# **Database outlineIMAGERETRIEVAL SYSTEM**

Backend (search or search\_voice)

Frontend Displays Images

Images

Query

Image Embeddings

Converts an input image into a **high-dimensional vector embedding**.

Uses **CLIP's vision model** to extract **semantic features** from the image.

Converts the input **text query** into a **feature vector**.

Normalizes the vector to improve **cosine similarity calculations**.

flask Receives & Processes Request

Finding the Most Similar Images

Iterates over all **images in the dataset**.

Computes **cosine similarity** between text and image embeddings.

Returns **top k most similar images**

text

Voice

Front end (User Inputs Query)

Flask Sends Response

# Image Retrieval System Documentation

## Overview

The Image Retrieval System is a Flask-based web application that allows users to search for images using text queries or voice input. It utilizes OpenAI's CLIP model to compute embeddings for images and text, enabling semantic search functionality.

## Features

• Text-Based Image Search: Users can enter a text query (e.g., 'children playing outside) to retrieve the most similar images from a dataset.

• Voice-Based Image Search: Users can use their microphone to speak a query, which is transcribed and used to retrieve images.

• Semantic Search: The CLIP model computes embeddings for images and text, enabling similarity-based retrieval.

## Technologies Used

• Backend:  
 - Flask (Python web framework)  
 - CLIP (OpenAI's vision-language model)  
 - PyTorch (for model inference)  
 - SpeechRecognition (for voice input)  
 - pyttsx3 (for text-to-speech)

• Frontend:  
 - HTML, CSS, JavaScript

## Usage

**• Text-Based Search:**  
 1. Enter a text query in the search box  
 2. Click Search.  
 3. The system will display the top 5 most similar images.

**• Voice-Based Search:**  
 1. Click the Voice Search button.  
 2. Speak your query into the microphone (e.g., 'show me a beach').  
 3. The system will transcribe your query, retrieve the most similar images, and provide audio feedback.

## Configuration

The application can be configured by modifying the `app/config.py` file. Key configurations include:  
- `IMAGE\_FOLDER`: Path to the folder containing images for retrieval.

## Dependencies

The project requires the following Python packages (listed in `requirements.txt`):

## Results

