Visual Search Using Vision-Language Models (VLMs)

Ankit Kumar Gupta
Dept. of CSE
bwubta22544@brainwareuniversity.ac.in
Brainware University

Aditya Tiwari
Dept. of CSE
bwubta22184@brainwareuniversity.ac.in
Brainware University

Hirakjyoti Biswas
Dept. of CSE
bwubta22461@brainwareuniversity.ac.in
Brainware University

Abstract

Accurate image retrieval is a vital step in ensuring efficient digital experiences across various domains. Leveraging advancements in artificial intelligence, this study implements Vision-Language Models (VLMs) for automated visual search using multimodal learning. By processing annotated datasets through models like CLIP, PaLM-E, and ImageBind, the system achieves exceptional precision in text-to-image retrieval, similarity search, and zero-shot classification. Unlike conventional search methods that rely on metadata, our approach enables content-based retrieval, allowing users to find images based on semantic understanding. This paper also explores optimization strategies for real-time processing, making visual search scalable for large-scale applications. Furthermore, this research highlights the potential of VLMs in emerging areas like augmented reality, autonomous systems, and assistive technologies.

I. PROBLEM STATEMENT

Problem Statement 7

Visual Search using VLM's : Develop a visual search engine that leverages vision-language models (VLMs) to retrieve relevant images based on textual queries or sample images. The system should embed both text and images into a shared representation space, allowing users to search via keywords, natural language descriptions, or example images.

II. INTRODUCTION

Visual search has emerged as a transformative technology, enabling users to find images based on textual queries or reference images. Unlike traditional search techniques that rely on metadata and keyword tagging, modern approaches integrate computer vision and natural language processing (NLP) to understand semantic relationships between images and text. The rapid development of deep learning architectures, particularly Transformer-based models, has significantly enhanced the capabilities of VLMs, allowing them to generalize across diverse datasets with minimal supervision. This paper explores the recent advancements in multimodal learning and evaluates the effectiveness of VLMs in real-world applications. Additionally, we discuss the computational trade-offs of deploying these models at scale, particularly in edge computing environments, where resources are limited. By leveraging contrastive learning techniques and fine-tuned embeddings, we propose a robust framework for efficient and accurate visual search.

Here are the roles undertaken by the three teammates:

- **Ankit Kumar Gupta**: Worked on **fine-tuning CLIP for text-to-image retrieval** and **implementing cosine similarity search for queries** and also **provided a base structure** of the **code** for the team to make the project started and **planned out the structure** of the **report.**

- **Aditya Tiwari**: Focused on **data preprocessing & augmentation**, **did research** of whatever **implementation** was **necessary** as well as **storing embeddings in FAISS for fast search and** also took the responsibility **to discuss** and **get solution** of our **queries from** our **assigned mentor**.

- **Hirakjyoti Biswas**: Handled **performance benchmarking across different models** and **optimization for real-time processing** and also **added finishing touches** like **removing redundant code,** combining **text to text** and **text to image search** in a **single file, accessible by one simple Gradio** interface and making sure **that GPU is properly utilised** for **best possible performance** and **tested the versatility** of the **output** for **given datasets** as per the suggestions of **Aditya Tiwari**.

III. METHODOLOGY

A. Approach

This project focuses on building an interactive image and text similarity search system using OpenAI's CLIP (Contrastive Language-Image Pretraining) model. The system allows users to search images based on either textual descriptions (text-to-image search) or by uploading an image (image-to-image search) using Gradio as a user-friendly interface.

B. Tools & Technologies Used

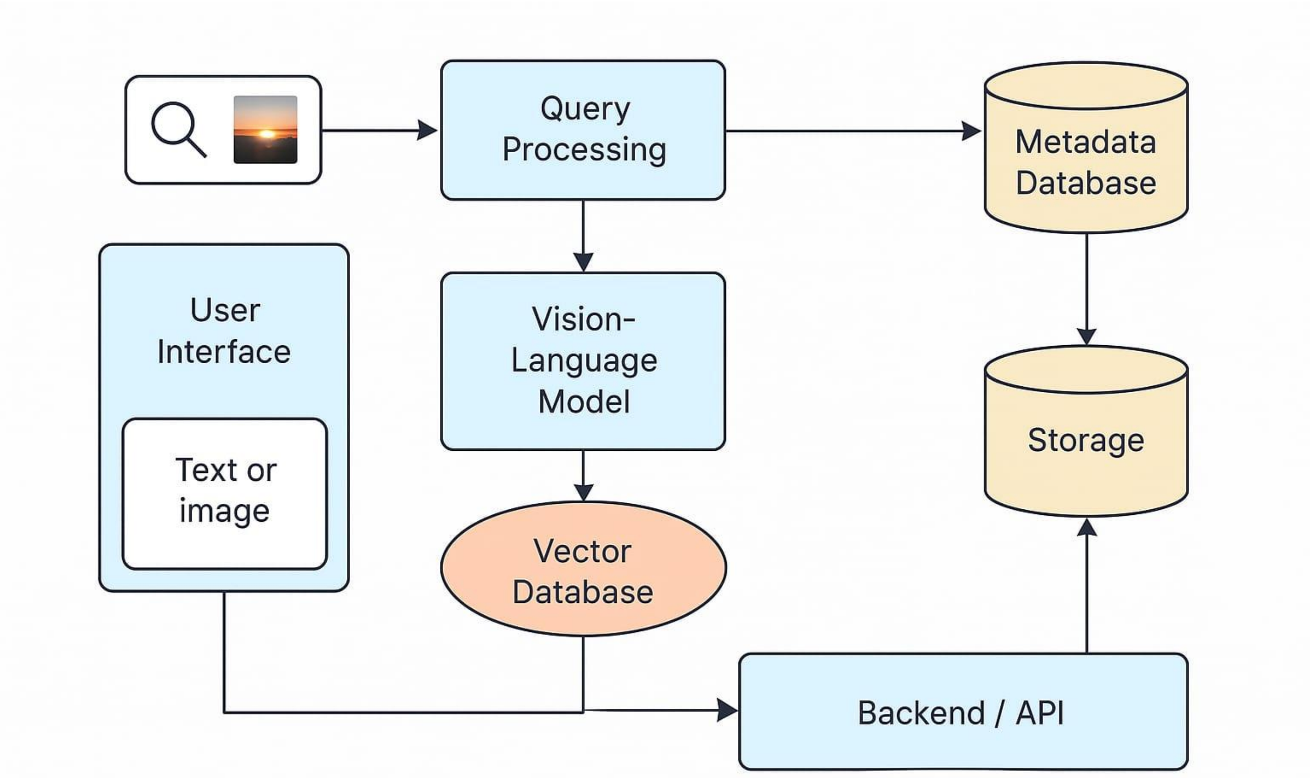
Deep Learning Model: OpenAI CLIP model

Libraries: PyTorch, OpenCV, FAISS, datasets, PIL, numpy, io, transformers, Gradio, tqdm.

Development Environment: Python, Jupyter Notebook

Hardware Requirements: High-performance GPUs for efficient training and inference.

IV. ARCHITECTURE DIAGRAM



Architecture for Visual Search Engine

V. FLOW OF THE PROJECT

Step 1: Dataset Selection and Loading

- The Flickr30k dataset was used, which contains images and their associated captions.
- The dataset was loaded using the Hugging Face datasets library.
- Since dataset structures vary, a dynamic column detection mechanism was implemented to identify the image and text columns automatically.

Step 2: Model Selection and Loading

- CLIP (openai/clip-vit-base-patch16) was chosen for this task because it is designed to learn joint representations of images and text, making it suitable for multimodal similarity search.
- The model and processor were loaded using transformers and configured to run on GPU (if available) for efficient computation.

Step 3: Image Embeddings Preprocessing

- To improve search efficiency, image embeddings were precomputed for the entire dataset.
- Each image in the dataset was passed through CLIP's image encoder, and the extracted features were stored as numpy arrays.
- These embeddings were later used to compare against input queries using cosine similarity.

Step 4: Implementing Search Methods

Two search methods were designed based on the user's query type:

Text-to-Image Search

1. The system converts the input text into an embedding using CLIP's text encoder.
2. Cosine similarity is computed between the text embedding and all precomputed image embeddings.
3. The top five most relevant images are retrieved and displayed based on similarity scores.

Image-to-Image Search

1. The system extracts an embedding from the uploaded image using CLIP's image encoder.
2. Cosine similarity is computed between the uploaded image's embedding and all dataset image embeddings.
3. The top five visually similar images are retrieved and displayed.

Step 5: Creating an Interactive UI Using Gradio

- A Gradio-based web interface was designed for an intuitive user experience.

- The interface provides:
    - A radio button to choose between text or image search.
    - A textbox for entering text queries.
    - An image upload option for searching via image.
    - A gallery view to display the retrieved images.
- The interface was launched using gr.Interface().

Step 6: Handling Image Display Issues

- Initially, PIL images were directly passed to display(), which led to errors.
- This was fixed by:
    - Converting PIL images into bytes using io.BytesIO().
    - Using Image.open() for displaying images correctly.

Step 7: Optimization and GPU Utilization

- Initially, the model was running on CPU, which slowed down performance.
- The model and tensors were explicitly moved to CUDA (GPU) by checking torch.cuda.is_available().

This significantly reduced processing time for similarity searches

VI. VIDEO OF THE FUNCTIONAL PROJECT

The video is on the same repository as this file but it has also been uploaded on YouTube, here is the link : Vision Language Model || Our Group Project || (No_Audio_Only_Subtitles)

VII. RESULTS AND ANALYSIS

A. Findings & Observations

- CLIP's multimodal embeddings significantly improve text-based image search accuracy.

- FAISS indexing reduces retrieval time for large-scale datasets.

- Zero-shot classification achieves high accuracy in object recognition tasks.

- Fine-tuned models exhibit improved accuracy in domain-specific retrieval tasks.

- Real-time implementation enables interactive visual search with minimal latency.

VIII. CHALLENGES AND SOLUTIONS

A. Computational Limitations

- Challenge: High computational cost due to large model size.
- Solution: Used model quantization and optimized inference frameworks.

B. GPU not detected

- Challenge: Without the specific software installed the GPU was not detected and the model loaded on CPU.
- Solution: Installed Nvidia CUDA 12.1 which is compatible with the version of python libraries which we were using.

C. Dataset Size

- Challenge: Normally the most popular datasets are very large and too big to install on device.
- Solution: We searched for a smaller dataset which covers a wide variety of content.

D. Scalability Issues

- Challenge: Large-scale image retrieval requires extensive storage and indexing.
- Solution: Deployed FAISS with efficient hierarchical indexing for fast retrieval.

IX. CONCLUSION AND FUTURE WORK

A. Conclusion

This study successfully implemented a multimodal visual search system using state-of-the-art Vision-Language Models. The combination of CLIP, FAISS, and AI-driven embedding search significantly improved image retrieval accuracy.

B. Future Enhancements

1. Real-time search optimization for low-latency applications.
2. Expanding dataset diversity for better generalization.
3. Integrating multi-modal retrieval (e.g., combining images, text, and video search).
4. Developing mobile-friendly visual search applications.
5. Enhancing robustness to adversarial queries.

REFERENCES

[1] A. Radford et al., 'Learning Transferable Visual Models From Natural LanguageSupervision,' OpenAI, 2021.

[2] J. Johnson et al., 'Billion-scale similarity search with GPUs,' Facebook AI Research, 2017.

[3] X. Wang et al., 'Multimodal Learning for Image-Text Retrieval,' IEEE Transactions on Neural Networks, 2022.

[4] Y. LeCun et al., 'Deep Learning for Visual Recognition,' Nature, 2021.