningful relationships between images. Using Euclidean distance across all methods ensures consistency in similarity measurement.

4 Experiments

Our experimental evaluation focuses on three key aspects of the pipeline: the performance of the embedding model by itself through active learning iterations, the effectiveness of various approximate nearest neighbor indexing methods, and qualitative analysis of the recommendations. We aim to validate both the technical performance metrics and practical utility of our system.

4.1 Embedding Model Performance

Training & Evaluation: During rounds of active learning, we evaluated the model using an F1 score metric on the training and validation sets randomly sampled from the labeled data pool created in the AL process. The F1 metric, defined as $F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$, was chosen specifically to address the significant class imbalance in our dataset, where dissimilar pairs naturally outnumber similar ones.

The results across training rounds as seen in 1 show the learning rounds improved model's understanding, and confusion matrices (Appendix B Figure 5) shows slight improvement as well. While the absolute F1 scores may appear lower than typical classification tasks, this is expected due to our architectural choice in the first part of a pipeline. Rather than implementing a dedicated classification layer, we focused on learning meaningful embeddings for similarity search, using distance from the margin as our decision boundary. These embeddings are the focus in the next steps of the pipeline, over a traditional classification task.

Independent Test Set Performance: To validate the robustness of our embeddings, we created a completely independent test set, randomly selected without any model influence. This "blind" test set provides an unbiased assessment of the model's generalization capabilities. The final model performed similarly on the "blind" test set as it did on the validation set, as can be seen in 1. This suggests the samples of labels with the active learning method, instead of a random sample, did not significantly harm the generalization ability of the model.

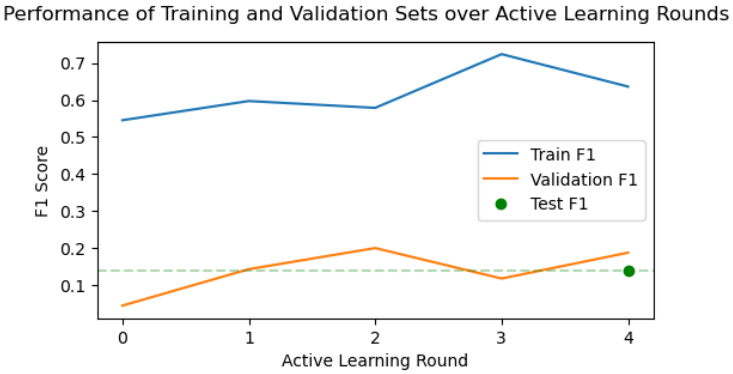


Figure 1: F1 progression across active learning rounds

4.2 Retrieval Model Performance

4.2.1 Approximation Quality Analysis

To understand the trade-off between the exact and approximate nearest-neighbor search, we measured recall@k for various k values. This analysis reveals how many true nearest neighbors are preserved by each indexing method, crucial to understanding the practical impact of approximation. As can be seen in Figure 2, the HNSW index performs significantly better for smaller k values with both models, and since the required k for this task is relatively small, we proceed with our analysis using HNSW exclusively.

4.2.2 Retrieval Run Time

The run-times for index creation using 400 indexed images, and for query searches with 50 query images (Appendix B Table 1), indicate that Siamese vectors are built and searched approximately twice as fast as CLIP vectors - with HNSW index over Siamese vectors running in 1 millisecond. This observation aligns with our expectations regarding the reduction in vector size in our model, and shows one consistent benefit of our approach to IR tasks where speed is crucial.

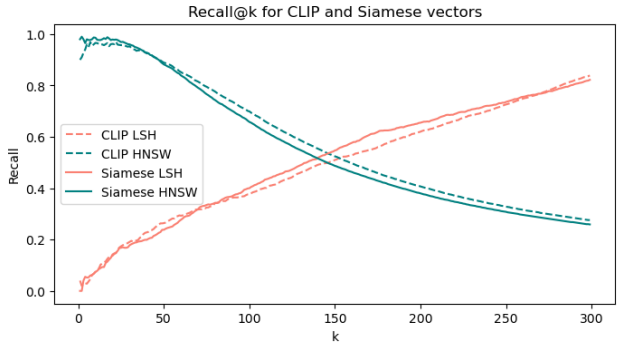


Figure 2: Recall@k curves for LSH and HNSW indexes, using CLIP and Siamese embeddings

4.2.3 Retrieval Quality

To analyze image retrieval quality, we consider widely used evaluation metrics: MAP and NDCG. These metrics prioritize precision in top-ranked results intended for retrieving only the most relevant images, and reward rankings where more relevant items appear earlier. Both of these qualities make them suitable for the goal of our pipeline.

In our evaluation, we use $metric@k$ scores, which consider only the top k retrieved results. For this analysis, we set $k = 5$, allowing us to annotate all relevant similarity labels within resource constraints and achieve more accurate results compared to partial-label methods we considered.

MAP@k (Mean Average Precision) calculates the average precision ammended with relevance scores, over the top k ranks - where Q is the number of queries, R_q the relevant items for query q, Precision@j the precision at rank j, and rel(j) is the relevance at rank j:

$$\text{MAP@}k = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{R_q}\sum_{j=1}^{k}\text{Precision@}j \cdot \frac{\text{rel}(j)}{\text{Max Relevance}} \tag{2}$$

NDCG@k (Normalized Discounted Cumulative Gain) measures both relevance and rank order, normalized by the ideal score:

$$\text{NDCG@}k = \frac{1}{\text{IDCG@}k}\sum_{i=1}^{k}\frac{2^{\text{rel}(i)}-1}{\log_2(i+1)} \tag{3}$$

Relevance Scores: We measured each metric with two variations on the relevance score. The first is binary score, which are 0 for images tagged as 0-2, and 1 for images tagged 3. This method matches the training of the Siamese network, and complies with the task of only highly similar matches being relevant. The second is multi-relevance scores, using the full label range 0-3 as they are. This method allows a more accurate mistake-penalization process, although it is more difficult for the Siamese model, as it was not trained on this range.

Results: The results of MAP (Table 3) and NDCG (Table 4) show the Siamese model generally performs closely to CLIP and better, both with binary and multiple relevance scores, excluding the multiple NDCG metric.

Model	MAP@5 Binary	MAP@5 Multiple
CLIP	0.366	0.632
Siamese	0.39	0.804

Figure 3: MAP scores for HNSW index across models

Model	NDCG@5 Binary	NDCG@5 Multiple
CLIP	0.428	0.72
Siamese	0.441	0.705

Figure 4: NDCG scores for HNSW index across models

4.3 Qualitative Analysis of Recommendations

4.3.1 Visual Analysis

To understand the value of the recommendations beyond metric results, we present a few example cases of the retrieved results from our model to highlight which patterns and features the model successfully captures, as well as instances where results deviate from expectations. These examples can be found in Appendix C.

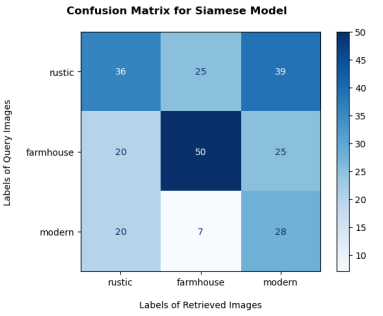
In cases where our model outperforms CLIP (e.g. Figure 8), it appears to capture features like color (e.g., white), roof shape, and house size, yielding images that closely match the query. In contrast, CLIP's results often fall into unrelated categories and do not resemble the query as closely. Potential sources of error might be variations in image lighting or an overemphasis on greenery in the images. Additionally, our model seems better at retrieving cropped images that show part of a house, a capability where CLIP is less effective.

In cases where CLIP outperforms our model (e.g. Figure 9), our model often retrieves images from entirely different categories. This may be due to the model's focus on features that can span across categories (e.g., size and style elements like stairs), which we emphasized in manual tagging. CLIP, without this specific fine-tuning, remains closer to images that match the query on a surface level and within the same category.

In cases where both models perform poorly (e.g. Figure 10), some retrieved images may show superficial similarities to the query, but many are from unrelated categories and lack visual resemblance. This suggests that unimportant features like photo angle or lighting might sometimes overshadow the desired features. There may also be insufficient training on images similar to the query, or inconsistent training of the query image itself.

4.3.2 Performance by Architectural Style

To help identify potential biases or limitations in our approach, we analyze whether certain house types are more challenging for our system to classify. The results of the Siamese model (Figure 7) show the best performance on the farmhouse style, with most accurate retrievals as well as fewer mistakes for other styles. This is likely due to the unique characteristics of a large mount of houses labeled "farmhouse", which are white, traditional, of similar size and very unique to other houses in the data. Surprisingly, the rustic and modern styles are confused the most, despite being very different visually. This might be due to many houses in both categories being much more extravagant, with similar properties like very large size, bodies of water or a large amount of land. Because we accounted for these features in our ratings, the model might be confused by this - which is supported by the fact the CLIP model makes fewer of these mistakes (Figure 6).



5 Discussion

Throughout our work, we encountered several limitations. A primary challenge was finding a dataset of house pairs labeled by similarity level, leading us to manually label each pair based on specific criteria. This process could be improved in both quality and efficiency by enlisting more judges who could label image pairs following the guidelines and aggregating their responses, to help mitigate bias and reduce reliance on individual taggers' perspectives.

Our dataset presented additional issues that may have impacted results, such as duplicate images and unusual cropping, which likely influenced the retrieval quality. Moreover, our initial approach to labeling houses from different categories as 0 proved problematic. Throughout our work, we observed that houses across categories can sometimes share more similarities than those we labeled 1, suggesting that a more nuanced labeling scheme could enhance model performance.

Our experiments show that active learning modestly improved our model's embeddings over CLIP's baseline, validating both the labeling and training approach. With binary labels, the model achieved higher MAP and NDCG scores than CLIP, indicating that it retrieves a slightly higher number of similar images. Surprisingly, with non-binary labels, despite initial expectations that the Siamese model would perform worse, it achieved a much higher MAP score. This suggests that even without access to graded labels the model developed an understanding of closer image relationships, although this improvement did not extend to NDCG.

While it is difficult to conclude that our model achieved high accuracy in returning similar images, it did maintain comparable performance with only a quarter of the embedding size and half the runtime - an important advantage in information retrieval. Additionally, the recall rate of the approximate nearest neighbor approach demonstrates that approximation, rather than exact search, does not negatively impact results and increases the retrieval speed by nearly threefold.

To conclude, the key results of this work introduce a dataset of similarity-labeled image pairs and a novel pipeline for a real estate similarity search engine, adaptable to any visual similarity search task. Future work could focus on improving our results by addressing dataset issues, such as duplicates and inconsistencies, and implementing a more nuanced labeling scheme, as our findings indicate that certain cross-category similarities were overlooked in binary labeling. Further progress could be achieved by customizing the retrieval due to the highly subjective nature of the task: identifying the features most relevant to users, refining similarity rules, and setting feature weights to better reflect user preferences.

References

[1] Jinhui Tang, Zechao Li, Meng Wang, and Ruizhen Zhao. Neighborhood discriminant hashing for large-scale image retrieval. *IEEE Transactions on Image Processing*, 24(9):2827–2840, 2015.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[4] Asim Smailagic, Pedro Costa, Hae Young Noh, Devesh Walawalkar, Kartik Khandelwal, Adrian Galdran, Mostafa Mirshekari, Jonathon Fagert, Susu Xu, Pei Zhang, and Aurélio Campilho. Medal: Accurate and robust deep active learning for medical image analysis. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 481–488, 2018.

[5] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh. Active learning for interactive 3d image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011: 14th International Conference, Toronto, Canada, September 18-22, 2011, Proceedings, Part III 14*, pages 603–610. Springer, 2011.

[6] Dror Aiger, Efi Kokiopoulou, and Ehud Rivlin. Random grids: Fast approximate nearest neighbors and range searching for image search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3471–3478, 2013.

[7] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1386–1393, 2014.

[8] Kelvin Gothman. House type/style detection, 2022.

[9] Marko Stamenovic. Towards cover song detection with siamese convolutional neural networks. *arXiv preprint arXiv:2005.10294*, 2020.

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[11] Sean Benhur. A friendly introduction to siamese networks, 2020.

[12] P Kingma Diederik. Adam: A method for stochastic optimization. 2014.

[13] Burr Settles. Active learning literature survey. 2009.

[14] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 413–424. Springer, 2006.

[15] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

A Full Tagging Manual

A.1 Objective

The users information need is finding houses with similar aesthetic characteristics, utilities, and environment.

A.2 Features for tagging:

1. Are the houses of similar size? (floors / area / people capacity)
2. Are the houses both in bold or regular colors?
3. Are the houses the same color palette? (light / dark)
4. Are the houses made from the same material? (wooden / concrete / glass / brick)
5. Do the houses share the same building style "vibe"? (roof shape / floors...)
6. Do both houses have some identical characteristics? Examples:
    - For modern: pool, parking space
    - For farmhouse: porch, garage
    - For rustic: chimney
7. Do both houses have or lack a garden or open space?
8. Are the houses in the same environment? (urban / rural)
9. Do the houses both feel open or closed? (windows / spaces / doors)

A.3 Features to avoid:

1. Avoid comparison by image size, quality, or photo style.
2. Avoid comparison influenced by angle.
3. Avoid comparison by time of year and time of day in the image.
4. Avoid considering people and objects in the image.

A.4 Labeling Rules:

Consider all the features above and answer those questions with "yes", "no", or "not relevant / not sure". Among the relevant features, calculate the positive answers. The general guidance for labels should be as follows:

- More than 2/3 positive answers: label 3
- Between 1/3 and 2/3 positive answers: label 2
- Less than 1/3 positive answers: label 1
- Different style categories: automatic label 0

B Supplementary Data and Analysis

Table 1: Index Building and Search Times for CLIP and Siamese Vectors (in milliseconds)

Operation	CLIP Vectors	Siamese Vectors
Built L2 index	2.083	0.703
Built LSH index	7.435	2.593
Built HNSW index	6.017	2.326
Average Build	5.178	1.874
L2 Search	3.904	2.934
LSH Search	2.235	0.928
HNSW Search	3.086	1.031
Average Search	3.075	1.631

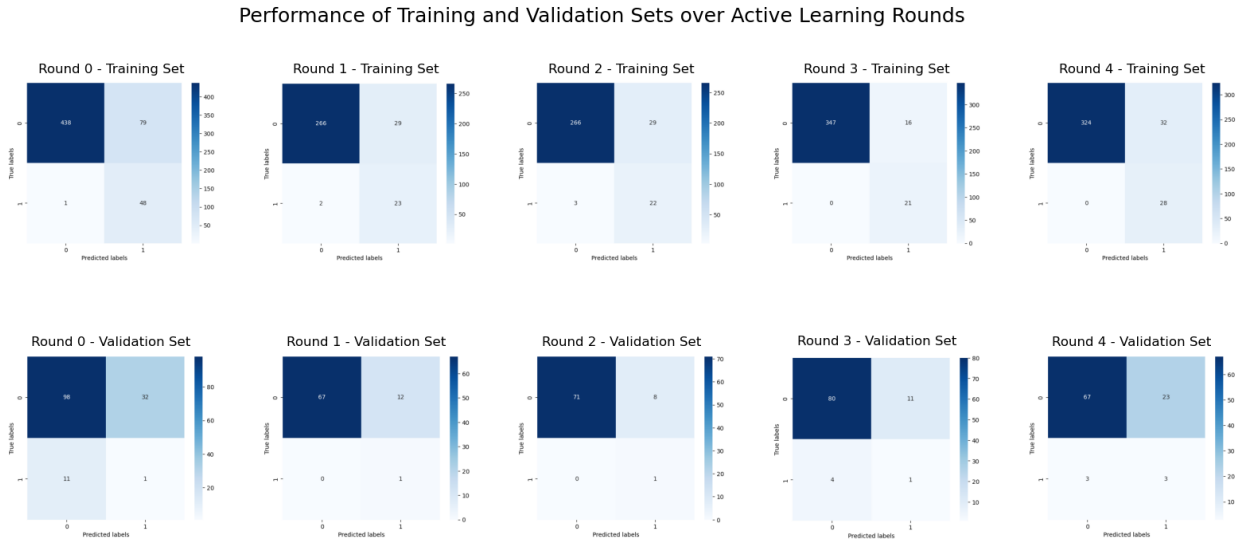


Figure 5: Confusion Matrices for Train and Evaluation Sets over Active Learning Rounds

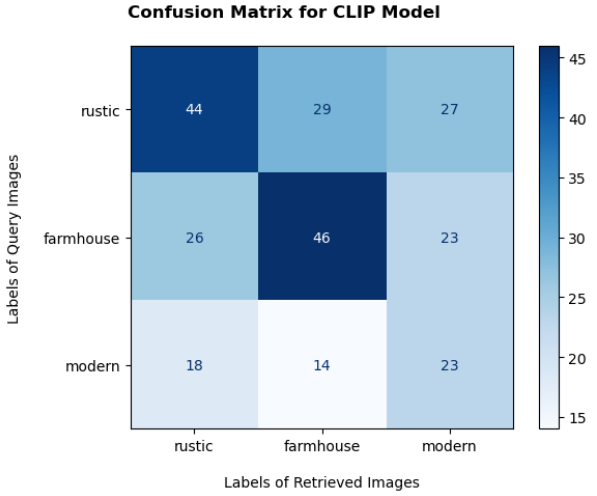


Figure 6: Confusion Matrix of House Types for CLIP Model

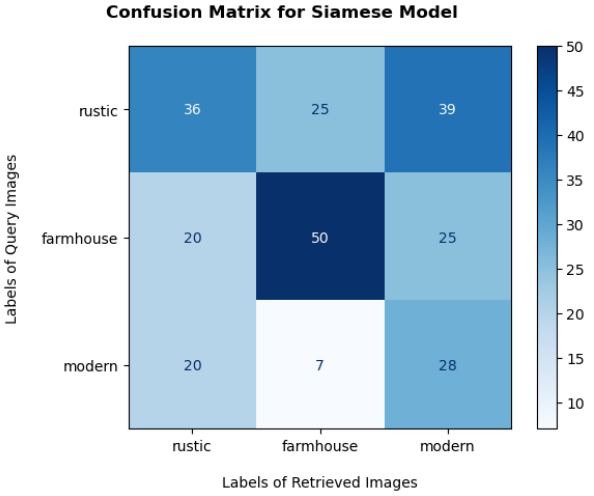


Figure 7: Confusion Matrix of House Types for Siamese Model

C Examples of Retrieval Results

Figure 8: Example of Retrieval with Siamese Model Outperforming CLIP Model

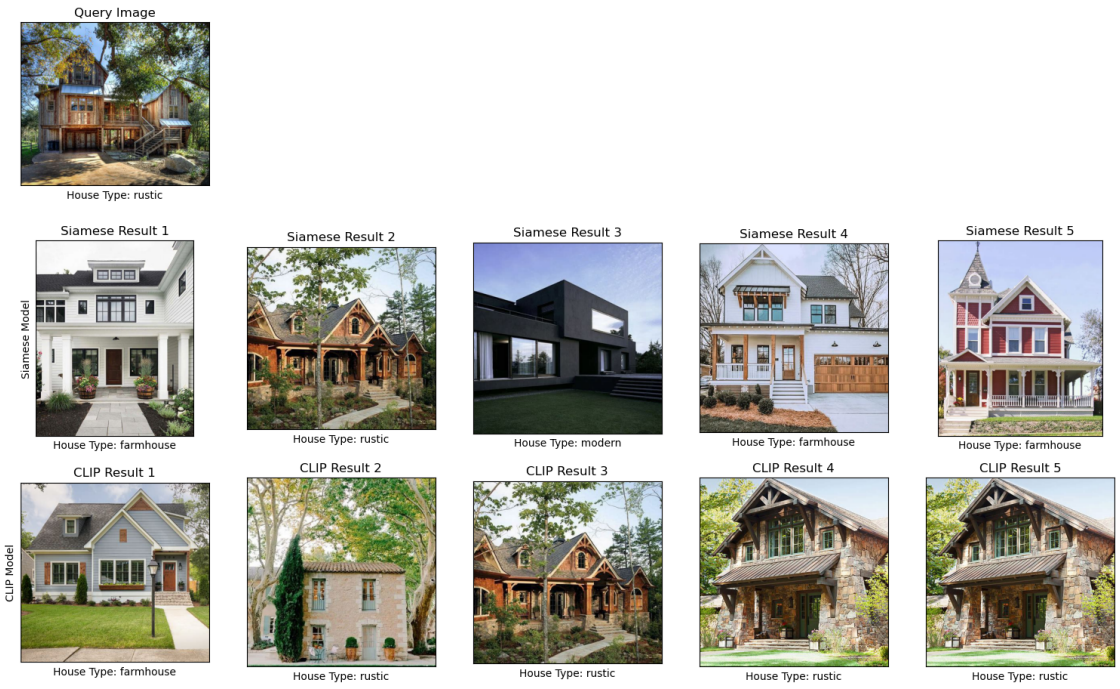


Figure 9: Example of Retrieval with CLIP Model Outperforming Siamese Model

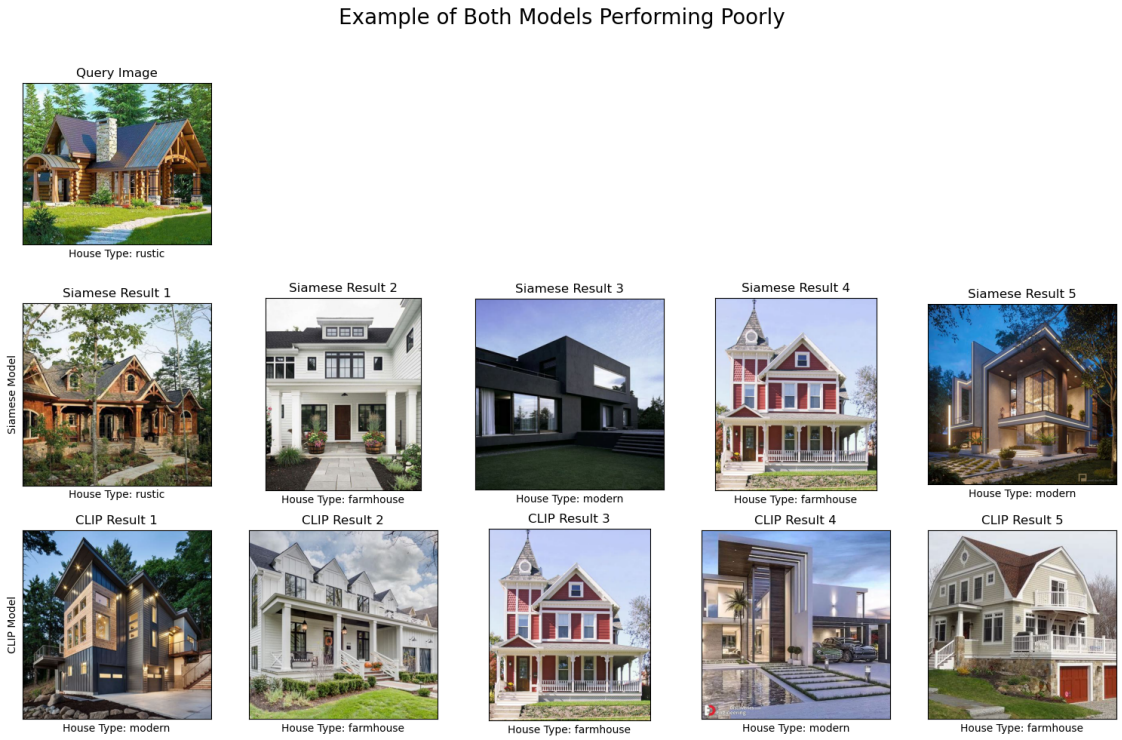


Figure 10: Example of Retrieval with Both Models Under-performing