Image Search Engine

Supervised by :

M. Tebourbi Riadh

Prepared by :

Barnat Rim

Sahli Molka

Année Universitaire :
2025-2026

Table des matières

Abstract

This report presents the design, implementation, and functionalities of an image search engine that leverages the VGG16 deep learning model for feature extraction, Elasticsearch for efficient indexing and FastAPI for the backend API. The system enables fast and accurate image retrieval based on visual similarity by combining deep learning techniques with optimized data indexing and scalable deployment.

Introduction

In today's digital world, the ability to efficiently search and retrieve images has become increasingly important. Traditional text-based search methods often fail to capture the visual content of images, making it difficult to find truly relevant results. This project addresses this challenge by developing an image search engine that combines deep learning and optimized indexing techniques to provide accurate and fast visual searches.

The system is built around several core components :

- **VGG16-based Feature Extraction :** A pretrained deep learning model is used to extract detailed feature vectors representing the content of images, enabling accurate similarity comparisons.
- **Elasticsearch Indexing :** Efficiently stores image features and supports rapid similarity searches over large image datasets.
- **FastAPI Backend :** Handles search requests, image uploads, and feature computations through a lightweight and high-performance API.
- **Interactive Web Interface :** Allows users to perform searches using keywords or images and displays results in an intuitive and interactive manner.
- **Refined Search Capability :** Users can select any result image to perform a new search, allowing iterative exploration of related visual content.

This report presents the architecture, technology stack, and implementation details of the system, highlighting how deep learning and efficient indexing are combined to build a robust and user-friendly image search platform.

1 Project Architecture

The image search engine is structured to deliver fast and precise retrieval of images based on visual similarity. The system is organized using a client-server model, integrating deep learning, advanced search indexing, and a web-based interface to provide a smooth user experience.

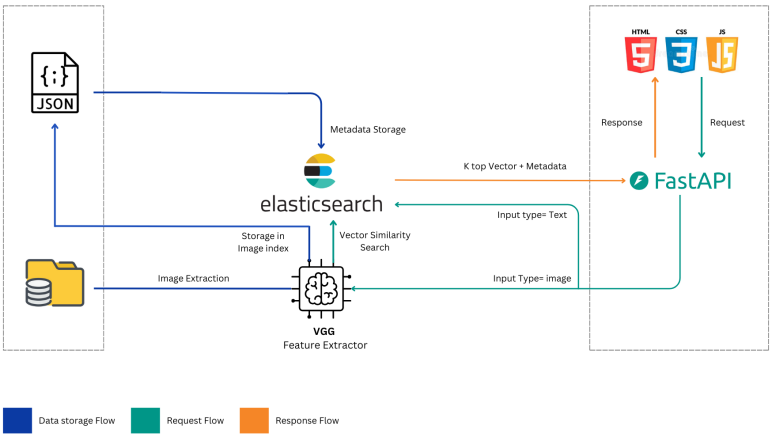

Figure 1 – Project Architecture

The overall architecture of the image search engine is illustrated in this Figure, which presents the interaction between the main system components — the frontend interface, the backend server, the deep learning model for feature extraction, and the Elasticsearch engine for indexing and retrieval. This diagram provides a clear overview of how data flows through the system, from image input to similarity-based search results.

1.1 VGG16 Feature Extraction

The system uses the VGG16 convolutional neural network (CNN), a deep learning model pretrained on large-scale image datasets such as ImageNet, to extract robust feature representations from images.

VGG16 consists of 16 layers, including convolutional layers, pooling layers, and fully connected layers. For this project, we use the outputs of one of the deeper layers (typically the last fully connected layer before classification) as a feature vector. This vector captures the high-level characteristics of an image, including :

-Texture : Patterns, gradients, and repetitive structures in the image.
-Shape : Edges, contours, and geometrical arrangements of objects.
-Color distribution : Overall color patterns and combinations across regions.

Each image in the database is processed through the VGG16 network, producing a fixed-length feature vector. These vectors provide a numerical representation of the visual content of images, allowing the system to measure similarity between images using distance metrics such as cosine similarity or Euclidean distance.

By relying on VGG16, the system benefits from deep hierarchical features learned from millions of images, making the search engine capable of recognizing subtle visual similarities even when images vary in scale, orientation, or lighting conditions.

These feature vectors are then indexed in Elasticsearch, forming the basis for fast and accurate image retrieval, enabling the system to return visually similar results almost instantly.

1.2 Efficient Indexing using Elasticsearch

Elasticsearch serves as the central component for storing, indexing, and querying the feature vectors extracted from images using VGG16. Its distributed and scalable architecture allows the system to handle extremely large datasets efficiently, supporting near real-time similarity searches even when millions of images are indexed. Users can perform searches by uploading a query image, or, when textual metadata such as tags or keywords is available, by combining visual and textual queries to improve search precision.

All image feature vectors and their associated metadata are directly stored and indexed in Elasticsearch, which provides powerful search and aggregation capabilities. This ensures that new images can be seamlessly added to the index without disrupting ongoing queries. The system is therefore able to maintain up-to-date, accurate search results while minimizing latency, enabling fast retrieval of visually or semantically similar images. Additionally, Elasticsearch's rich querying features, such as filtering, ranking, and scoring, allow the system to support complex search requirements, making it highly flexible for a variety of image retrieval scenarios.

1.3 Backend Services with FastAPI

The backend of the image search engine is implemented using FastAPI, a modern, high-performance Python framework designed for building fast and scalable APIs. It handles all server-side operations, including :

- Receiving image uploads and search requests from the client. - Processing query images through VGG16 for feature extraction. - Performing similarity searches on Elasticsearch. - Returning results as JSON responses to the frontend.

Its asynchronous processing capabilities allow the backend to efficiently handle multiple concurrent requests, ensuring low latency and high responsiveness even when many users are performing searches simultaneously.

1.4 Web Client Interface

The frontend of the image search engine is a responsive web application built using HTML, CSS, and JavaScript, providing a lightweight and interactive platform for users. The interface allows users to :
- Upload images for visual similarity searches.
- Enter keywords to perform optional textual searches.
- View retrieved images, which are ranked according to their similarity scores.
- Select any image from the results to initiate a refined search, enabling iterative exploration of visually related content.

Using Vanilla JavaScript, the frontend handles user interactions, dynamically updates the displayed search results, and communicates with the FastAPI backend via asynchronous HTTP requests. The interface is designed for simplicity and usability, ensuring smooth navigation and immediate feedback without relying on additional frameworks, which keeps the system lightweight and easy to maintain.

2 System Workflow

The system workflow describes the step-by-step process of handling user queries and returning relevant images, integrating all components of the architecture. The workflow can be summarized as follows :

1. **User Query Initiation :**
   — The user interacts with the web frontend built with HTML, CSS, and JavaScript.
   — The query can be an uploaded image for visual search or a keyword for optional textual search.
2. **Request Handling by Backend :**
   — The frontend sends the request to the FastAPI backend via asynchronous HTTP requests.
   — If the query is an image, the backend processes it through the VGG16 model to extract a feature vector representing the visual content.
3. **Similarity Search in Elasticsearch :**
   — The extracted feature vector (or textual metadata) is used to query the Elasticsearch index.
   — Elasticsearch computes similarity scores between the query vector and the stored feature vectors and returns the most relevant image URLs ranked by similarity.
4. **Result Delivery to Frontend :**
   — The backend formats the search results in JSON and sends them back to the frontend.

— The frontend dynamically displays the images in a responsive gallery, showing similarity scores and allowing users to interact with the results.

3 Technologies

The image search engine combines deep learning, search indexing, web development, and containerization technologies to provide fast and accurate image retrieval. The main technologies used in this project are :

- **VGG16 :** A deep convolutional neural network used to extract high-level feature vectors from images for similarity-based searches.
- **Elasticsearch :** A distributed search engine that indexes feature vectors and performs fast similarity queries over large datasets.
- **FastAPI :** A high-performance Python framework that handles search requests, processes images, and returns results to the frontend.
- **Frontend (HTML, CSS, JavaScript) :** Provides an interactive and responsive interface for uploading images, entering keywords, and displaying search results.

4 Results

The developed image search engine provides an intuitive and responsive interface that allows users to explore large image datasets efficiently. The web interface is designed to be simple and user-friendly, enabling both visual and textual queries.

When performing an image-based search, the user uploads an image, and the system returns visually similar results based on the extracted deep features. In contrast, the text-based search allows users to find images that match specific keywords or tags from the indexed metadata. Both modes display the retrieved images in an organized gallery, enabling quick visual comparison and relevance evaluation.

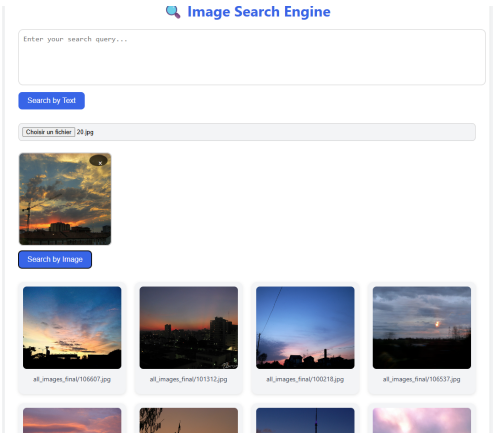

Figure 2 – Image-Based Search

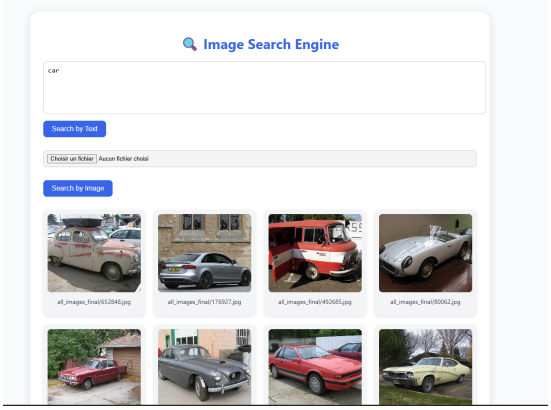

Figure 3 – Text-Based Search

5 Conclusion

This project demonstrates the effectiveness of integrating deep learning techniques, advanced search engines, and modern web frameworks to develop a high-performance image search engine. By leveraging VGG16 for feature extraction, the system is capable of capturing rich and discriminative visual representations of images, which enables accurate similarity comparisons. Elasticsearch provides an efficient and scalable indexing and querying infrastructure, allowing the system to perform near real-time searches even across millions of images.

The platform supports both visual and textual queries, giving users the flexibility to search using images, keywords, or tags, and providing relevant results quickly. Its modular architecture ensures that the system is easily maintainable, extendable, and adaptable to different deployment environments.

Overall, this project highlights how combining deep learning with robust search and retrieval systems can create intelligent, scalable, and user-friendly image retrieval platforms. It also opens the door for further enhancements, such as incorporating more advanced neural network architectures, implementing relevance feedback, or integrating multimodal search capabilities to improve search precision and user experience.