# **VLM Search Engine - Project Documentation**

## **Overview**

VLM Search Engine is a high-quality visual-language multimodal search engine that supports both text-to-image and image-to-image search using state-of-the-art deep learning models. It leverages CLIP (Contrastive Language–Image Pre-training) for embedding generation and FAISS for fast similarity search, enhanced with ESRGAN for image upscaling.

This engine offers a visual search interface via Gradio, making it simple to explore image datasets through natural language or visual input.

## **Architecture Components**

|  |  |
| --- | --- |
| **Component** | **Description** |
| CLIP | Used to generate embeddings for both text and images. |
| FAISS | Indexes and performs fast nearest neighbor search on embeddings. |
| ESRGAN | Enhances image quality before indexing and displaying. |
| STL-10 Dataset | The engine uses a high-quality, preprocessed version of this dataset. |
| Gradio UI | Provides a user-friendly web interface for search queries. |

## **Installation**

Install all necessary dependencies:

1. pip install transformers pillow faiss-cpu numpy torch==2.0.1 torchvision==0.15.2 gradio
2. pip install basicsr realesrgan

## **Dataset Preparation**

The system uses STL-10 images (resized to 256×256 and saved in PNG format) as the dataset:

* Downloads 1000 training images from the STL-10 dataset.
* Saves them as high-quality .png files in stl10\_images/.
* Falls back to synthetic images if download fails.

## **Embedding and Indexing**

CLIP is used to generate 512-dimensional embeddings for each image:

* Each image is resized and optionally upscaled for better visual feature extraction.
* Embeddings are normalized and stored in a FAISS inner-product index.

## **Search Functionality**

You can search using:

* **Text queries**: Descriptive text like "red sports car" or "a dog running".
* **Image queries**: Upload an image and search for visually similar ones.

The search process includes:

1. Embedding the query (text/image).
2. Normalizing and querying the FAISS index.
3. Returning and upscaling top-K matches.

## **Image Enhancement**

Real-ESRGAN (Enhanced Super-Resolution GAN) is used to upscale results for a more polished and clearer output.

Upscaling is applied to both:

* Dataset images during embedding (optional).
* Retrieved results before display.

## **GUI with Gradio**

The engine provides a clean Gradio UI with two tabs:

* **Text Search**: Enter natural language descriptions.
* **Image Search**: Upload an image to search by similarity.

Features:

* Fast response via FAISS.
* Results shown in upscaled quality.
* Examples and input widgets for ease of use.

## **Notable Highlights**

## Multimodal: Text and image inputs supported.

## High-Fidelity: Uses ESRGAN to enhance visuals.

## Fast: Powered by FAISS for efficient search.

## Portable UI: Runs locally or can be shared via Gradio.

## **Project Structure**

.

├── data/

├── stl10\_images/

├── main.py

└── (dependencies installed via pip)

## **Dependencies**

* transformers
* torch, torchvision
* faiss-cpu
* gradio
* Pillow, numpy
* basicsr, realesrgan

## **Future Improvements**

* Add support for custom datasets.
* Allow filtering or refining search results.
* Add logging and search analytics.
* Deploy as a web service with backend support.

**Results**



