

Hướng dẫn sử dụng Live Caption Logger

Phiên bản: 1.0

Tác giả: Manus AI

Ngày cập nhật: 13/06/2025

Giới thiệu

Live Caption Logger là một ứng dụng được thiết kế để ghi lại và lưu trữ bản ghi từ tính năng Live Caption trên Windows 11. Ứng dụng sử dụng công nghệ OCR (Optical Character Recognition) để đọc văn bản hiển thị trên màn hình và chuyển đổi thành các bản ghi transcript có thể tìm kiếm và xuất ra nhiều định dạng khác nhau.

Tính năng chính

- Ghi lại Live Caption tự động:** Sử dụng OCR để đọc văn bản từ vùng hiển thị Live Caption
- Xử lý văn bản thông minh:** Loại bỏ trùng lặp, làm sạch văn bản và ghép nối các đoạn
- Lưu trữ có tổ chức:** Quản lý các phiên ghi chép với metadata đầy đủ
- Xuất đa định dạng:** Hỗ trợ xuất ra Text, Markdown, JSON, CSV, SRT và PDF
- Giao diện thân thiện:** Giao diện đồ họa dễ sử dụng với Tkinter
- Báo cáo thống kê:** Tạo báo cáo chi tiết về nội dung và chất lượng transcript

Yêu cầu hệ thống

- Hệ điều hành:** Windows 11 (khuyến nghị) hoặc Windows 10
- Python:** Phiên bản 3.8 trở lên
- Tesseract OCR:** Phiên bản 4.0 trở lên
- RAM:** Tối thiểu 4GB, khuyến nghị 8GB
- Dung lượng ổ cứng:** Tối thiểu 500MB cho ứng dụng và dữ liệu

Cài đặt

Bước 1: Cài đặt Python

- Tải Python từ python.org
- Chọn phiên bản 3.8 trở lên

3. Trong quá trình cài đặt, đảm bảo chọn "Add Python to PATH"
4. Kiểm tra cài đặt bằng cách mở Command Prompt và gõ: `python --version`

Bước 2: Cài đặt Tesseract OCR

1. Tải Tesseract từ [GitHub releases](#)
2. Chọn file installer phù hợp với hệ thống (thường là `tesseract-ocr-w64-setup-v5.x.x.exe`)
3. Cài đặt với các tùy chọn mặc định
4. Thêm đường dẫn Tesseract vào PATH (thường là `C:\\Program Files\\Tesseract-OCR`)

Bước 3: Tải và cài đặt Live Caption Logger

1. Tải mã nguồn từ repository
2. Giải nén vào thư mục mong muốn (ví dụ: `C:\\LiveCaptionLogger`)
3. Mở Command Prompt với quyền Administrator
4. Chuyển đến thư mục ứng dụng: `cd C:\\LiveCaptionLogger`
5. Cài đặt dependencies: `pip install -r requirements.txt`

Bước 4: Kiểm tra cài đặt

Chạy script kiểm tra để đảm bảo mọi thứ hoạt động bình thường:

```
python test_modules.py
```

Nếu tất cả tests đều PASS, ứng dụng đã sẵn sàng sử dụng.

Hướng dẫn sử dụng cơ bản

Khởi động ứng dụng

1. Mở Command Prompt
2. Chuyển đến thư mục ứng dụng
3. Chạy lệnh: `python src/main.py`
4. Giao diện chính sẽ xuất hiện

Thiết lập vùng chụp Live Caption

Trước khi bắt đầu ghi chép, bạn cần chỉ định vùng màn hình nơi Live Caption hiển thị:

1. **Tự động phát hiện:**
2. Nhấn nút "Tự động phát hiện"
3. Ứng dụng sẽ cố gắng tìm vùng Live Caption trên màn hình
4. Nếu thành công, thông tin vùng sẽ hiển thị
5. **Chọn thủ công:**
6. Nhấn nút "Chọn vùng Live Caption"
7. Màn hình sẽ chuyển sang chế độ chọn vùng
8. Kéo chuột để chọn vùng chứa Live Caption
9. Nhấn Escape để hủy nếu cần

Bắt đầu ghi chép

1. Nhập tiêu đề cho phiên ghi chép (hoặc sử dụng tiêu đề mặc định)
2. Đảm bảo đã chọn vùng chụp Live Caption
3. Bật Live Caption trên Windows 11 (Windows + Ctrl + L)
4. Nhấn nút "Bắt đầu ghi"
5. Ứng dụng sẽ bắt đầu ghi lại văn bản từ Live Caption

Theo dõi quá trình ghi chép

Trong khi ghi chép, bạn có thể theo dõi:

- **Văn bản Live Caption:** Hiển thị trong vùng text chính
- **Thống kê thời gian thực:**
 - Thời gian phiên hiện tại
 - Số từ đã ghi được
 - Độ tin cậy OCR trung bình

Kết thúc phiên ghi chép

1. Nhấn nút "Dừng ghi"
2. Phiên sẽ được lưu tự động vào cơ sở dữ liệu
3. Bạn có thể xuất transcript ngay lập tức hoặc sau này

Tính năng nâng cao

Xuất transcript

Ứng dụng hỗ trợ nhiều định dạng xuất:

1. **Text (.txt):** Định dạng văn bản thuần túy với timestamp
2. **Markdown (.md):** Định dạng Markdown với cấu trúc rõ ràng
3. **JSON (.json):** Dữ liệu có cấu trúc với metadata đầy đủ
4. **CSV (.csv):** Định dạng bảng tính cho phân tích dữ liệu
5. **SRT (.srt):** Định dạng subtitle cho video
6. **PDF (.pdf):** Báo cáo chuyên nghiệp (từ Markdown)

Quản lý phiên ghi chép

- **Xem danh sách phiên:** Tất cả phiên được lưu trong tab "Phiên đã ghi"
- **Xem chi tiết:** Double-click vào phiên để xem nội dung đầy đủ
- **Xuất phiên cũ:** Có thể xuất bất kỳ phiên nào đã lưu

Cấu hình nâng cao

Chỉnh sửa file `src/utils/config.py` để tùy chỉnh:

- **Khoảng thời gian chụp:** `CAPTURE_CONFIG['interval']`
- **Độ tin cậy OCR tối thiểu:** `TEXT_PROCESSING_CONFIG['min_confidence']`
- **Ngôn ngữ OCR:** `OCR_CONFIG['language']`
- **Giao diện:** `UI_CONFIG['theme']`

Xử lý sự cố

Lỗi thường gặp

1. **"Tesseract not found"**
2. Kiểm tra Tesseract đã được cài đặt
3. Đảm bảo đường dẫn Tesseract trong PATH
4. Thử chạy `tesseract --version` trong Command Prompt
5. **"Cannot capture screen"**
6. Kiểm tra quyền truy cập màn hình
7. Đảm bảo không có phần mềm bảo mật chặn

8. Thử chạy ứng dụng với quyền Administrator

9. **"Low OCR confidence"**

10. Kiểm tra chất lượng hiển thị Live Caption

11. Tăng kích thước font của Live Caption

12. Đảm bảo vùng chụp chính xác

13. **"Database error"**

14. Kiểm tra quyền ghi file trong thư mục ứng dụng

15. Đảm bảo đủ dung lượng ổ cứng

16. Thử xóa file database cũ nếu bị lỗi

Tối ưu hóa hiệu suất

1. **Giảm khoảng thời gian chụp** nếu máy yếu
2. **Tăng độ tin cậy tối thiểu** để lọc văn bản kém chất lượng
3. **Đóng các ứng dụng không cần thiết** khi ghi chép
4. **Sử dụng SSD** để cải thiện tốc độ lưu trữ

Demo và ví dụ

Chạy demo

Để trải nghiệm nhanh các tính năng:

```
python demo.py
```

Demo sẽ tạo một phiên ghi chép mẫu với dữ liệu giả lập.

Xuất nâng cao

Để trải nghiệm tính năng xuất nâng cao:

```
python advanced_export.py
```

Script này cho phép xuất nhiều định dạng cùng lúc và tạo báo cáo chi tiết.

Phát triển và tùy chỉnh

Cấu trúc mã nguồn

```
live_caption_logger/
├── src/
│   ├── core/           # Các module cốt lõi
│   ├── ui/             # Giao diện người dùng
│   └── utils/          # Tiện ích và cấu hình
├── demo.py             # Script demo
├── test_modules.py     # Script kiểm thử
└── requirements.txt    # Dependencies
```

Mở rộng tính năng

1. **Thêm ngôn ngữ OCR mới:** Chỉnh sửa `OCR_CONFIG` trong `config.py`
2. **Tùy chỉnh giao diện:** Chỉnh sửa các file trong `src/ui/`
3. **Thêm định dạng xuất:** Mở rộng `StorageManager` trong `src/core/storage.py`
4. **Cải thiện OCR:** Tinh chỉnh `OCRProcessor` trong `src/core/ocr_processor.py`

Đóng góp

Nếu bạn muốn đóng góp cho dự án:

1. Fork repository
2. Tạo branch mới cho tính năng
3. Commit các thay đổi
4. Tạo Pull Request
5. Mô tả chi tiết các thay đổi

Bảo mật và quyền riêng tư

Dữ liệu cục bộ

- Tất cả dữ liệu được lưu trữ cục bộ trên máy tính
- Không có dữ liệu nào được gửi lên internet
- Database SQLite được mã hóa cơ bản

Quyền truy cập

Ứng dụng cần các quyền sau:

- **Chụp màn hình:** Để đọc Live Caption
- **Ghi file:** Để lưu transcript và database
- **Đọc cấu hình:** Để tải settings

Khuyến nghị bảo mật

1. **Sao lưu dữ liệu** định kỳ
2. **Không chia sẻ** file database chứa thông tin nhạy cảm
3. **Cập nhật** ứng dụng thường xuyên
4. **Kiểm tra** quyền truy cập của ứng dụng

Hỗ trợ và liên hệ

Báo cáo lỗi

Nếu gặp lỗi, vui lòng cung cấp:

1. **Mô tả lỗi** chi tiết
2. **Các bước tái tạo** lỗi
3. **Thông tin hệ thống** (Windows version, Python version)
4. **Log files** nếu có

Yêu cầu tính năng

Để đề xuất tính năng mới:

1. **Mô tả tính năng** cần thiết
2. **Lý do** cần tính năng này
3. **Ví dụ sử dụng** cụ thể
4. **Độ ưu tiên** mong muốn

Cộng đồng

- **GitHub Issues:** Để báo cáo lỗi và yêu cầu tính năng
 - **Discussions:** Để thảo luận và chia sẻ kinh nghiệm
 - **Wiki:** Để tài liệu bổ sung và hướng dẫn nâng cao
-

Tài liệu này được tạo bởi Manus AI và sẽ được cập nhật thường xuyên để phản ánh các tính năng mới và cải tiến.