****

**义眼盯真 —— 图像篡改检测系统  
用户文档说明书**

目录

[义眼盯真 —— 图像篡改检测系统 用户文档说明书 I](#_Toc198108800)

[1 软件技术描述 3](#_Toc198108801)

[1.1 技术原理 3](#_Toc198108802)

[1.2 系统架构 3](#_Toc198108803)

[1.3 数据处理与安全 3](#_Toc198108804)

[2 安装说明 4](#_Toc198108805)

[2.1 Web 端安装与使用 4](#_Toc198108806)

[2.2 服务器端安装与部署 4](#_Toc198108807)

[2.2.1 硬件环境准备 4](#_Toc198108808)

[2.2.2 用户体验目标 4](#_Toc198108809)

[2.2.3 网络环境搭建 5](#_Toc198108810)

[3 使用说明 6](#_Toc198108811)

[3.1 图像检测 6](#_Toc198108812)

[3.2 视频检测 6](#_Toc198108813)

[3.3 系统交互 7](#_Toc198108814)

# 软件技术描述

## 技术原理

“义眼盯真 —— 图像篡改检测系统” 依托深度学习技术，以基于 YOLO-11n-seg 优化的深度学习模型为核心。该模型融合卷积神经网络（CNN）架构与注意力机制，实现高效的图像特征提取与分析。其集成目标检测和实例分割功能，能精准识别 Deepfake 深度伪造、Photoshop 局部篡改、图像拼接移植等主流伪造手段，检测精度可达像素级篡改轨迹分析。系统借助多任务学习框架，同步执行图像分类（判断真伪）和区域定位（标注篡改区域）任务，并利用边缘检测算法，对疑似篡改区域进行动态标注，以颜色梯度展示篡改置信度分布。

## 系统架构

系统采用 B-S 架构与前后端分离模式。前端基于 HTML5、CSS3、JavaScript 构建交互界面，负责接收用户操作指令并展示检测结果，涵盖图像上传、检测进度展示、结果可视化及报告导出等功能，且界面具备响应式设计，可适配多种设备。后端选用 Python 的 Flask 框架处理业务逻辑，集成 PyTorch 深度学习框架进行图像检测模型的推理运算，运用 MySQL 数据库存储用户信息、图像元数据和检测结果，通过云存储保存图像和视频文件，前后端之间以 JSON 格式进行数据交互。

## 数据处理与安全

系统运用数据脱敏技术，借助差分隐私和区域模糊化处理手段，对用户上传图像中的敏感信息（如人脸、车牌等）进行不可逆加密，严格遵循相关法规的合规要求。数据库设计遵循第三范式，使用 MySQL 关系型数据库存储结构化数据，并结合云存储管理图像、视频等大文件，以此确保数据的规范性、完整性和可扩展性。此外，系统配备完善的权限控制和日志记录机制，保障系统安全和操作的可追溯性。

# 安装说明

## Web 端安装与使用

Web 端使用：本系统 Web 端无需安装，用户可通过主流浏览器（如 Chrome、Firefox、Safari 等）访问。在浏览器地址栏输入已部署系统的服务器提供的网址，即可进入系统。

## 服务器端安装与部署

### 硬件环境准备

* 服务器推荐使用 Intel Xeon 处理器，具备强大的计算能力，可满足系统复杂运算需求。
* 内存要求 16GB 以上，保障系统运行时高效处理数据，避免内存不足导致的性能瓶颈。
* 配备大容量硬盘，用于存储系统文件、用户数据、检测结果等。
* 若涉及模型训练，需安装 NVIDIA GPU，加速深度学习模型的训练和推理，提升系统检测效率。

### 用户体验目标

* **操作系统选择**：服务器操作系统可选 Linux 或 Windows Server。Linux 系统凭借其稳定性、安全性和开源特性，适合技术能力较强的运维团队；Windows Server 具有良好的图形界面和兼容性，便于初学者管理。
* **后端依赖安装**：后端以 Python 3.12 为主要编程语言。首先，安装 Anaconda 进行虚拟化隔离，确保开发环境的一致性。在 Anaconda 环境中，使用pip install -r requirements.txt命令安装 PyTorch 深度学习框架、OpenCV 图像处理库等项目所需的依赖包，这些依赖包是系统后端功能实现的基础。
* **数据库设置**：使用 MySQL 8.0 + 作为数据库管理系统。在 MySQL 中，执行image\_db.sql文件来创建数据库并构建数据表结构，用于存储用户信息、图像元数据、检测记录等结构化数据。创建完成后，需配置数据库连接信息，确保后端服务能够正确访问数据库。
* **前端环境支持**：前端支持主流桌面和移动操作系统，兼容常见浏览器新版本。

### 网络环境搭建

* 服务器网络需稳定且带宽充足，保证用户使用系统时数据传输快速，避免卡顿、延迟。
* 移动端用户建议使用 4G 及以上移动网络或稳定 Wi-Fi 连接，确保与服务器数据交互顺畅。

# 使用说明

## 图像检测

1. 进入系统 “图像检测” 页面，用户可选择 “拼接篡改检测” 或 “AI深度篡改检测”。
2. 点击 “上传图片” 按钮，从本地设备选择要检测的图片。系统会自动识别图像格式，若格式不支持，将提示 “文件格式不支持，请上传正确格式的图像”。
3. 图片上传成功后，系统自动进行预处理，随后调用优化的深度学习模型开展检测工作。检测完成后，生成包含图像基本信息、检测结果、篡改区域定位、篡改置信度分析等内容的检测报告，并展示在页面上。同时，系统支持多种格式（如 PDF、HTML）导出报告，用户点击 “导出报告” 按钮，选择所需格式即可下载。

## 视频检测

1. 进入系统 “视频检测” 页面。点击 “选择视频” 按钮，从本地设备的存储路径中选取要检测的视频文件。系统支持常见的视频格式，如 MP4、AVI、MKV 等。
2. 选定视频后，点击 “上传视频” 按钮，系统开始上传视频。上传过程中，页面会实时显示上传进度条，让用户清楚了解上传状态。上传速度取决于网络状况和视频文件大小，若网络不稳定或视频文件过大，可能需要一些时间，请用户耐心等待。
3. 用户点击“开始检测”按钮后，系统立即调用基于深度学习的检测模型对视频进行检测。检测过程中，页面会展示动态的检测进度，包括已检测的帧数、预计剩余时间等信息，让用户随时了解检测进展。由于视频检测的数据量较大，检测过程可能会花费一定时间，尤其是时长较长的视频。在此期间，请用户不要进行可能中断检测进程的操作，如关闭页面或切换网络。
4. 检测完成后，系统生成检测后的视频。检测结果视频包括由原视频抽帧检测的检测框和掩膜，清晰地标注视频中被篡改的部分及置信度。
5. 检测视频生成后，会展示在页面的右方区域。用户可以在该区域查看详细的视频内容。

## 系统交互

1. 系统界面设计简洁，操作便捷。检测过程中，进度展示区实时反馈检测进度，结果展示区通过热图和标注精准呈现图像篡改区域，并搭配详细文字报告，帮助用户了解图像的真伪、篡改方式及置信度。
2. 系统集成 OpenAI 大语言模型接口（LLM），在 “AI 分析助手” 输入框中输入问题，用户即可与 AI 互动，获取检测结果解读、图像篡改技术原理、防范措施等信息。该功能支持自然语言交互、批量提问和多轮对话，且对用户提问和模型回答进行加密处理，保障数据安全。