Visual Search Engine using Vision-Language Models (VLMs)

Overview

We developed a visual search engine that uses vision-language models (VLMs) to retrieve relevant images based on text or sample image queries. The system uses OpenCLIP to embed both text and images into a shared space, enabling intuitive search capabilities. The core components include OpenCLIP for embedding, FAISS for efficient indexing and similarity search, and Streamlit for the user interface.

Objectives

1. Shared Embedding Space

Utilized OpenCLIP to generate embeddings for both text and image data.

Ensured semantically similar text and images align in the shared space.

2. Indexing & Retrieval

Used FAISS to index embeddings and perform efficient nearest neighbor searches.

Applied cosine similarity to retrieve top-matching results.

3. Multi-Modal Querying

Supported text-based queries (e.g., "a mountain with snow") and image-based queries.

Considered implementing advanced queries (e.g., compositional prompts) as a future enhancement.

4. Evaluation & Metrics

Performed qualitative assessments to ensure retrieved results matched query intent.

Focused on accuracy and diversity of returned images.

Prerequisites

Familiarity with Python, PyTorch, and libraries like NumPy, Pandas.

Understanding of computer vision and NLP fundamentals.

Basic knowledge of indexing structures and Streamlit for frontend development.

Implementation Details

Model: Used OpenCLIP (a pre-trained VLM) for generating embeddings.

Indexing: Leveraged FAISS for scalable vector similarity search.

UI: Developed a user interface using Streamlit to enable interactive querying.

Dataset: Collected image data from multiple websites, filtered and categorized into relevant groups.

Challenges

1. Data Collection

Sourcing diverse and balanced datasets across categories.

2. Similarity Search Tuning

Selecting and fine-tuning the most effective similarity metric (cosine similarity).

3. Performance

Ensuring low latency search results and smooth UI responsiveness.

Future Enhancements

Add pagination (e.g., search result pages 1, 2, 3...) for improved navigation.

Expand dataset categories for broader query coverage.

Incorporate user feedback to refine search results

Team Contribution

Amandeep singh

Arshdeep kaur

Keshav

Expected Outcome

A fully functional visual search engine capable of returning semantically similar images for text or image-based queries.

Implementation Tips

Choose the Right Model: OpenCLIP worked well out of the box. Domain-specific tuning can improve performance.

Efficient Indexing: FAISS enabled fast and scalable similarity search.

Iterative Approach: Start small and scale after validating initial results.

User Feedback: Use real-world feedback to enhance retrieval quality and user experience.