20XD96: Information Retrieval and Web Search Lab

Closet Companion: Your Personalized Outfit Finder

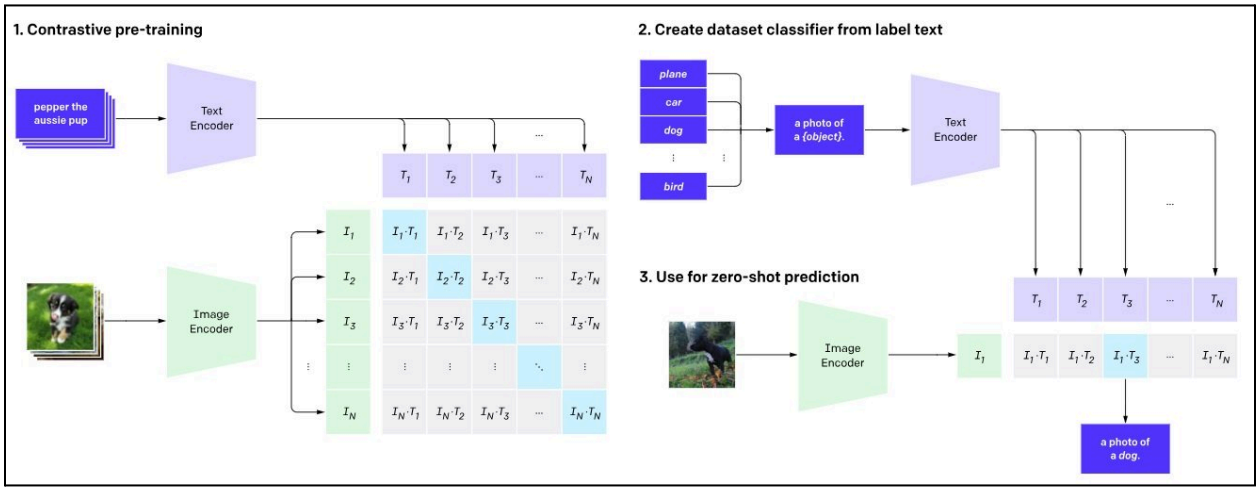
20PD02 - Aditya
20PD11 - Kartheepan
20PD13 - Kesavan

Introduction

In the era of digital fashion and personalized shopping, efficient and accurate image retrieval systems are essential to create an interactive and personalized user experience. This project explores a retrieval-based approach where users can input either a text description or an image, and the system responds with fashion items that are visually or semantically similar. The foundation of this retrieval model is the CLIP (Contrastive Language–Image Pretraining) model, which has been fine-tuned on a fashion-specific image dataset. This application has been developed to demonstrate the potential of neural-based retrieval models for personalized recommendations and shopping.

Contrastive Language-Image Pretraining (CLIP)

OpenAI's CLIP model combines image and text encoders to align image features with semantic text features in a shared embedding space, making it ideal for cross-modal retrieval tasks. Unlike traditional vision models that are solely image-based, CLIP is inherently designed to handle both image and text inputs, providing flexibility and robustness to diverse query types. Additionally, with advancements in transfer learning and fine-tuning, CLIP has shown exceptional performance in domain-specific tasks when fine-tuned with specialized datasets. Given the complex nature of fashion, where both visual details and textual descriptions hold significance, CLIP was selected for its multimodal capabilities and its ease of adaptation through fine-tuning.



Features and Working of CLIP

- **Dual-Encoder Architecture**: CLIP uses two distinct encoders for text and images, enabling it to process and embed them in a shared latent space.
- **Contrastive Pretraining**: The model is pre-trained on pairs of images and text, using contrastive learning to align text and image representations.
- **Unified Embedding Space**: CLIP maps both images and text into the same vector space, allowing for the comparison between text-image pairs based on cosine similarity.
- **Cross-Modal Retrieval**: The shared embedding enables flexible retrieval, allowing for either text-to-image or image-to-image searches.

The above features make CLIP especially suitable for applications where users can input different types of queries and still obtain relevant results, making it an optimal choice for fashion retrieval.

Fine-Tuning the CLIP Model on Fashion Product Images (Small) Dataset

- **Dataset Preparation**: A curated fashion image dataset was collected, containing images and descriptions of around 44000 fashion items across categories, colors, and styles. Each image was paired with a text description to facilitate the alignment of image-text features.

- **Fine-Tuning Process**:
    1. **Dataset Formatting**: Images and text were preprocessed according to CLIP's requirements. Images were resized, normalized, and formatted to match CLIP's input specifications.
    2. **Embedding Initialization**: The pre-trained weights from OpenAI's CLIP vit b 32 model were used as the starting point.
    3. **Training Objective**: Fine-tuning was performed using a contrastive loss function that helps further align text and image embeddings based on the fashion dataset.
    4. **Batching and Epochs**: The model was trained for 10 epochs with batch sizes of 8 for available GPU resources, ensuring efficient convergence without overfitting.

- **Result of Fine-Tuning**: Fine-tuning CLIP on the fashion dataset refined its capability to capture subtle nuances in fashion, such as color shades, styles, and categories. This process improved the model's ability to respond accurately to both text and image queries in the fashion domain. The final loss was 10e-3.

Using the Fine Tuned Model for Image Retrieval

- **Text Query-Based Retrieval**: For text queries, the user enters a description (e.g., "red polo shirt"), which is encoded by CLIP's text encoder into a query embedding. The system then calculates cosine similarities between this embedding and the image embeddings, retrieving the top K images that are most similar.

- **Image Query-Based Retrieval**: For image queries, the user uploads an image, which is passed through CLIP's image encoder to generate an embedding. Similar images are identified by finding image embeddings that are closest to this query image embedding.

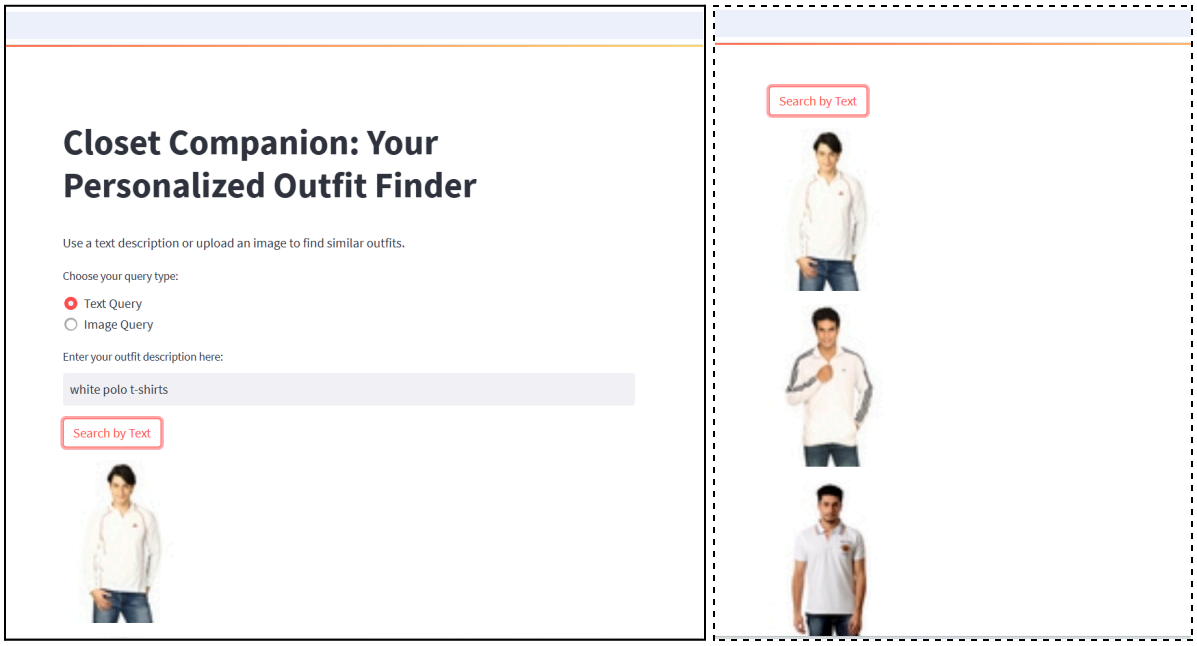
- **Cosine Similarity Scoring**: Both query types leverage cosine similarity to rank retrieved images. By retrieving the closest matches, the system ensures the most relevant items are shown, ranked from most to least similar.

These functionalities were implemented in a Streamlit app, providing users with an interactive interface to search for fashion images by either text descriptions or image examples.

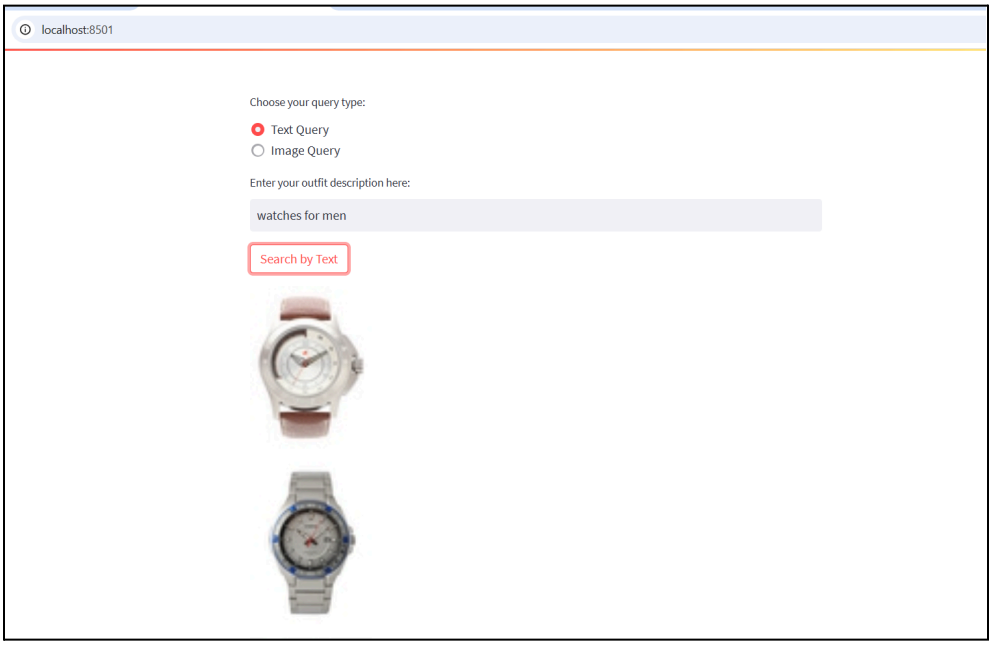
Results

- **Text Query Examples**:

Query: "white polo t-shirts"



Query: “watches for men”



- **Image Query Examples**:

  Image: Related to running shoes

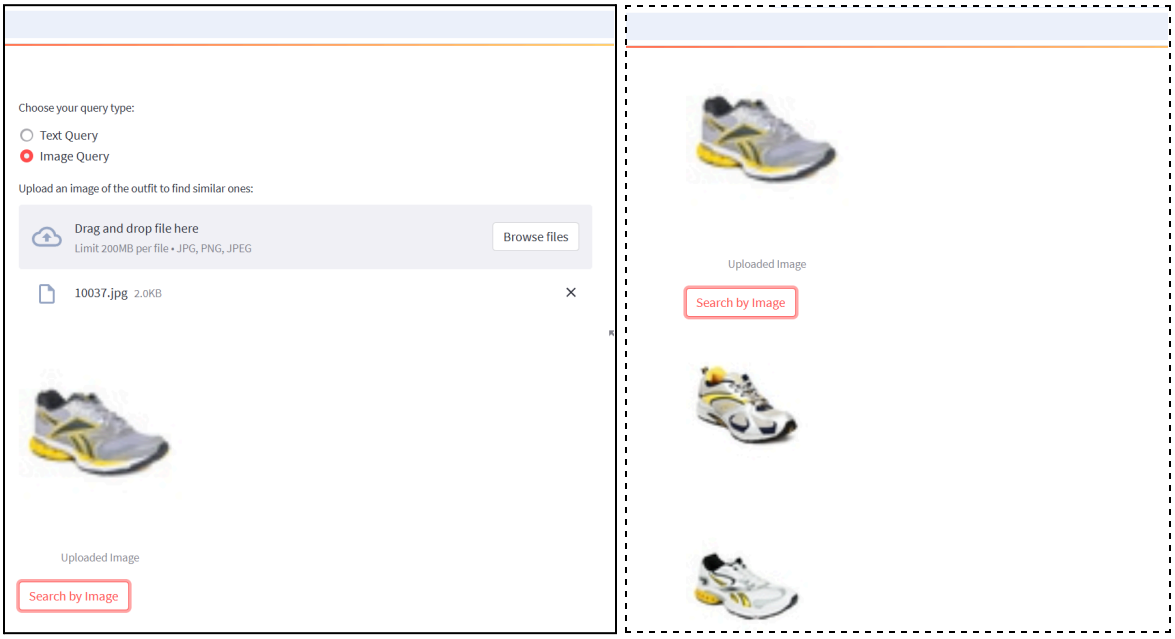
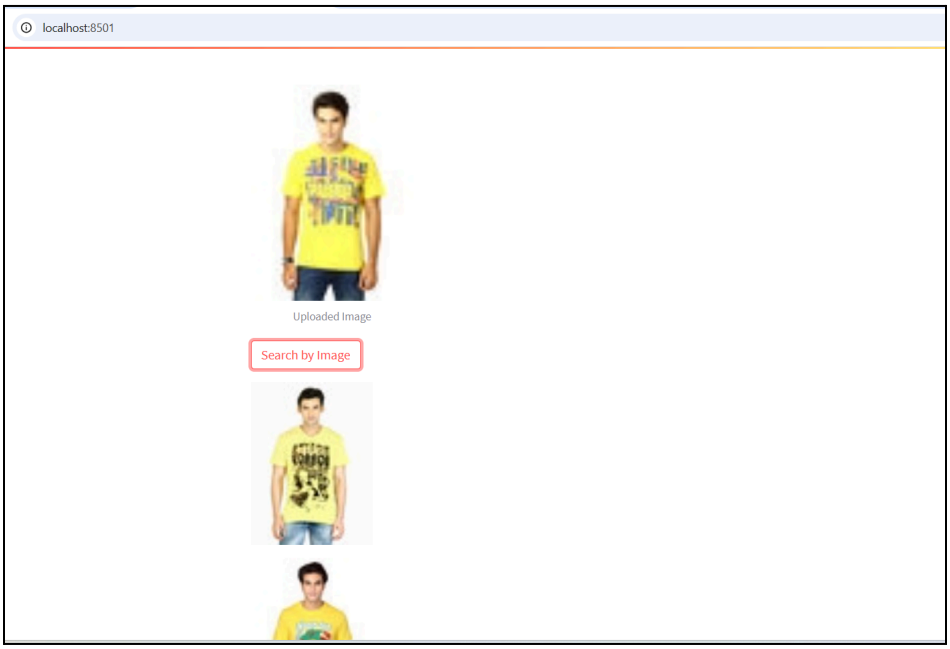


Image: Related to yellow t-shirts



Across both query types, the fine-tuned CLIP model demonstrated high retrieval relevance, with a qualitative assessment showing that the retrieved items matched user expectations closely.

Conclusion

The project effectively showcases the application of a fine tuned CLIP model for cross-modal retrieval in the fashion domain. By leveraging CLIP's powerful multimodal embeddings and aligning them through fine-tuning, the model demonstrates strong retrieval accuracy for both text and image queries. This approach highlights the potential of neural-based retrieval systems for enhancing personalized shopping experiences and digital fashion recommendations. Future work could focus on further refining the dataset, incorporating broader categories, and integrating user feedback to continuously enhance the relevance of retrieval.

Future Scope

- **Enhanced Dataset**: Expanding the dataset to include a wider variety of fashion styles, demographics, and brands can improve retrieval diversity.
- **User Feedback Integration**: Including user feedback in model tuning can help improve retrieval personalization, adapting to individual preferences.
- **Real-World Deployment**: Integrating the app with e-commerce platforms can enable personalized shopping and recommendation experiences.