Image Based Search Engine - Using BM25 and Text Annotations

Ayesha Ishrath
Dublin City University
Dublin, Ireland **Student Number: 14224**
ayeshaishrath2@mail.dcu.ie

Abstract

This project realizes a functional prototype of an image search engine from Google Image Search's early days. The system has three components: a custom web crawler, a search engine based on BM25, and a minimal web interface. The web crawler periodically retrieves and indexes a set of over 1,500 high-quality, nature-themed images from publicly available online resources. All images are provided with related metadata, such as filenames, alt-text, and context webpage text of the image, to enable efficient retrieval. Underlying, the search engine is driven by the BM25 ranking algorithm, a classical information retrieval strategy ranking images by textual relevance of supporting metadata to the queries submitted by users. The system offers a basic, web-based, user-responsive interface where users provide search keywords and navigate ranked image results.

As a stretch goal, the project integrates OpenAI's CLIP (Contrastive Language–Image Pre-training) model to enrich the annotation and ranking process. By generating and analyzing joint text-image embeddings, the system enhances retrieval accuracy through the combination of visual semantics and textual cues. This hybrid approach results in more contextually relevant and visually aligned search outcomes, offering a significant improvement over text-only methods.

Overall, the project demonstrates the viability of building an efficient image search engine by combining traditional information retrieval techniques with contemporary AI techniques.

1 Introduction

Image retrieval has become a fundamental component of modern information systems, enabling users to search and browse visual data by natural language queries. In this project, we tackle the overlap of traditional text-based search techniques and visual content indexing by extending a classical search engine to the visual domain. Rather than being founded solely upon raw image features, the system indexes and retrieves images on the basis of textual surrogates—descriptive text pertaining to each image, such as alt-text, filenames, and semantic annotations.

The system architecture is modular and consists of several important components: a custom web crawler, an annotation module for augmenting image metadata, a retrieval mechanism founded upon the BM25 ranking algorithm, and a full web interface for user interaction.

To construct the image corpus, a Python-based crawler was developed using tools like Selenium and the requests library. The crawler was executed to crawl over 1,500 nature-based images from Google Images. Apart from downloading the image URLs, the crawler also collected surrounding HTML metadata (e.g., alt-text and captions) to be utilized as initial textual descriptors.

To improve the quality and depth of these annotations, the system incorporates and uses OpenAI's CLIP (Contrastive Language–Image Pre-training) model. CLIP allows the generation of semantic descriptors for each image by embedding both visual and textual information into a shared vector space. Hence, allowing the system to automatically assign high-level textual labels to each image based on its visual content, thereby capturing abstract concepts and improving retrieval accuracy.

The completed dataset, composed of both metadata and CLIP-generated annotations, is

then indexed within a structured JSON-based index. User queries at retrieval time are matched against the index by a bespoke implementation of the BM25 ranking function, scoring and ranking the images by textual relevance of their annotations.

This approach supports effective and scalable image searching under a familiar search paradigm while leveraging the newest AI techniques to bridge the language-vision divide.

Architecture Overview:

- **Crawler:** Custom-built using Selenium and requests
- **Annotation:** HTML alt-text + CLIP vision-based labels
- **Indexing:** JSON-based lightweight index
- **Retrieval:** Custom BM25 ranking function
- **Web UI:** Flask + responsive HTML

2 Annotation

All images within the dataset are annotated according to two complementary sources of information to produce effective textual surrogates that can be indexed and searched. The dual-annotation process allows the system to combine traditional metadata with machine-generated semantic meaning so that it generates more accurate and contextually appropriate responses in search.

2.1 HTML Metadata Extraction

RThe primary source of annotation relies on metadata that already exists in the HTML source of Google Images search results web pages which the images are being crawled from. Whenever Google Images search result web pages are crawled, it attempts to extract the following information:

- **Alt-text:** Alt-text is an HTML attribute which generally provides one with a description text for an image, for accessibility. It tells about the image contents and can be a very useful one for searching and indexing.
- **Image Filenames or Captions:** If there is no alt-text available, then the crawler will try to use image filenames, or nearby texts, or captions provided in the HTML. These provide additional hints about the content of the image, but they may vary in quality and consistency.

While HTML metadata can be meager or noisy, it is a good seed, especially for publisher-manually tagged or described images.

2.2 CLIP-Based Semantic Annotation (Stretch Goal)

To enhance and standardize image annotations beyond what is available in the raw HTML(alt-text) , the project integrates a more advanced, AI-driven annotation method using OpenAI's CLIP model, specifically, the clip-vit-base-patch32 variant. CLIP (Contrastive Language–Image Pre-training) is a very powerful vision-language model trained to understand and relate images and natural language descriptions in a shared embedding space.

Here, the model is being used in a zero-shot classification system. A list of descriptive prompts—is carefully selected and given, for instance:

- "a mountain landscape"
- "a tropical island"
- "a dense forest"
- "a waterfall in nature"
- "a snowy mountain range"

—is fed into CLIP along with every image. The model generates similarity scores between the image and every prompt, effectively ranking the prompts in terms of how well they characterize the visual content. The top-scoring captions are treated as annotations for that image.

This approach enables the system to generate semantically rich and coherent descriptions, even for images lacking useful HTML metadata. It also allows for greater comprehension of the content by encoding abstract or high-level visual concepts, which would be difficult to deduce using traditional methods.

3 Indexing

The indexing module is the central component of the retrieval system, transforming annotated image data into a computationally tractable, structured format which can be searched and ranked. To this end, the choice was made to use a lightweight, human-readable JSON format to store all data about each image.This method is adaptable. It makes the debugging process easier and also facilitates easy integration with the web and search modules.

Each record in the JSON index contains the following fields:

- **Image Path:** A local pointer to where the image resides or is being accessed, presented as the web view.
- **Improved Textual Surrogate:** Combined text field of all textual descriptors of the image containing:
  - Extracted HTML metadata (alt-text, file names, captions).
  - CLIP-synthesized semantic labels (e.g., "a tropical beach", "a snow-covered mountain"). This is the master content used in matching against user queries.
- **Original URL:** The original URL where the image was retrieved. This is kept for attribution and traceability.

3.1 Preprocessing Pipeline

To ensure consistent and reliable matching during retrieval, all textual data is preprocessed using a series of normalization steps:

- **Tokenization:** Tokenization is the process of splitting text into individual words or terms (tokens) that can be indexed.The surrogate text is split into unique tokens (words or significant phrases) using the whitespace and punctuation symbols as separators.
- **Lowercasing:** The entire collection of tokens or words is converted to lowercase case in order to prevent case sensitivity during the matching process.
- **Punctuation Removal:** Commas, special characters, periods, and other undesired punctuation marks are removed in order to minimize the set of tokens to their core and remove noise.

Notably, stopword removal (i.e., removing high-frequency words like "the", "is", "in") and stemming (converting words to their base forms, e.g., "running" $\rightarrow$ "run") were deliberately avoided during the development of this system. This was to preserve the full semantic nuance of the textual surrogates, most notably the CLIP-generated phrases, which have a tendency to rely on specific word sequences in order to denote accurate visual meaning. Removing or altering these words would most probably significantly decrease the effectiveness of semantic matching.

3.2 Design Rationale:

By employing a light, non-relational JSON data format, one can:

- Iterate quickly and develop rapidly.
- Inspect indexed data easily.
- Integrate with Python-based search logic trivially.

In short, the indexing step takes raw annotations and normalizes them into clean, searchable form, which preserves the contextual richness of the original and AI-augmented descriptions. This provides the assurance that user queries will be matched against a high-quality representation of every image's content, which in turn enables more meaningful and satisfying search results.

4 Retrieval

The core of the retrieval system used in this project is driven by the BM25 model, a robust probabilistic ranking function for information retrieval. BM25 is an extension of the underlying TF-IDF models, incorporating term frequency saturation and document length normalization to produce more accurate and unbiased relevance scores.

Here, BM25 is applied to match user queries with the enhanced textual surrogates of images (i.e., the combined alt-text, metadata, and CLIP-written descriptions). What is aimed to be determined is the amount each image is relevant to the query of the user based on the textual description of that image alone.

The BM25 scoring function is defined as follows:

$$\text{score: } q, D = \sum_{i=1}^{n} \text{IDF} q_i \cdot \frac{f q_i, D \cdot k_1 \ 1}{f q_i, D \ k_1 \cdot \left(1 - b \ b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

Where:

- $q = \{q_1, q_2, \ldots, q_n\}$ is the user query consisting of n terms.
- D is a document (e.g., the textual surrogate of an image).
- $f q_i, D$ is the frequency of query term q_i in document D.
- $|D|$ is the length of the document (total number of tokens).
- avgdl is the average document length in the corpus.
- $\text{IDF} q_i$ is the inverse document frequency of term q_i.
- k_1 and b are hyperparameters, usually chosen empirically.

[3]

5 Evaluation

The image retrieval system was evaluated for performance in terms of a combination of qualitative analysis and system-controlled observations on ranking. Because ground-truth relevance annotations were unavailable, formal quantitative assessment using standard measures such as precision@k or mean average precision (MAP) could not be conducted. Alternatively, evaluation focused on investigating the ability of the system to retrieve semantically related images given natural language queries through the BM25-based ranking framework.

5.1 Retrieval and Ranking Process

When it is queried by a user, the system performs retrieval from a corpus of enriched textual surrogates for each image. The surrogates are produced by merging metadata extracted from HTML (e.g., image file names, alt-text) with semantic labels produced using OpenAI's CLIP model. The search proceeds by following the below steps:

- **Query Preprocessing:** User input is tokenized, lowercased, and punctuation stripped to match the preprocessing at indexing time. Stopword removal and stemming are not performed to preserve the semantic consistency of the CLIP-generated phrases.
- **Document Matching:** Every image document is surfacted by its textual surrogate. The frequency of query terms in each surrogate is calculated by the system and the BM25 scoring function is used to approximate relevance.
- **BM25 Scoring:**

$$\text{score} q, D = \sum_{i=1}^{n} \text{IDF} q_i \cdot \frac{f q_i, D \cdot k_1 \ 1}{f q_i, D \ k_1 \cdot \left(1 - b \ b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

where:

$f q_i, D$ is the frequency of query term q_i in document D,
$|D|$ is the length of document D,
avgdl is the average document length in the corpus,
$k_1 = 1.5$ (term frequency scaling),
$b = 0.75$ (length normalization parameter).

[3]

- **Ranking and Output:** All the documents are ordered in descending order based on their BM25 scores. The top results are displayed to the user in the web interface, rendered in grid layout to allow for visual inspection.

6 Analysis of Results

A representative sample of queries with nature imagery (e.g., "snowy mountain", "tropical forest", "sunset over water") was used to manually

assess the quality of retrieval. For each query, the highest-ranked images were examined to take into account:

- **Semantic Relevance:** The extent to which the visual content of the query is matched.
- **Descriptive Matching:** Whether retrieved results are justified by their surrogate annotations.
- **Result Diversity:** To what extent several visual interpretations of the query are represented.

Overall, the system demonstrated consistent retrieval of semantically similar and applicable images, especially where CLIP annotations enhanced incomplete or ambiguous HTML metadata.

7 Discussion and Limitations

The combination of BM25 with CLIP-augmented text surrogates successfully bridged visual and textual modalities. Some limitations were, however, noticed:

- Results at times favored images with surrogates of larger lengths due to the term-matching bias of BM25.
- In the absence of user-annotated relevance data, systematic performance benchmarking remains an issue.

8 Conclusion and Future Work

The system performs well on the dataset captured and ranks the relevance score along with the retrieved image.

Future evaluation can be made stronger with a blend of user-centered and model-based methods. One way is to get explicit relevance judgments from users by asking them to rank the relevance of retrieved images for a set of predefined queries. This would enable the use of traditional IR metrics such as Precision@k, MAP, and nDCG.

A second method is gathering click-through data from the actual behavior of the user. Click behavior can serve as implicit feedback to indirectly fine-tune the ranking model and gauge user intent as time passes.

In addition, CLIP-based similarity scoring offers a scalable programmatic solution. By comparing the CLIP embedding of a text query against image embeddings, the model can approximate semantic alignment without needing human annotation. Such scores could be utilized to verify or augment BM25 ranking through pseudo-relevance feedback.

These methods taken together would give a more realistic and comprehensive setup for testing and iteratively improving the system.

9 Code and Implementation

For the above project implementation, the code can be found here: **GitHub Repository containing the code**.

The working link for search engiene hosted on Azure can be found here: **ImageSearchEngine**.

Suggested Prompts:

- Waterfall
- Trees
- Mountains
- Autumn
- Flowers

References

[1] Ashwath Krishnan, Sudhanva Rajesh and Shylaja SS *Text-based Image Retrieval Using Captioning.* 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT) Available at: https://ieeexplore.ieee.org/document/9616897.

[2] Annapurna P. Patil, Abhinav Benagi, Charith Rage, Dhanyatha Narayan, Susmitha A and Pragya Paramita Sahu *CLIP-Based Image Retrieval: A Comparative Study Using CEITM Evaluation.* 2024 1st International Conference on Communications and Computer Science (InCCCS) Available at: https://ieeexplore.ieee.org/document/10593420.

[3] Jinyin Zhang and Rongsheng Xie *Word2vec- Powered Algorithm for Efficient Retrieval of Bill of Quantities.* 2022 International Conference on Image Processing and Computer Vision (IPCV) Available at: https://ieeexplore.ieee.org/document/10405620.

[4] Youtube Video *Supercharge eCommerce Search: OpenAI's CLIP, BM25, and Python.* Available at: https://www.youtube.com/watch?v=AELtGhiAqio.

[5] YouTube Video. *Image Search in Python with OpenAI CLIP.* Available at: https://www.youtube.com/watch?v=S7VZErcTN5Y.