Flexible Interactive Retrieval System for Content-Based Image Retrieval (CBIR)

Quan Hoang Ngoc
Dai-Truong Le-Trong
Professor: Thanh Duc Ngo
Course: Multimedia Information Retrieval - CS336
Institution: University of Information Technology (UIT)

January 24, 2025

Abstract

Content-Based Image Retrieval (CBIR) from textual queries is a critical task at the intersection of computer vision and natural language processing. It involves retrieving images using textual descriptions as queries. In this study, we evaluate the performance of two state-of-the-art models, CLIP and SigLIP, on the blended dataset of MSCOCO and Flickr. Building on these findings, we propose the QT-FIRS (Flexible Interactive Retrieval System). This framework enhances image retrieval by developing strategies that combine results from models or retrieval phases, incorporating relevant feedback iteration, semantic hashing, collection expansion, and leveraging Large Language Models (LLMs) alongside Referring Image Segmentation techniques. Our contributions include a comparison of CLIP and SigLIP baseline with various enhancement approaches, the design of a user-friendly image retrieval system, and improving the interpretability of CBIR results, the interactivity and flexibility of the system, as well as adapting it to large scale and efficiency.

1 Introduction

Recent advancements in deep learning have enabled substantial progress in computer vision (CV) and natural language processing (NLP), resulting in powerful applications such as Content-Based Image Retrieval (CBIR). CBIR enables efficient retrieval of images from textual queries, proving valuable in various domains including e-commerce and multimedia search.

Models such as CLIP [1] and SigLIP [2] have significantly contributed by embedding text and images in a shared vector space. While CLIP sets the foundation for multimodal understanding, SigLIP enhances embedding stability. However, both models face challenges in interpretability, interactivity, flexibility, and scalability. This report aims to evaluate these models on the blended dataset of MSCOCO [3] and Flickr [4], analyzing strengths, and weaknesses, and proposing potential enhancements via a flexible interactive framework, QT-FIRS.

Problem Definition

The primary goal of CBIR is to retrieve images relevant to user needs through textual queries from a collection of images. For example:

- "A group of people riding bicycles in a park."
- "A black cat sitting on a wooden bench."

The system outputs ranked images based on similarity scores. Challenges arise in handling ambiguous or complex queries, as well as large-scale images from the collection, necessitating iterative user refinement and semantic compression. QT-FIRS addresses these through interactive feedback and advanced query and collection techniques.

2 Collection

To evaluate QT-FIRS, we use a blended dataset, illustrated in Figure 1 including **36,014 images** comprising the MSCOCO test set and the Flickr30k dataset. This combination leverages their complementary strengths, providing a comprehensive benchmark to assess the system's ability to handle diverse and complex Content-Based Image Retrieval (CBIR) queries.

Flickr30k Dataset

The Flickr30k [4] dataset includes 31,783 images that primarily depict people and animals in various activities, events, and scenes. Each image is paired with five human-generated captions, resulting in 158,915 detailed and semantically rich descriptions. Captions are created by annotators via Amazon Mechanical Turk, where quality measures, such as grammar and spelling checks, ensure clear and accurate textual data.

Flickr30k builds upon the Hodosh dataset, which includes 8,092 images with annotations focused on visible content rather than unseen context. For example, a single image of a man in a yellow tie is described with captions such as:

- "A gray-haired man in a black suit and yellow tie working in a financial environment."
- "A businessman in a yellow tie gives a frustrated look."

This variability in descriptions provides diverse perspectives, improving semantic understanding in CBIR tasks.

MSCOCO Dataset

The MSCOCO [3] (Microsoft Common Objects in Context) dataset offers a broader range of natural scenes with its test set. Each image is annotated with five captions detailing objects, interactions, and environmental context. For example:

- "A boy holding a kite in an open field."
- "A cat sitting on a couch next to a stack of books."

Unlike Flickr30k, MSCOCO emphasizes the relationship between objects in everyday environments, providing additional diversity and complexity in both imagery and text.

Why Blend These Datasets?

Blending these datasets creates a versatile benchmark that balances Flickr30k's focus on human-centric semantic detail with MSCOCO's rich environmental diversity. This combination enables:

- Assessment of QT-FIRS's performance across human-specific and contextual queries.
- Enhanced evaluation of system adaptability to varied levels of query granularity and scene complexity.
- A challenging dataset mix for developing and testing models robust to real-world scenarios.

By blending MSCOCO and Flickr30k, we ensure QT-FIRS is tested against a rich spectrum of visual and textual pairings, fostering a more robust, flexible, and scalable retrieval system.

	Flickr30k		MS-COCO
S01	A boy holding a racket is standing in the grass.	S01	This is two men in cleats on the grass.
S02	A boy in a purple shirt and blue jeans is standing in the grass holding a round toy with a handle.	S02	Two young men playing frisbee in a park.
S03	A boy playes with a stringless racket in his backyard.	S03	A couple of men playing frisbee against each other on a field.
S04	A boy walks through the lawn.	S04	Two teenagers playing of game of Frisbee in a field.
S05	A child in purple clothing is playing with a toy inside a yard with other toys and a fence.	S05	Two young men play with a Frisbee in a grassy area.

Figure 1: An example of dataset images, showcasing the variety of visual and textual pairings from MSCOCO and Flickr30K, tested in the QT-FIRS system.

3 Methodology

3.1 Baseline

To tackle the challenges of Content-Based Image Retrieval (CBIR), our methodology is designed to bridge the semantic gap between textual queries and visual content. It leverages a contrastive learning framework to embed both text and images into a shared vector space, effectively capturing and utilizing semantic relationships for retrieval.

Traditional Vector Space Models, illustrated in Figure 2, assume that query and collection data share the same domain. However, they fail to address the intrinsic differences between textual and visual data. Text and images require unique feature extraction techniques to preserve semantic richness and differentiation. Our approach employs modern contrastive learning techniques to overcome these limitations, ensuring modality-specific processing while aligning the two in a unified representation.

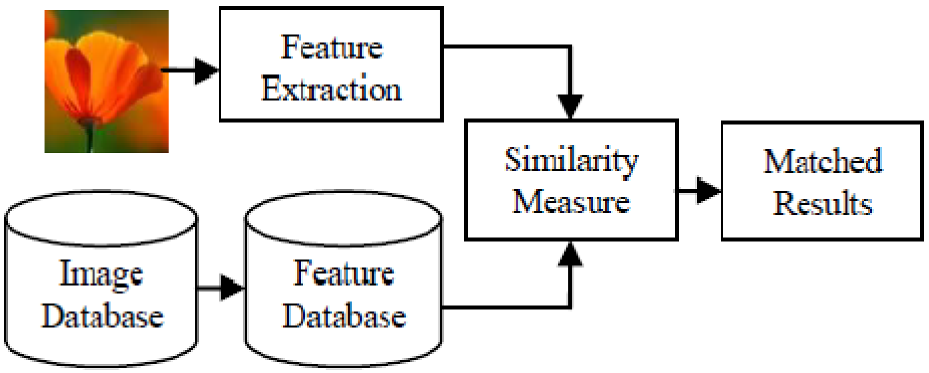


Figure 2: Example of a Content-Based Image Retrieval (CBIR) system, illustrating how visual data is processed and retrieved based on its content, rather than relying solely on textual metadata.

Contrastive Learning Framework

The CBIR task is redefined as a contrastive learning problem with two primary processes:

- **Textual Data Representation (Query Side):** Textual queries are transformed into high-dimensional embedding vectors using a pre-trained text encoder. This step captures the semantic intricacies of natural language.
- **Visual Data Representation (Collection Side):** Visual data is processed using a distinct pre-trained image encoder, generating embeddings that encapsulate visual semantics in a structured form.

The textual and visual embeddings are projected into a shared vector space. Here:

- **Similarity Representation:** Semantically related text-image pairs are positioned closer together.
- **Disparity Management:** Semantically unrelated pairs are placed farther apart, preserving the clarity and distinction required for retrieval tasks.

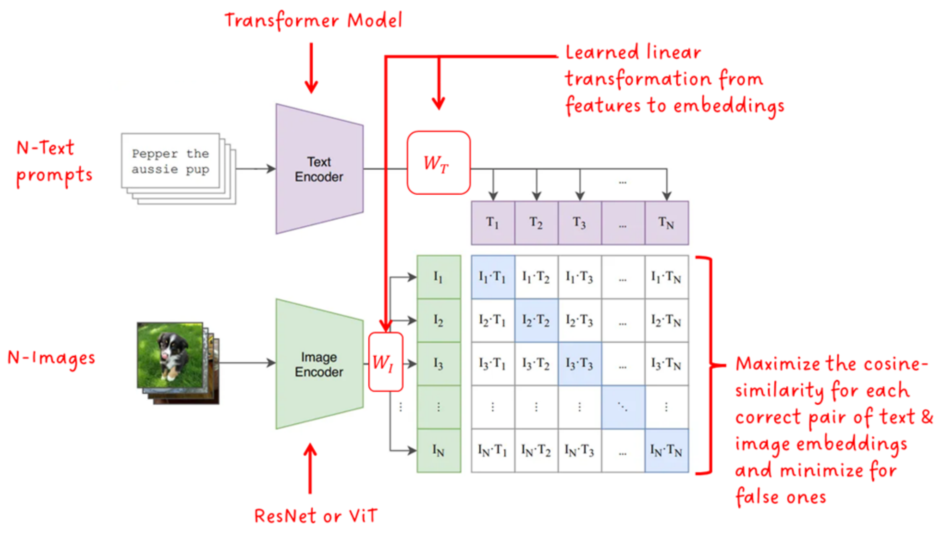
The similarity between the embeddings is measured using the cosine similarity metric, which evaluates the angular closeness of vectors efficiently.

The proposed methodology addresses the challenges of semantic alignment in Content-Based Image Retrieval (CBIR) by employing dedicated encoders for text and image modalities, ensuring the extraction of domain-specific, meaningful features. By mapping both modalities into a unified embedding space through contrastive learning, the approach reduces the semantic gap, enhances result interpretability, and enables efficient similarity computations for large-scale datasets. Moreover, its adaptability integrates advanced retrieval techniques like semantic hashing and collection expansion, enhancing system flexibility for real-world applications. Leveraging pre-trained models such as CLIP and SigLIP ensures robust cross-modal alignment, efficient feature extraction without training, and a flexible foundation for incorporating task-specific refinements. This framework effectively bridges the inherent differences between text and image data, offering a robust, interactive, and scalable CBIR system.

3.2 Contrastive Models

Contrastive learning is a central concept for aligning representations in multimodal models by maximizing the similarity between positive pairs (related text and image) and minimizing it for negative pairs (unrelated pairs). In the context of CBIR, this ensures meaningful retrieval by preserving semantic relationships across modalities. Two prominent models used in this framework are CLIP and SigLIP, each with distinct strengths and limitations.

CLIP [1] (Figure 3), developed by OpenAI, employs a contrastive learning architecture with two main encoders: an Image Encoder for extracting visual features and a Text Encoder for capturing semantic nuances in text. Both encoders map their respective modalities into a shared embedding space where similarities are measured using cosine similarity. CLIP's training process optimizes embeddings by adjusting them based on correct (positive) and incorrect (negative) pair similarities within each batch. This enables tasks such as zero-shot classification and image retrieval without task-specific fine-tuning, making it highly versatile. The strength of CLIP lies in its ability to generalize effectively from large-scale pretraining across diverse multimodal datasets.



Transformer Model
Learned linear transformation from features to embeddings
N-Text prompts
Pepper the aussie pup
Text Encoder
W_T
T_1 T_2 T_3 ... T_N
N-Images
Image Encoder
W_I
I_1 I_2 I_3 I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
Maximize the cosine-similarity for each correct pair of text & image embeddings and minimize for false ones
ResNet or ViT

Figure 3: Overview of CLIP's model architecture, highlighting its dual encoder design and contrastive learning mechanism for bridging visual and textual modalities. The model's ability to operate in a shared embedding space underpins its effectiveness in various vision-language tasks.

SigLIP [2], an enhancement of CLIP, introduces a significant modification by replacing the softmax-based normalization with a sigmoid loss function. This change eliminates the dependency on global pairwise similarity views, enabling SigLIP to operate efficiently at smaller batch sizes. The model performs robustly with batches smaller than 32,000, which is computationally advantageous compared to CLIP. Moreover, SigLIP achieves comparable or better results without the computational overhead required by CLIP's softmax normalization, making it more scalable for smaller datasets or constrained computational environments. However, at extremely large batch sizes ($> 256k$), both models tend to experience reduced performance.

In summary, both models leverage contrastive learning for aligning image and text representations effectively, but their design differences address specific scenarios. CLIP provides a robust general-purpose framework for multimodal retrieval, while SigLIP offers improved efficiency for scenarios demanding smaller computational resources, thus complementing the retrieval system's flexibility.

3.3 Ranking Aggregation

To aggregate rankings from multiple ordered lists, we implement a voting method that consolidates scores for each item based on its rank in the individual lists. A logarithmic weighting scheme is employed, where the top-ranked item in a list receives a score of 2.0, and lower-ranked items are weighted according to the formula $s_i = \log(1/(i+1))$, where i is the item's index. Scores from all lists are accumulated, ensuring that items appearing in multiple lists are weighted proportionally to their ranks. The final ranking is then determined by sorting items in descending order of their accumulated scores. This method balances contributions across lists while emphasizing higher-ranked items in the CBIR task.

3.4 Iterative Retrieval

Iterative retrieval serves to progressively enhance image retrieval results by refining query embeddings based on user feedback. This step is crucial for addressing ambiguous queries and adapting to the diverse needs of users in Content-Based Image Retrieval (CBIR). It ensures that the retrieval system becomes more precise and user-centric with each iteration. The process utilizes the Rocchio algorithm to update the query vector as follows:

$$\text{newQuery} = \alpha \cdot \text{query} + \beta \cdot \text{centroid(positive)} - \gamma \cdot \text{centroid(negative)} \tag{1}$$

Where α, β, and γ control the relative influence of the original query, positive examples, and negative examples, respectively. Centroids represent the aggregated embeddings of each feedback group.

Iterative retrieval is integral to refining Content-Based Image Retrieval (CBIR) systems by learning user preferences and dynamically adapting the retrieval process. It enhances the retrieval precision by incorporating user feedback to refine query embeddings. Positive feedback emphasizes semantically relevant results, ensuring that retrieved images match user intent more effectively. Negative feedback, on the other hand, eliminates irrelevant images, tailoring results to meet specific user needs. This user-centric process not only improves the accuracy of retrieved results but also increases the system's flexibility in handling ambiguous or evolving queries. The iterative nature of retrieval ensures a progressive alignment between the user's intent and system output, making the experience interactive and adaptive.

To simulate feedback during experiments, we employ *pseudo feedback*, treating the top P retrieved images as positive and the bottom P as negative. While this approach simplifies evaluation, it may diverge from outcomes observed with genuine user feedback.

3.5 Leveraging LLMs for Query Refinement

The integration of Large Language Models (LLMs) [5] significantly enhances query refinement by bridging gaps between user intent and retrieval outputs. LLMs add value by generating precise captions and fusing them with queries, yielding improved accuracy and interpretability in CBIR tasks.

Caption Generation

LLMs generate succinct, contextually relevant descriptions of retrieved images. These captions encapsulate key visual elements such as objects, actions, and settings. For example:

- *Image Input:* A young boy holding a kite in a park.
- *Generated Caption:* "A boy holding a kite in a sunny park."

Caption Fusion and Query Refinement

The system synthesizes these captions into a semantically enriched query that integrates the user's initial input and retrieved image descriptions. This step ensures:

- **Enhanced Clarity:** Ambiguous queries are clarified through descriptive elements.
- **Reduced Redundancy:** Conflicting or repetitive details are removed for concise refinement.

Illustrative Example:

- Initial Query: "A cat on a sofa."
- Generated Captions:
  - "A golden cat is sleeping on a sofa."
  - "A cat is relaxing on a gray sofa."
- Refined Query: "A golden cat relaxing on a gray sofa."

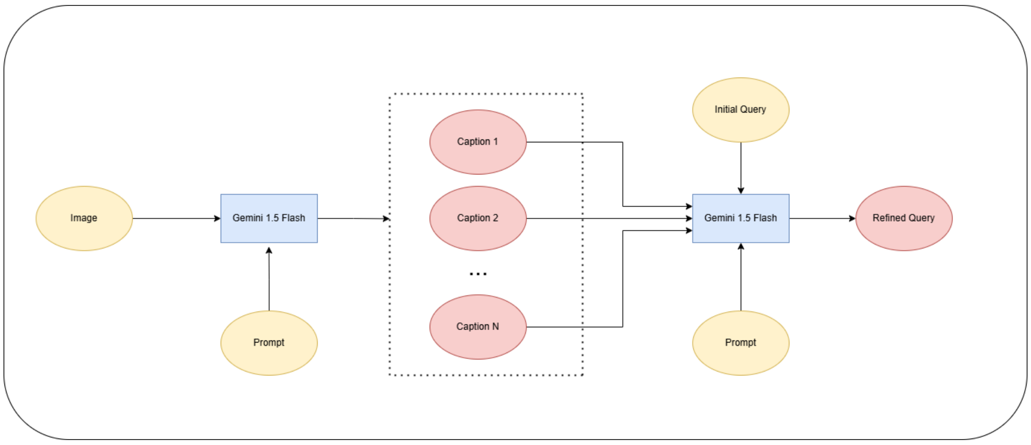


Figure 4: Query Refinement using Gemini 1.5 Flash.

The integration of Large Language Models (LLMs), shown in Figure 4 significantly narrows the intention gap and elevates user engagement within CBIR systems. LLMs refine textual queries by generating precise and contextually relevant captions for retrieved images. These captions, which succinctly encapsulate visual elements such as objects, actions, and scenes, are synthesized with user input to produce semantically enriched queries. By reducing ambiguity and resolving conflicting or redundant query details, LLMs ensure improved clarity and accuracy.

Moreover, this approach enhances flexibility by enabling users to iteratively refine their queries, adapting the system's outputs to evolving preferences. The framework also ensures adaptive brightness through its robust API, allowing seamless application across diverse CBIR scenarios. By integrating LLMs, QT-FIRS delivers semantically rich, contextually aligned retrieval results, offering a more interactive, dynamic, and engaging user experience.

3.6 Semantic Hashing

While the above approach provides real-time performance at the current size of the collection, we assume scenarios where the scale of image collections increases to potentially millions of images. QT-FIRS addresses this challenge through vector compression techniques, with semantic hashing being a particularly effective solution.

Semantic hashing involves converting high-dimensional embedding vectors into compact binary representations, termed bitsets. Instead of storing embeddings as float64 vectors, which are computationally intensive and consume significant storage, semantic hashing encodes each dimension of the vector into a single binary value. For a given dimension, the binary value is assigned based on whether the embedding's value exceeds a predefined threshold, such as the mean or median of the feature values. Dimensions greater than the threshold are mapped to 1, while those below are mapped to 0.

This binary encoding method offers several advantages:

- **Storage Efficiency:** Compression reduces storage requirements by a factor of **64**, as each dimension is represented by a single bit rather than a 64-bit float value.
- **Computational Speed:** Retrieval speed is enhanced by replacing cosine similarity calculations with lightweight **bitwise operations** (e.g., XOR) for measuring vector similarity.
- **Scalability:** The approach is particularly advantageous for devices with limited hardware resources, as it minimizes memory usage and computational overhead.

The trade-off between accuracy and efficiency is carefully managed through the semantic hashing process. While this method compresses the embeddings, its design ensures that the most meaningful information for similarity comparisons is preserved, maintaining high retrieval performance even for large-scale datasets. By leveraging semantic hashing, QT-FIRS achieves a significant improvement in scalability and response time, making it well-suited for real-world applications with extensive image collections.

3.7 Collection Expansion

Similar to Query Expansion, we hypothesize that Collection Expansion can enhance query results. To achieve this, we introduce textual captions into the image collection, effectively creating a combined collection that includes both images and their corresponding textual descriptions. This approach is feasible because the contrastive framework enables both images and text to be represented in the same form, embedding vectors.

To generate the textual captions, we can leverage user queries as descriptions for related images in the dataset. Additionally, Large Language Models (LLMs) can be utilized to automatically generate captions for images, transforming the system into a dynamic, self-updating repository. This dynamic approach not only enhances the retrieval process but also ensures that the system stays aligned with emerging trends and contexts.

An alternative strategy involves refining the collection's embedding vectors using the Rocchio Formula rather than directly adding textual captions. This iterative process enhances the image collection's representation over time based on user interactions, allowing the system to adapt more effectively to new data.

This approach enables the system to better capture different descriptive facets of each image, especially when not all images in the collection receive equal query attention. At any given moment, some images, referred to as "hot" images, are queried frequently by multiple users, while the majority of the images might rarely be accessed. By enriching the collection and continuously refining the embeddings, we enhance the system's adaptability to both trending images and the broader collection.

4 Evaluation Experiment

4.1 Enviroment and Configuration Setup

The experimental framework is implemented utilizing widely acknowledged tools and libraries to ensure reproducibility and standardization. Below, we detail the environment and configuration pivotal for the study.

Our implementation leverages **PyTorch** as the primary deep learning framework, facilitated by its seamless integration with key libraries such as **Hugging Face**, **Transformers** for downloading pre-trained encoder models, and **Datasets** for handling, storing, and loading large-scale datasets. Additional libraries include **NumPy** for numerical computation and representation, **Scikit-learn** for machine learning techniques, and **Pillow** (PIL) for image processing. Moreover, **Google-generative API** is also integrated to leverage the capabilities of Large Language Models.

Configuration:

- CLIP version
- SigLIP version
- LLMs version: Gemini-1.5-flash, temperature: 0.4

4.2 Workflow

The workflow is designed to streamline the processes of feature extraction, storage, and retrieval, ensuring efficiency and scalability. During the preparation stage, each image in the collection is passed through an image encoder to generate embedding vectors that encapsulate its semantic and visual attributes. These embeddings are precomputed and stored in a structured NumPy matrix for efficient access during retrieval. For a query input, typically in textual form, a corresponding query embedding is obtained through a text encoder, mapping the query into the same embedding space. The similarity between the query embedding and stored embeddings is computed using a predefined metric, such as cosine similarity, to rank the stored items. This ranking process ensures that the most relevant results are retrieved and presented. The separation of embedding computation from retrieval not only enhances processing speed but also allows the system to scale effectively with larger datasets, aligning seamlessly with the experimental framework's goals.

4.3 Metrics

To evaluate the effectiveness of our retrieval system, we employ the **Recall@K** metric, with $K = 1, 5, 10$. This metric is particularly suited for our setup, where each image in the dataset is described by multiple captions (five captions per image), and each textual query (captions) corresponds to only one unique relevant image. Recall@K assesses the fraction of queries for which the relevant image appears within the top K retrieved results, emphasizing the importance of high-ranking positions (**rank sensitivity**). Additionally, we record the rank of the relevant image for each query and calculate the mean and median rank across all queries. These statistics provide valuable insights into the overall ranking quality of the system, ensuring a comprehensive evaluation aligned with the study's objectives.

5 Baseline Evaluation

The baseline evaluation compares the effectiveness of different approaches in varied scenarios using a random subset of 5,605 queries (3,100 from Flickr30k and 2,505 from MSCOCO) over a blended collection of 36,014 images. The evaluation incorporates iterative feedback retrieval to assess performance improvements. For queries where the relevant results do not appear in top rankings (breaking rank = 1), we refine the query using the Rocchio algorithm, selecting the top 3 pseudo-positive images ($P = 3$).

Table 1: CLIP and SigLIP Metrics with Iterative Retrieval

Model	Iteration	R@1	R@5	R@10	Med r	Mean r
CLIP (no-iter)	0	23.10	43.66	53.17	9.00	196.76
CLIP (iter)	1	26.16	40.16	45.80	16.00	431.92
SigLIP (no-iter)	0	40.36	62.77	70.98	2.00	107.10
SigLIP (iter)	1	43.98	60.23	66.48	2.00	133.51

Results demonstrate that **SigLIP** significantly outperforms **CLIP** in all metrics. Iterative refinement increases performance at top rankings but does not improve average rank for all queries. CLIP, without iteration, achieves $R@1 = 23.10\%$, $R@5 = 43.66\%$, $R@10 = 53.17\%$, and a median rank of 9. After one iteration, $R@1$ improves to 26.16%. In contrast, SigLIP starts with $R@1 = 40.36\%$, $R@5 = 62.77\%$, $R@10 = 70.98\%$, and a median rank of 2. Post-iteration, $R@1$ rises to 43.98%. This indicates a better baseline performance for SigLIP.

We further tune hyperparameters such as the number of top positive images and breaking rank to assess the SigLIP's sensitivity to these parameters. Results on a smaller query subset reveal that increasing top positives and allowing higher breaking rank enhance $R@10$, but changes in $R@1$ and mean rank remain minimal. The findings underscore SigLIP's robustness and the impact of iterative feedback retrieval in fine-tuning query results.

Table 2: Performance Metrics Across Different Configurations (Part Table)

Iteration	Top Positive	Breaking Rank	R@1	R@5	R@10	Mean r
0	3	1	41.95	63.32	71.51	103.79
1	3	1	45.27	60.29	67.41	132.21
2	3	1	46.24	59.51	64.98	181.09
0	3	5	41.95	63.32	71.51	103.79
1	3	5	41.95	64.98	70.05	131.06
2	3	5	41.95	65.17	68.78	177.50
0	3	10	41.95	63.32	71.51	103.79
1	3	10	41.95	63.90	74.44	129.04
2	3	10	41.95	63.90	74.63	170.66
0	3	15	41.95	63.32	71.51	103.79
1	3	15	41.95	63.41	72.98	127.62
2	3	15	41.95	63.41	72.98	166.85
0	5	1	41.95	63.32	71.51	103.79
1	5	1	45.27	62.83	68.20	131.57
2	5	1	46.54	61.46	66.73	179.71
x	x	x	x	x	x	x
0	15	15	41.95	63.32	71.51	103.79
1	15	15	41.95	63.51	72.29	120.03
2	15	15	41.95	63.51	72.29	154.81

5.1 LLMs Query Refinement Evaluation

To evaluate the effectiveness of the LLMs Query Refinement approach, we compare the retrieval results obtained from SigLIP with those refined by LLMs. In this study, the top $P = 3$ images from the initial retrieval serve as pseudo-feedback to refine the query. However, due to API limitations, the fast evaluation is conducted on a subset of 60 queries, consisting of 30 from the MSCOCO dataset and 30 from the Flickr dataset.

Table 3: Performance Comparison: Baseline Retrieval vs. LLMs Refinement

Metric	**Baseline Retrieval**	**LLMs Refinement**
R@1 (%)	45.00	48.33
R@5 (%)	68.33	56.67
R@10 (%)	76.67	63.33
Median Rank	2.00	2.00
Mean Rank	100.85	223.45

The results indicate that similar to Rocchio Iterative Retrieval, LLMs Query Refinement improves retrieval performance at the top rank, as reflected in higher Recall@1 scores. However, it does not consistently enhance the average rank for all queries. A notable advantage of this approach lies in its ability to propose refined queries to users, offering greater interpretability and fostering user engagement. This makes the system more user-friendly, as it allows users to understand and interact with the retrieval process meaningfully, beyond simply selecting and clicking on images.

Nevertheless, the approach has limitations. The computational cost of LLMs Query Refinement is significantly higher than Rocchio Iterative Retrieval, given the complexity of processing image and text data through LLMs. Additionally, reliance on API calls introduces constraints, especially when dealing with non-paid plans, which may limit the scalability of the solution for large-scale applications.

5.2 Semantic Hashing Evaluation

Table 4: Performance Comparison for Semantic Hashing Evaluation

Threshold	**Iteration**	**R@1 (%)**	**R@5 (%)**	**R@10 (%)**	**Med rank**	**Mean rank**
Baseline	0	40.36	62.77	70.98	2.00	107.10
Baseline	1	43.98	60.23	66.48	2.00	133.51
Median	0	21.25	39.82	49.06	11.00	285.84
Median	1	21.59	39.88	48.47	12.00	261.79
Mean	0	21.27	40.09	48.80	11.00	283.49
Mean	1	21.50	39.84	48.94	12.00	268.25

Semantic hashing significantly reduces storage costs and computational complexity by converting high-dimensional embedding vectors into compact binary representations, enabling fast bitwise operations for similarity computation. However, this compression introduces a trade-off between efficiency and retrieval accuracy. To evaluate this trade-off, we compared system performance using different thresholding methods (**mean and median**) over multiple iterations. Results show that while retrieval accuracy slightly decreases with binary thresholds (**e.g.**, R@1: 21.25% for median threshold, iter=0) compared to floating-point embeddings (**e.g.**, R@1:

40.36% for non-hashed retrieval), the method achieves up to **8-to-64 fold compression** in storage and up to **4-to-8 fold computation speed-up**. Future studies may explore multi-level thresholds, such as 10 or 256 bits per dimension, balancing increased storage and computational costs with potentially higher retrieval accuracy. This demonstrates semantic hashing's practicality for scalable image retrieval systems while highlighting areas for further optimization.

5.3 Collection Expansion Evaluation

The evaluation of the Collection Expansion approach leverages a subset of textual captions from the dataset to simulate the expansion process. Specifically, we use two out of five captions per image to augment the collection, while the remaining three captions are used as queries. This setup ensures an objective evaluation under the out-of-sample condition, aligning with standard testing protocols.

Table 5: Performance Comparison of Collection Expansion vs. No Expansion

Method	**Iteration**	**R@1**	**R@5**	**R@10**	**Med r**	**Mean r**
No Expansion	0	43.24	66.07	74.04	2.00	89.55
	1	46.71	63.40	69.31	2.00	112.05
Expansion	0	39.22	59.95	67.83	3.00	192.03
	1	41.75	58.16	64.70	3.00	170.06

The results indicate that Collection Expansion does not consistently meet expectations, as observed from the decline in Recall@K metrics when compared to the baseline without expansion. For instance, with no expansion, the performance achieves R@1: 43.24%, R@5: 66.07%, and R@10: 74.04%, while expanded collections report decreased scores such as R@1: 39.22%, R@5: 59.95%, and R@10: 67.83%. Furthermore, the median rank increases, suggesting a degradation in retrieval accuracy.

The reduced performance can be attributed to the excessive addition of textual captions (72,028 captions compared to 36,014 images), inadvertently introducing noise into the collection. This dilution affects evaluation results, particularly as the evaluation focuses on a small subset of frequently queried simulated "hot" images (1,121 images with 3,363 queries). These findings highlight the importance of limiting expansion to a controlled threshold, prioritizing trending images, and minimizing noise.

In conclusion, while Collection Expansion holds the potential for enhancing retrieval, the results underscore the necessity of balanced strategies. Future implementations should aim for a balanced enrichment process, emphasizing the relevance and adaptability of the system to trending content.

6 Open Discussion

6.1 Referring Image Segmentation Module

To enhance the interpretability of the system's retrieval results, we integrate a Referring Image Segmentation [6] module utilizing the pre-trained model from CLIPSeg [7]. This module, demonstrated in Figure 5, 6 allows the system to segment specific objects mentioned in the query within the returned images. By highlighting these relevant regions, the feature enables users to quickly filter essential information and understand why the system considers certain images relevant.

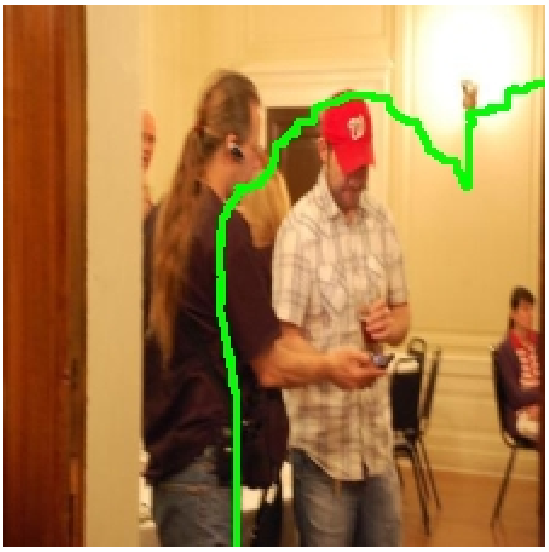


Figure 5: Query: Red hat people

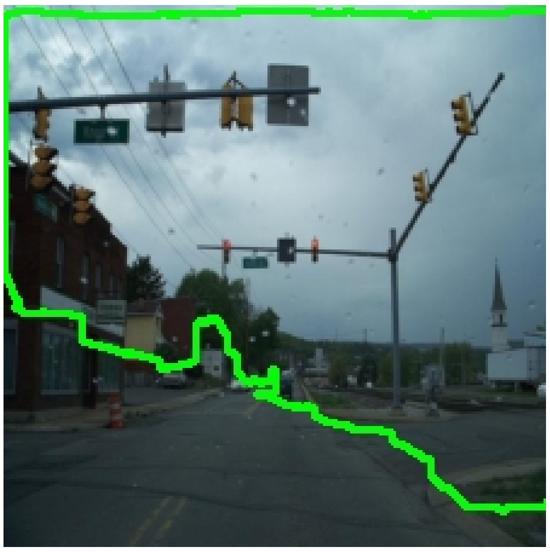


Figure 6: Query: A couple of traffic lights sitting under a cloudy sky

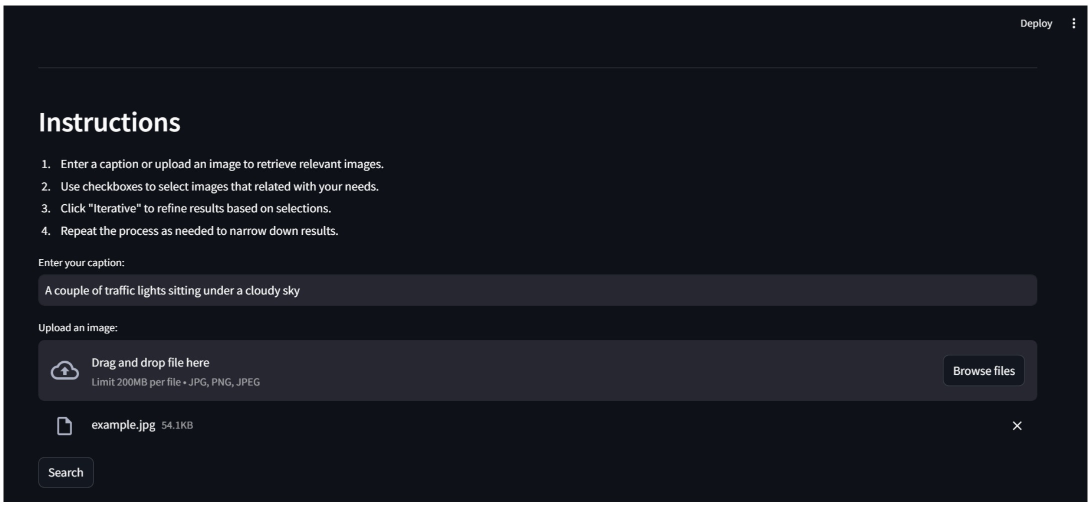
However, the primary limitation of this module lies in its computational cost for performing segmentation. Consequently, it is selectively applied to a small subset of the returned images that are considered most relevant to the user's query needs. This targeted approach ensures the module remains efficient while improving user comprehension and experience.

6.2 Multimedia Query Focus by Two-Phase Retrieval

Some concepts are challenging to describe purely in textual form but can be effectively illustrated with images. To address this intention gap, we allow users to enhance their queries by incorporating example images, enabling multimedia queries that combine text and image inputs.

To handle such queries, we implement a Two-Phase Retrieval strategy. In the first phase, the system uses the textual query to retrieve the top M relevant results. In the second phase, we rerank these M results based on the user-provided example images. This reranking is performed using the Ranking Aggregation method described earlier in this report.

This approach ensures that no significant results are overlooked while enabling the system to focus more on results relevant to the user's illustrative examples. It effectively bridges the gap between multimodal inputs and improves the retrieval accuracy and relevance for complex concepts.



Deploy

Instructions

1. Enter a caption or upload an image to retrieve relevant images.
2. Use checkboxes to select images that related with your needs.
3. Click "Iterative" to refine results based on selections.
4. Repeat the process as needed to narrow down results.

Enter your caption:

A couple of traffic lights sitting under a cloudy sky

Upload an image:

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

Browse files

example.jpg 54.1KB

Search

Figure 7: Use both text and example images to query.

Table 6: Performance Comparison on ImageNet1k and MSCOCO From SigLIP paper

Method	**ViT Size**	**# Patches**	**Validation**	**v2**	**ReaL**	**ObjectNet**	**I → T**	**T → I**
CLIP	B	196	68.3	61.9	-	55.3	52.4	33.1
OpenCLIP	B	196	70.2	62.3	-	56.0	59.4	42.3
EVA-CLIP	B	196	74.7	67.0	-	62.3	58.7	42.2
SigLIP	B	196	76.2	69.6	-	70.7	64.4	47.2
SigLIP	B	256	76.7	70.0	83.1	71.3	65.1	47.4
SigLIP	B	576	78.6	72.1	84.5	73.8	67.5	49.7
SigLIP	B	1024	79.2	73.0	84.9	74.7	67.6	50.4
CLIP	L	256	75.5	69.0	-	69.9	56.3	36.5
OpenCLIP	L	256	76.4	70.4	-	66.4	62.1	41.6
CLIPA-v2	L	256	77.9	72.8	-	71.1	64.1	47.1
EVA-CLIP	L	256	79.8	72.9	-	75.3	63.7	47.5
SigLIP	L	256	80.5	74.2	85.9	77.9	69.5	51.1
CLIP	L	576	76.6	72.0	-	70.9	57.9	37.9
CLIPA-v2	L	576	80.3	73.0	-	74.3	65.5	47.9
EVA-CLIP	L	576	82.0	75.7	-	82.3	68.8	51.1
SigLIP	L	576	82.1	75.9	87.0	81.0	70.6	52.7
OpenCLIP	G (2B)	256	80.1	73.6	-	73.0	67.3	51.4
CLIPA-v2	H (630M)	256	80.3	73.8	-	75.9	66.8	51.1
EVA-CLIP	E (5B)	256	82.0	75.7	-	82.0	68.8	51.4
SigLIP	SO (400M)	729	83.2	77.2	87.5	82.9	70.2	52.0

6.3 Challenges and Contributions

Blending two different datasets into a unified collection introduces significant complexity, as evidenced by the notable performance drop in various models compared to published results from prior scientific literature, shown in Table 6. This challenging collection also poses difficulties for the proposed strategies, which have not yet fully met the expected outcomes. Nevertheless, the strategies presented in this report, including Ranking Aggregation, LLM-based Query Refinement, Semantic hashing, and Collection Expansion, as well as Referring Image Segmentation, Multimedia Query with Two-Phase Retrieval offer foundational approaches for tackling these challenges with various case studies. The diverse experimental evaluations in this work are expected to contribute meaningfully to future research efforts and improvements, while also advancing the field of Content-Based Image Retrieval (CBIR).

7 Acknowledgment and Inspiration

We deeply thank our professor PhD. Thanh Duc Ngo for the insightful and high-quality discussions on mindset, life, and various topics covered during the course. These sessions provided me with fresh perspectives, innovative ideas, and a new way of thinking about both academic and life challenges. The techniques and methods presented in this report are not advanced but originate from the inspiration provided during the lectures and the key questions raised in the final exam. While this project still contains limitations, it represents my efforts to address the essential topics discussed throughout the course. I sincerely thank my professor for the guidance and wish the professor and their families happiness, health, and further success in the coming new year.

References

[1] A. Radford, J. W. Kim, C. Hallacy, M. Rytov, G. Sastry, A. T. Sumar, S. Subramanian, X. Wang, A. See, P. Chen, A. Neelakantan, P. Shyam, E. Chen, K. J. R., and I. Sutskever, “Clip: Learning transferable visual models from natural language supervision,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 11 899–11 909.

[2] C. Wang, P. Zhang, Z. Ma, and Y. Yang, “Siglip: Sigmoid loss for language image pre-training,” in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1393–1402.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Kohli, and D. R. Anderson, “Microsoft coco: Common objects in context,” 2014.

[4] B. Plummer, Y. Shi, X. Chen, D. Parikh, and D. Batra, “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1997–2005.

[5] S. Liu, X. Lu, S. Feizi, and A. Loukas, “An overview of large language models,” *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 1, pp. 33–42, 2023.

[6] T. Yao, Y. Li, Y. Jiang, S. Zhang, Z. Xie, and X. Lin, “Towards robust referring image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5374–5383.

[7] M. H. Lindner, L. H. J. Min, and Y. Bisk, “Clipseg: Clip supervision for semantic segmentation of textual prompts,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 652–15 661.