

# 录音录像应用

---

一个基于Flask和WebSocket的实时录音录像应用，支持摄像头录制、AI聊天和视频合成功能。

## 功能特性

---

### 摄像头录制

- 实时摄像头预览
- 一键开启/关闭摄像头
- 同步录音录像功能
- 实时状态显示

### AI聊天助手

- 集成大模型API
- 实时聊天交互
- 美观的聊天界面
- 消息历史记录

### 视频处理

- 缓存最近5秒音视频数据
- 每秒自动合成视频
- 视频列表管理
- 一键下载功能

## Web界面

- 响应式左右布局设计
- 现代化UI界面
- 实时状态反馈
- 移动端适配

## 技术架构

---

### 后端技术栈

- **Flask**: Web框架
- **Flask-SocketIO**: WebSocket通信
- **OpenCV**: 图像处理
- **ffmpeg-python**: 音视频封装与转码
- **NumPy**: 数据处理
- (可选) **MoviePy**: 扩展视频处理

### 前端技术栈

- **HTML5**: 页面结构
- **CSS3**: 样式设计
- **JavaScript**: 交互逻辑
- **Socket.IO**: 实时通信
- **WebRTC**: 媒体流处理

# 安装部署

---

## 1. 环境要求

- Python 3.8+
- 现代浏览器（支持WebRTC）
- 摄像头和麦克风设备

## 2. 安装依赖

```
# 激活conda环境（如果使用）
conda activate new_vllm

# 安装Python依赖
pip install -r requirements.txt
```

## 3. 配置环境变量

创建 `.env` 文件：

```
# OpenAI API 配置（可选）
OPENAI_API_KEY=your-api-key-here
OPENAI_API_BASE=https://api.openai.com/v1
OPENAI_MODEL=gpt-3.5-turbo

# 本地模型配置（可选）
QWEN_MODEL_PATH=/path/to/Qwen3-Omni-30B-A3B-Instruct

# Flask/服务配置
SECRET_KEY=your-secret-key-here
FLASK_ENV=development
PORT=5000
```

## 4. 启动应用

```
python app.py
```

应用将在 `http://localhost:5000` 启动。

## 使用说明

---

### 摄像头录制

1. 点击"开启摄像头"按钮
2. 授权浏览器访问摄像头和麦克风
3. 点击"开始录制"开始录音录像
4. 点击"停止录制"结束录制

### AI聊天

1. 在右侧聊天框输入消息
2. 点击发送按钮或按Enter键
3. AI助手将实时回复

### 视频管理

- 录制的视频会自动保存到 `videos/` 目录
- 在界面下方可查看已保存的视频列表
- 点击下载按钮可下载视频文件

# 项目结构

```
video_capture/
├── app.py                # 主应用与路由/事件
├── config.py            # 配置项
├── modules/             # 业务模块
│   ├── openai_client.py # Chat API 客户端
│   ├── qwen_omni.py     # 本地模型推理器
│   ├── qwen_service.py  # 模型服务（单例）
│   ├── realtime_manager.py # 实时管线协调
│   ├── video_processor.py # 视频合成与任务队列
│   └── video_queue.py   # 会话队列与状态
├── templates/           # 前端页面
│   └── index.html       # 主页面
├── static/              # 静态资源
│   └── image.png        # 架构图
├── requirements.txt      # 依赖列表
├── README.md            # 项目说明
└── videos/              # 视频保存目录
```

# 接口与事件

## HTTP 路由

- GET /：主页
- GET /videos、GET /api/videos：视频列表
- GET /videos/<filename>：下载视频文件
- POST /api/chat：AI聊天
- POST /api/chat/clear：清除聊天历史
- GET /api/chat/history：查询聊天历史
- GET /status：系统/会话状态

- GET /api/qwen/status : Qwen 模型服务状态
- POST /api/qwen/analyze : 视频情感分析

## SocketIO 事件

- connect / disconnect
- start\_recording / stop\_recording
- video\_frame : 接收帧, 后端回 video\_frame\_ack{server\_ts, client\_ts, queue\_info}
- audio\_chunk : 接收音频块 (自动清理前缀与填充)
- get\_session\_status → session\_status

## 核心功能实现

---

### 音视频数据流处理

```
# 视频帧处理
@socketio.on('video_frame')
def handle_video_frame(data):
    frame_data = data['frame']
    # 解码base64图像数据
    # 添加到缓存队列
    # 定时合成视频

# 音频数据处理
@socketio.on('audio_chunk')
def handle_audio_chunk(data):
    audio_data = data['audio']
    # 解码音频数据
    # 添加到音频缓存
    # 同步音视频时间戳
```

## 视频合成算法

- 使用 `opencv` 写入临时 `AVI` 帧序列
- 使用 `ffmpeg-python` 合并 `AVI` + `WAV` 输出 `MP4`
- 编码参数: `vcodec=libx264`、`acodec=aac`
- 自动处理音视频同步与尺寸对齐
- 队列+工作线程设计, 优化内存与处理速度

## WebSocket通信

- 实时双向数据传输
- 支持大文件分块传输
- 自动重连机制
- 错误处理和状态同步

## 配置说明

---

### 音视频参数

```
VIDEO_CODEC = 'libx264'      # 视频编码格式
AUDIO_CODEC = 'aac'          # 音频编码格式
VIDEO_BITRATE = '1000k'      # 视频比特率
AUDIO_BITRATE = '128k'       # 音频比特率
FRAME_RATE = 10              # 帧率
BUFFER_DURATION = 5          # 缓存时长 (秒)
SAVE_INTERVAL = 1            # 保存间隔 (秒)
```

## API配置

```
OPENAI_API_KEY = '...'
OPENAI_API_BASE = 'https://api.openai.com/v1'
OPENAI_MODEL = 'gpt-3.5-turbo'
OPENAI_MAX_TOKENS = 1000
OPENAI_TEMPERATURE = 0.7
```

## 注意事项

---

### 浏览器兼容性

- 需要支持WebRTC的现代浏览器
- 建议使用Chrome、Firefox、Safari最新版本
- 需要HTTPS环境（生产环境）

### 性能优化

- 视频帧率设置为10fps以平衡质量和性能
- 音频采样率44.1kHz，单声道
- 自动清理过期缓存数据
- 支持并发用户访问
- 前端根据 `video_frame_ack` 的 RTT 与队列负载自适应调整图像质量/分辨率/发送间隔

### 安全考虑

- 生产环境需配置HTTPS
- API密钥通过环境变量配置
- 文件上传大小限制



- 输入数据验证和过滤

## 故障排除

---

### 常见问题

1. 摄像头无法访问: 检查浏览器权限设置
2. 录制无声音: 确认麦克风权限和设备状态
3. 视频合成失败: 检查FFmpeg安装和依赖
4. **API**调用失败: 验证API密钥和网络连接

### 日志查看

应用日志会输出到控制台，包含详细的错误信息和调试信息。

## 扩展功能

---

### 可扩展特性

- 支持多用户同时录制
- 添加视频滤镜和特效
- 集成云存储服务
- 支持直播推流
- 添加人脸识别功能

### 二次开发

项目采用模块化设计，便于扩展和定制： - 音视频处理模块可独立使用 - WebSocket通信层可复用 - 前端界面支持主题定制 - 配置系统支持多环境部署

# 许可证

---

本项目仅供学习和研究使用。

# 联系方式

---

如有问题或建议，请通过以下方式联系： - 项目地址: `/data/testm1lm/project/ver1/video_capture/` - 技术支持: 请查看代码注释和文档