

AI-Powered Image Retrieval System

This project implements a content-based image retrieval system using deep learning and vector similarity search. The system allows users to upload an image and find visually similar images from a pre-indexed collection.

Features

- Upload any image and find visually similar images
- Uses pre-trained ResNet model fine-tuned for image similarity
- Fast search with FAISS vector database
- Interactive web interface built with Streamlit
- Efficient feature extraction and indexing process

Project Structure

```
├── caltech101/           # The caltech dataset used for training
├── data/                 # Directory for storing features and FAISS
index (faiss_index.bin, features_path.json)
├── weights/             # Directory for storing trained model weights
(model.pth)
├── src/                  # Source code for the model
│   ├── __init__.py
│   └── model.py          # Model definition
├── utils/                # Utility functions
│   ├── __init__.py
│   ├── data_utils.py    # Data loading utilities
│   ├── display_utils.py # Display functions for Streamlit
│   ├── faiss_utils.py   # FAISS indexing utilities
│   ├── image_utils.py   # Image processing utilities
│   └── precompute_features.py # Script to precompute image features
├── app.py                # Main Streamlit application
└── precompute.sh         # Script to precompute features
```

Usage

Step 1: Prepare your model

Place your trained model in the `weights/` directory as `model.pth`. The model should be a saved PyTorch state dictionary compatible with the `ResNetTransferModel` class defined in `src/model.py`. Change `src/model.py` if you plan to use a different model definition.

Step 2: Precompute features

Run the precompute script to extract features from your dataset and build the FAISS index:

```
sh precompute.sh
```

You can customize the number of images per class:

```
sh precompute.sh --num-per-class 20
```

This will extract features from your dataset, build a FAISS index, and save everything to the **data/** directory.

Step 3: Run the web app

Launch the Streamlit web application:

```
streamlit run app.py
```

This will start the web server and open the application in your default browser. From there, you can:

1. Upload an image
2. Click "Search" to find similar images
3. View the results with similarity scores

App Features

Customizing Search Results

The sidebar includes several options to customize your search experience:

- **Number of Results:** Use the slider to adjust how many similar images to display (1-10)
- **Show All Categories:** Enable this checkbox to display all available image categories in the sidebar. This helps you understand what types of images are in the database.

The application will display the uploaded image in the sidebar and show the retrieved similar images with their categories and similarity scores in the main panel.

Settings

Custom Paths

FAISS Index Path

data/faiss_index.bin

Features Paths File

data/features_paths.json

Number of Results

1

7


10

☐ Show All Categories

Using device: cpu

FAISS index contains 2020 images

Uploaded Image



Deploy

AI Powered Image Retrieval Demo

Find Similar Images using Vector Databases

Upload an Image

Choose an image...

Search

Clear

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Dalmatian.jpg

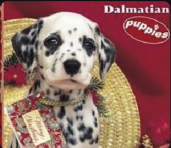
52.5KB

×

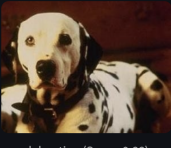
Model loaded successfully!

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.


Retrieved Similar Images




dalmatian (Score: 0.89)




dalmatian (Score: 0.88)




dalmatian (Score: 0.88)




dalmatian (Score: 0.86)



dalmatian (Score: 0.86)





3 / 3