

科研项目图片相似性检测引擎系统

用户手册

版本: 1.0.0

日期: 2025 年 9 月 15 日

目录

- [第一章：系统概况](#)
 - [1.1 功能简介](#)
 - [1.2 技术特点](#)
 - [1.3 开发语言与主要技术栈](#)
- [第二章：系统部署](#)
 - [2.1 Windows 部署方法](#)
 - [2.1.1 环境准备](#)
 - [2.1.2 安装依赖](#)
 - [2.1.3 配置环境变量](#)
 - [2.1.4 启动服务](#)
 - [2.2 Docker 部署方法 \(推荐\)](#)
 - [2.2.1 环境准备](#)
 - [2.2.2 配置文件说明](#)
 - [2.2.3 构建与启动](#)
- [第三章：系统使用 \(API 接口\)](#)
 - [3.1 认证](#)
 - [3.2 监控接口](#)
 - [3.2.1 健康检查 \(GET /health\)](#)
 - [3.3 数据入库接口](#)
 - [3.3.1 单张图片入库 \(POST /api/vectors\)](#)

- [3.3.2 批量图片入库 \(POST /api/vectors/batch\)](#)
- [3.3.3 删除数据 \(DELETE /api/vectors/delete\)](#)
- [3.4 数据查询接口](#)
- [3.4.1 向量检索 \(POST /api/vectors/search\)](#)
- [3.4.2 批量向量检索 \(相同过滤器\) \(POST /api/vectors/batch-search-same-filters\)](#)
- [3.4.3 批量向量检索 \(独立过滤器\) \(POST /api/vectors/batch-search-individual-filters\)](#)
- [3.5 管理接口](#)
- [3.5.1 初始化/重建数据库 \(POST /api/admin/initialize-database\)](#)
- [第四章：功能模型 HTML 页面使用方法](#)
- [4.1 访问页面](#)
- [4.2 页面布局](#)
- [4.3 功能使用](#)
- [4.3.1 图片添加](#)
- [4.3.2 图片查询](#)

第一章：系统概况

1.1 功能简介

科研项目图片相似性检测引擎系统（CDS Image Engine）是一个高性能、可扩展的 Web 服务，专为处理和比较图像内容而设计。它适用于科研图表查重、设计素材管理、以图搜图等多种业务场景。

系统的核心功能包括：

- 图片向量化**：将输入的图片（Base64 编码）通过深度学习模型转换为高维度的数学向量，捕捉其核心视觉特征。
- 相似性检索**：接收一张图片，在海量图片库中快速、准确地找出视觉内容上最相似的 N 张图片记录。
- 数据管理**：提供对图片向量化数据的增、删、查等管理接口，支持批量操作。

1.2 技术特点

- 高性能**：基于 FastAPI 框架构建，支持高并发异步请求，确保接口的快速响应。
- 高精度**：采用业界主流的 **Vision Transformer (ViT)** 模型进行图片特征提取和嵌入，保证了图片内容理解的准确性。
- 海量数据支持**：集成 Milvus 向量数据库，专为大规模向量检索设计，可支持十亿级别的图片向量数据。
- GPU 加速**：自动检测并优先使用 CUDA 进行模型推理，大幅提升图片向量化的处理速度。
- 易于部署**：提供 Docker 和 Docker Compose 部署方案，实现一键式环境搭建和应用启动，极大简化了部署和运维难度。

- **安全性:** 通过 API Key 机制保护接口, 确保只有授权用户才能访问服务。同时提供高危操作 (如数据库重建) 的密码保护。
- **可扩展性:** 系统采用微服务架构设计, 各个组件 (API 服务、嵌入模型、数据库) 松耦合, 便于独立升级和扩展。

1.3 开发语言与主要技术栈

- **开发语言:** Python 3.11
- **Web 框架:** FastAPI - 一个现代、高性能的 Python Web 框架。
- **向量数据库:** Milvus - 一款开源的、专为向量相似性搜索和分析而设计的云原生数据库。
- **图片嵌入模型:** 基于 Vision Transformer 架构的强大图片嵌入模型。
- **深度学习框架:** PyTorch & Hugging Face Transformers - 用于加载和运行深度学习模型。
- **配置管理:** Pydantic-Settings - 用于从环境变量或 `.env` 文件加载和验证配置。
- **容器化:** Docker & Docker Compose - 用于构建、分发和运行应用的容器化解决方案。