同济大学

Tongji University

CLIP Image-Text Search System: Implementation and User Experience

Course Name: Human-Computer Interaction
Author: 2351050 Ruichen Yang,
2350989 Qizheng Zhang

May 18, 2025

Abstract

This report describes the implementation of our image search system based on the **CLIP (Contrastive Language-Image Pre-Training)** model. The system enables users to search for images using either text descriptions or image queries, providing a rich and intuitive interface for exploring image collections. This implementation demonstrates how modern AI models can bridge the gap between linguistic and visual representations, creating more natural ways for users to interact with image databases.

1 Introduction to the Dataset

The application utilizes the **Grocery Store Dataset**, a specialized collection of images focused on grocery products. This dataset was sourced from the GroceryStoreDataset GitHub repository[1] and includes:

- Multiple product categories (fruits, vegetables, packaged goods, etc.)
- Approximately 5,000 images of grocery items
- Iconic images with clean backgrounds as well as real-world images
- Hierarchical organization by product categories

The dataset's structure includes training, validation, and test splits, though our application primarily utilizes the iconic images for search functionality. The relatively constrained domain of grocery products provides a good test case for evaluating the effectiveness of the CLIP model in distinguishing between visually similar but semantically distinct items.

2 System Architecture

The system utilizes a combination of state-of-the-art technologies:

1. **CLIP Model**: OpenAI's CLIP (clip-vit-large-patch14) provides the core functionality for encoding both images and text into a common embedding space.
2. **Upstash Vector Database**: Stores the pre-computed embeddings of all images in the dataset, enabling fast similarity search.
3. **Gradio**: Powers the user interface, providing interactive components for querying and displaying results.

The application follows a modular architecture where image embeddings are precomputed and stored, allowing for efficient real-time searching without needing to regenerate embeddings for the entire dataset with each query.

3 User Interface and Experience

3.1 Design Principles

The design of the user interface (UI) for the CLIP Image-Text Search System was guided by the **Five-Stage Search Framework** (Formulation, Initiation, Review of Results, Refinement, Use) to ensure a comprehensive and intuitive user experience.

[1] `https://github.com/marcusklasson/GroceryStoreDataset`

Our interface(Fig: 1) explicitly supports each stage of the framework:

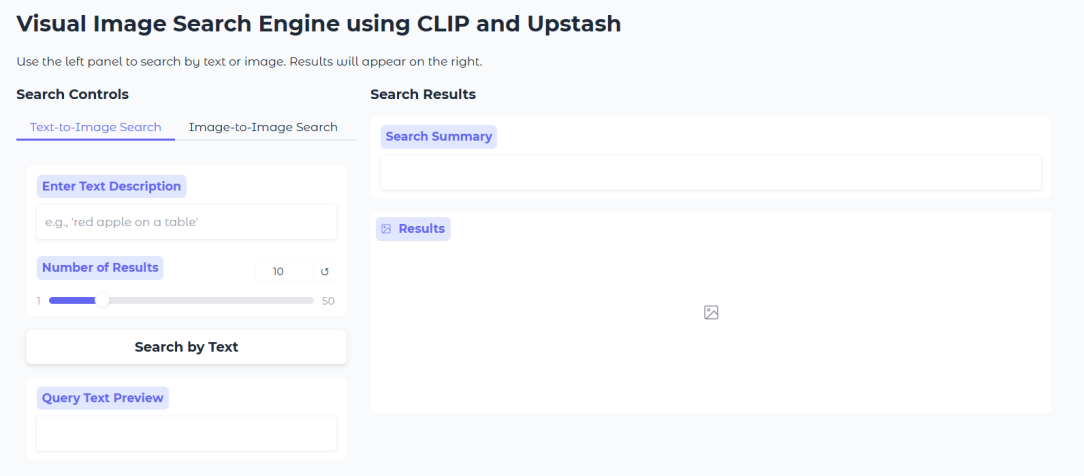
1. **Formulation**: This stage is supported through a thoughtfully designed query construction interface.
    - For *text-to-image* search: Users can enter detailed text descriptions in a generous input area. A real-time "Query Text Preview" confirms their input before execution, helping to clarify search intent. (See Fig: 2)
    - For *image-to-image* search: Users can upload query images from multiple sources (local files, clipboard, camera). An "Upload Query Image" preview is displayed, ensuring the correct image is selected. (See Fig: 3)
    - Intuitive radio buttons allow users to easily switch between text and image-based search modalities.
2. **Initiation of Action**:
    - A prominently positioned "Search by Text" or an implicit search initiation upon image upload, with clear visual hierarchy, makes the path to action obvious.
    - During search execution, visual feedback indicators (e.g., loading messages, status updates) keep users informed about processing status, eliminating uncertainty about whether the system is responding.
3. **Review of Results**:
    - This stage is enhanced through a carefully organized gallery display that presents search results in a visually coherent manner.
    - A "Search Summary" area provides contextual information, such as "Found X unique images for query: 'Y' (Requested up to Z)," confirming the query and number of results.
    - Each image is displayed with its similarity score, allowing users to gauge relevance.
4. **Refinement**: The interface offers tools for users to refine their search if the initial results are unsatisfactory.
    - Users can adjust the "Number of Results" using a slider, providing precise control over information density.
    - If users are unsatisfied, they can easily modify their original query (either text or by uploading a new image) and resubmit the search. The interface maintains the context of their previous query parameters where appropriate, facilitating iterative refinement.
5. **Use**: This stage is facilitated through integrated download functionality.
    - Users can select any result image and download it with a simple click. (See Fig:4) This allows them to save and utilize discovered content directly.

This comprehensive implementation allows users to freely navigate between these stages, repeating them as needed—modifying their queries and refining results—until they find exactly what they're looking for, satisfying their information needs.

3.2 Impact of Input Modalities and Ensuring User-Friendliness

The system supports both **text-to-image** and **image-to-image** search search, which inherently have different initial user operation flows but are designed to converge into a consistent experience.

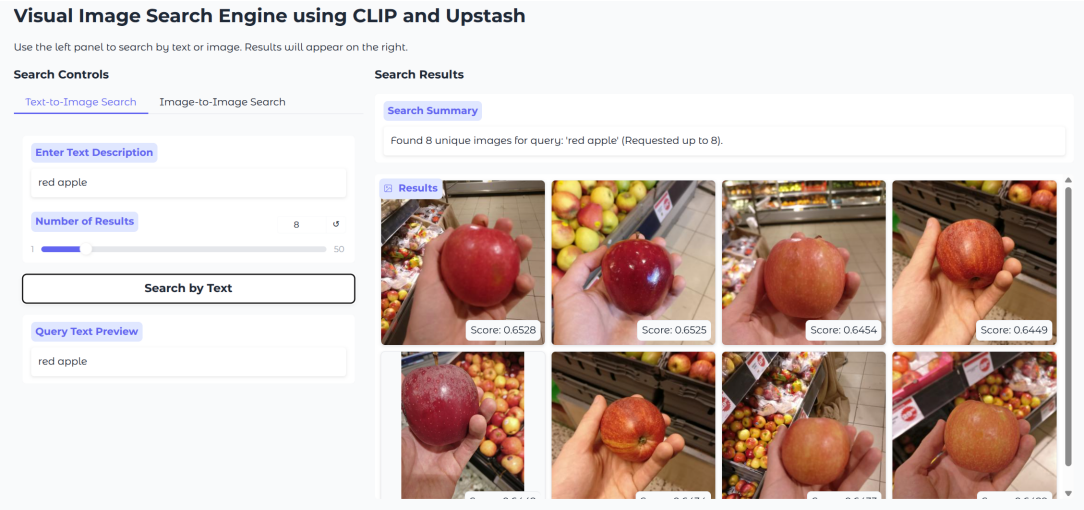
Figure 1: Main interface screenshot



3.2.1 Text Search Considerations

The text search interface offers a thoughtful design focused on natural language input. Users benefit from generous input areas that accommodate detailed descriptive queries of any length. The system provides immediate query previews, allowing users to confirm their search intent before execution. One of the interface's strengths is its ability to handle natural language variations, meaning users can describe images in everyday terms rather than requiring technical vocabulary or specific syntax. This flexibility makes the system approachable for users of all technical backgrounds.

Figure 2: Text search results example

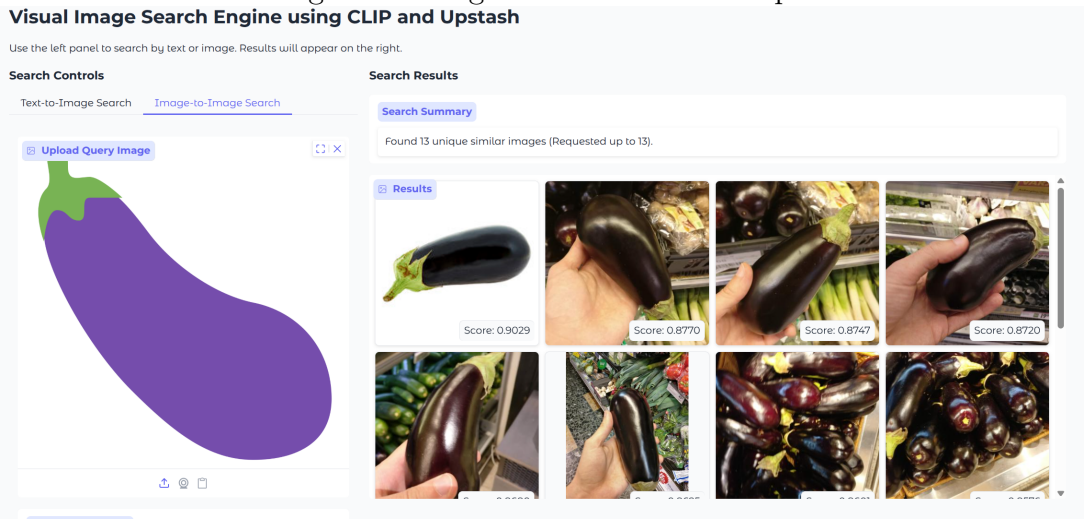


3.2.2 Image Search Considerations

The image search functionality delivers a seamless user experience with a focus on accessibility and feedback. Users can upload query images from multiple sources, including local files, clipboard content, or camera captures. Upon upload, the interface displays a clear preview of the selected image, giving users confidence that the correct image is being used for searching. Behind the scenes, sophisticated image processing occurs

transparently, handling various image formats and sizes without requiring user intervention. Throughout the process, the system provides ongoing status feedback, keeping users informed about upload progress and any potential issues that might require attention.

Figure 3: Image search results example



3.2.3 Unifying the Experience

Creating a consistent experience across both text and image search modalities was a key design priority. We achieved this uniformity through several interconnected approaches.

First, both search types provide query previews that help users understand what they're searching for before executing the query. The results presentation maintains identical formatting and structure regardless of query type, creating a familiar environment for users switching between modalities. All refinement options work consistently across both search types, allowing users to filter and adjust results with the same controls whether they started with text or an image. Perhaps most importantly, both search types follow the same logical flow pattern—formulate, initiate, review, refine, and use—ensuring that the mental model users develop with one search type transfers seamlessly to the other.

3.3 Enhanced User Experience Features

3.3.1 Image Download Functionality

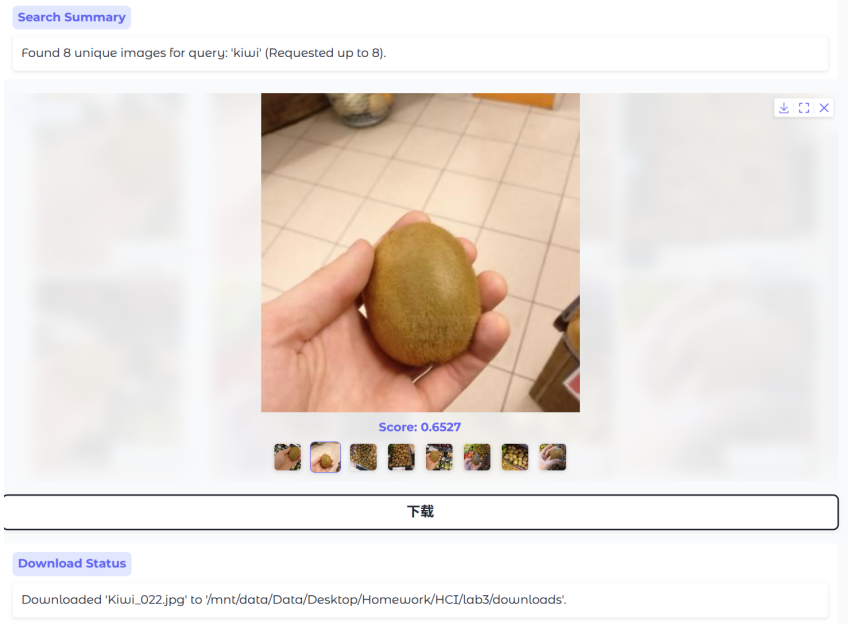
The application implements a comprehensive download mechanism that significantly enhances the user interaction experience by directly supporting the "Use" stage of the Five-Stage Search Framework and providing tangible value beyond just finding images.

- **Seamless Access to Content:** When users identify valuable images, they can download any result with a simple click. This transforms the search from a purely exploratory activity into one with a practical outcome.
- **Easy Organization:** The system automatically stores downloaded images in a dedicated "downloads" directory, maintaining original filenames where possible for easy identification and organization post-search.
- **Clear Feedback:** Throughout the download process, clear status messages (e.g., "Downloaded 'image.jpg' to 'path/downloads'") keep users informed about success or failure, eliminating uncertainty.

- **Reduced Interface Clutter:** The download button is designed to appear intelligently, typically only when an image is selected or hovered over, reducing visual clutter during general browsing of results.

This capability completes the search cycle, allowing users not just to find relevant content but also to easily acquire and utilize it, thereby making the entire interaction more purposeful and satisfying.

Figure 4: Download functionality demonstration



4 Implementation Challenges and Solutions

4.1 Vector Search Optimization

To improve search performance, especially for larger datasets, several optimizations were implemented to balance accuracy with response time.

The system employs a query oversample and filter strategy that requests more results than needed from the vector database and then filters them to ensure sufficient valid results. This approach compensates for potentially invalid or inaccessible images in the database without requiring multiple round-trip queries.

Path resolution caching was implemented to significantly reduce computational overhead, particularly during batch operations. By maintaining a cache of previously resolved file paths, the system avoids redundant file system operations when processing multiple results with common base paths.

Additionally, image embedding generation is performed in batches rather than individually, reducing memory pressure and leveraging hardware acceleration more effectively. This batched processing approach allows the system to scale more gracefully as the dataset size increases, maintaining responsive performance even with thousands of images.

4.2 User Interface Responsiveness

To ensure the interface remains responsive even during complex operations, several technical approaches were implemented to create a fluid user experience.

Heavy computation tasks, including embedding generation and similarity searches, are performed asynchronously to prevent UI freezing. This separation allows the interface to remain interactive while resource-intensive operations continue in the background. The results are displayed progressively as they become available, allowing users to begin reviewing initial matches while additional results are still being processed and ranked. This approach feels much more responsive than waiting for complete result sets, especially with larger queries.

Throughout all operations, clear status messages keep users informed about ongoing processes, completed stages, and any issues that might require attention. This transparent communication reduces user frustration by eliminating uncertainty about system state and progress.

5 Conclusion

The CLIP Image-Text Search System demonstrates how modern AI models can create intuitive bridges between language and visual content. By explicitly designing the user interface around the Five-Stage Search Framework and focusing on user experience details such as consistent interactions across different input modalities and practical "Use" features like image downloading, the system provides a robust and user-friendly tool for exploring image collections.

The enhancement to the basic search functionality through the download feature showcases how targeted interface improvements can significantly enhance the overall user experience. This feature directly supports the "Use" phase of the search framework.

Future work could expand on this foundation by implementing more advanced filtering options, supporting collections of downloaded images, and introducing personalized result ranking based on user preferences.