

Find Similar Photos - Discover Images

Introduction

I am excited to present the findings and progress of my project, which involved developing a web-based application called "FindSimilarPhotos."

This project aimed to provide users with a novel way to rediscover their image collections, leveraging a combination of advanced technologies and user-friendly design.

"FindSimilarPhotos" is an innovative web application that empowers users to explore the visual connections within their image libraries.

This project was undertaken with a strong focus on simplicity and effectiveness, enabling users to uncover hidden relationships among their images.

Technologies

- **Flask:** Serving as the backbone of my application, Flask facilitated smooth and efficient interactions between the user interface and the backend processing.
- **Keras:** The utilization of Keras, a high-level neural networks API, allowed me to employ deep learning techniques to analyze, extract and identify visual similarities among images.
- **HTML, CSS, and JavaScript:** These front-end technologies collectively shaped the user interface, delivering an intuitive and engaging platform for users to upload, explore, and enjoy their image collections.

Setup Instructions

Clone the Repository (No Technical Steps Required):

- You can access the "Find Similar Photos" application by simply cloning it from GitHub.

Upload Your Photos

- On the website's homepage, you'll find a user-friendly interface. Click the "Choose File" button to select and upload your photos.
- You can upload whatever you want.

Discover Similar Photos

- After choosing your photo, click the "Upload" button.
- The application will analyze your images and quickly provide you with visually similar photos from the database.

Explore Your Collection

- Browse through the results and explore visually similar photos.

Data Collection

For the "Find Similar Photos" application, I've gathered a diverse dataset of over 10,000 images.

These images include various categories, including clothing, animals, fruits, vehicles, and more. We collected this data from Kaggle.

API Usage

My application offers an API endpoint that allows you to find visually similar photos based on an input image.

How to Use the API

- **Make a POST Request:** To use the API, you need to send a POST request.

Include Your Image

- In your request, include your image as a base64-encoded string.

Receive Similar Photos

- Upon receiving your request, the server will analyze the image and return up to 9 visually similar photos.

Speed Optimization

- The application uses a dictionary for efficient image comparison, resulting in faster response times, this dictionary saves all paths and features of the 10k image data.

Limitations

- Please note that the API returns a maximum of 9 similar photos to keep responses concise.

Helper Class

The "Helper" class plays a crucial role in the "Find Similar Photos" application, assisting in the image comparison and similarity calculation process.

Purpose:

- **Image Preprocessing**

One of the primary functions of the Helper class is to preprocess the input images. It prepares the image data to meet the model's expectations by resizing and normalizing it.

- **Feature Extraction**

The class is responsible for extracting important features from the images. These features are used to quantify the content of each image, making it easier to compare them.

- **Cosine Similarity Calculation**

The Helper class calculates the cosine similarity between images. This metric quantifies the similarity between two images' feature vectors, enabling the system to identify visually similar photos effectively.

- **Data Storage**

The class maintains a dictionary that maps image paths to their corresponding feature vectors. This data structure significantly speeds up the image comparison process.

- **Loading the Pretrained Model**

The model initialization process involves loading a pretrained deep learning model. This model has been trained on a large dataset to recognize and extract intricate features from images.

Model Selection

I tested multiple models, including Xception, ResNet, VGG16, and Inception ResNet V2, to select the best one. After careful evaluation,

I chose the model that performed the best for our image similarity task which is Inception ResNet V2.

