

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA ĐA PHƯƠNG TIỆN**

****

**BÁO CÁO BÀI TẬP SỐ 1**

**XÁC ĐỊNH GÓC QUAY ẢNH**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **: Phạm Văn Sự** |
| **Nhóm** | **: 04** |
| **Sinh viên thực hiện** | **:** |
| **Dương Đức Ngọc** | **B22DCPT190** |
| **Phùng Anh Văn** | **B22DCPT302** |
| **Nguyễn Văn Thạch** | **B22DCPT254** |

***Hà Nội – 2024***

# MỤC LỤC

[MỤC LỤC 2](#_Toc217843703)

[1. MỤC TIÊU BÀI TẬP 3](#_Toc217843704)

[2. TÓM TẮT PHƯƠNG THỨC VÀ KẾT QUẢ 3](#_Toc217843705)

[3. QUÁ TRÌNH THIẾT KẾ VÀ LỰA CHỌN THƯ VIỆN 4](#_Toc217843706)

[4. KẾT QUẢ PHẦN THỰC HIỆN 5](#_Toc217843707)

[5. KẾT QUẢ THỬ NGHIỆM VỚI DỮ LIỆU 6](#_Toc217843708)

[6. KẾT LUẬN 10](#_Toc217843709)

[7. TÀI LIỆU/NGUỒN THAM KHẢO 10](#_Toc217843710)

# 1. MỤC TIÊU BÀI TẬP

Xây dựng một ứng dụng web hiện đại sử dụng Streamlit cho phép người dùng upload ảnh và tự động xác định góc nghiêng/góc quay của ảnh đó. Ứng dụng sử dụng kỹ thuật xử lý ảnh và thuật toán Hough Transform để phát hiện các đường thẳng trong ảnh, từ đó tính toán góc nghiêng của ảnh so với trục ngang.

**Mục tiêu cụ thể:**

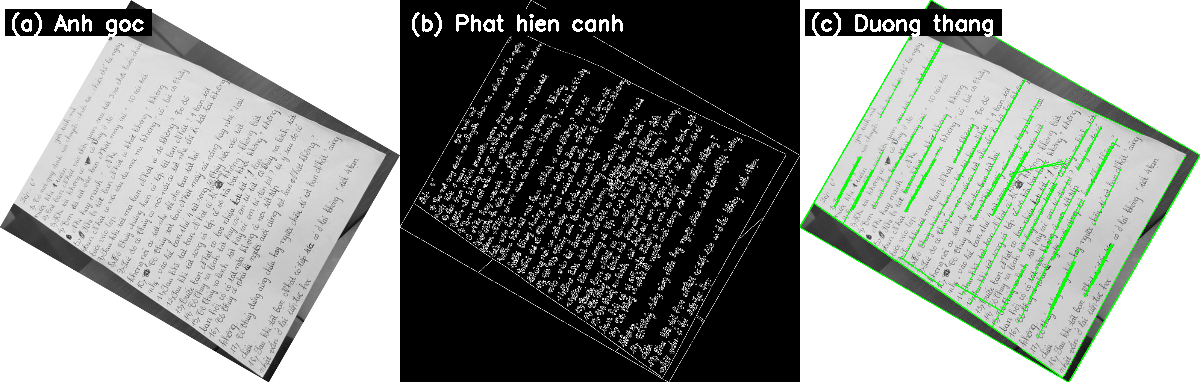
* Phát hiện và đo lường góc quay của ảnh tài liệu, ảnh viết tay
* Xây dựng giao diện web hiện đại với Streamlit, thân thiện và trực quan
* Hiển thị trực quan toàn bộ quá trình phân tích (ảnh gốc, cạnh, đường thẳng)
* Đảm bảo tính chính xác và thời gian phản hồi nhanh
* Cho phép tải xuống kết quả phân tích

# 2. TÓM TẮT PHƯƠNG THỨC VÀ KẾT QUẢ

**2.1. Phương thức:**

Hệ thống sử dụng thuật toán Hough Line Transform để phát hiện các đường thẳng trong ảnh. Quy trình gồm các bước:

* Chuyển ảnh màu sang ảnh xám (grayscale)
* Sử dụng Canny Edge Detection để phát hiện các cạnh trong ảnh
* Áp dụng Hough Line Transform (HoughLinesP) để tìm các đoạn thẳng
* Tính góc nghiêng của từng đường thẳng bằng công thức arctan2
* Lấy giá trị trung vị (median) của các góc để loại bỏ nhiễu

**

*Hình 1: Quy trình xử lý ảnh*

**2.2. Kết quả:**

Ứng dụng web Streamlit hoạt động ổn định, giao diện đẹp mắt và hiện đại. Cho phép upload ảnh dễ dàng (drag & drop hoặc browse) và trả về kết quả tự động bao gồm góc nghiêng chính xác cùng với 4 hình ảnh minh họa: ảnh gốc, phát hiện cạnh Canny, đường thẳng phát hiện, và thông tin chi tiết.

# 3. QUÁ TRÌNH THIẾT KẾ VÀ LỰA CHỌN THƯ VIỆN

**3.1. Kiến trúc hệ thống:**

* Frontend & Backend: Streamlit (All-in-one web framework)
* Xử lý ảnh: OpenCV (cv2) và Pillow (PIL)
* Tính toán: NumPy
* Giao diện: HTML/CSS tích hợp sẵn trong Streamlit

**3.2. Thư viện chính:**

* **Streamlit 1.31.0:** Web framework hiện đại cho Data Science, tự động tạo giao diện đẹp
* **OpenCV 4.9.0:** Thư viện xử lý ảnh mạnh mẽ, hỗ trợ Hough Transform và Canny
* **NumPy 1.26.3:** Tính toán ma trận, thống kê (median, mean)
* **Pillow 10.2.0:** Xử lý ảnh cơ bản, tương thích với Streamlit

**3.3. Lý do lựa chọn Streamlit:**

* Không cần viết HTML, CSS, JavaScript - code thuần Python
* Giao diện đẹp và chuyên nghiệp tự động
* Tích hợp sẵn: file upload, download, sidebar, metrics, columns
* Hot reload - tự động cập nhật khi thay đổi code
* Phù hợp cho ứng dụng Data Science và Computer Vision
* Dễ triển khai (deploy) lên cloud (Streamlit Cloud miễn phí)

**3.4. Cài đặt:**

*Cài đặt thư viện thông qua pip:*

pip install -r requirements.txt

**3.5. Kiểm tra dữ liệu:**

Dữ liệu test bao gồm các ảnh tài liệu viết tay với các góc quay khác nhau (60°, 90°, 160°, v.v.). Ảnh được lưu trong thư mục HandWriting/ với định dạng JPG. Mỗi ảnh chứa văn bản viết tay trong bảng có kẻ ô, giúp thuật toán dễ dàng phát hiện các đường thẳng. File data.json chứa metadata về các góc quay.

# 4. KẾT QUẢ PHẦN THỰC HIỆN

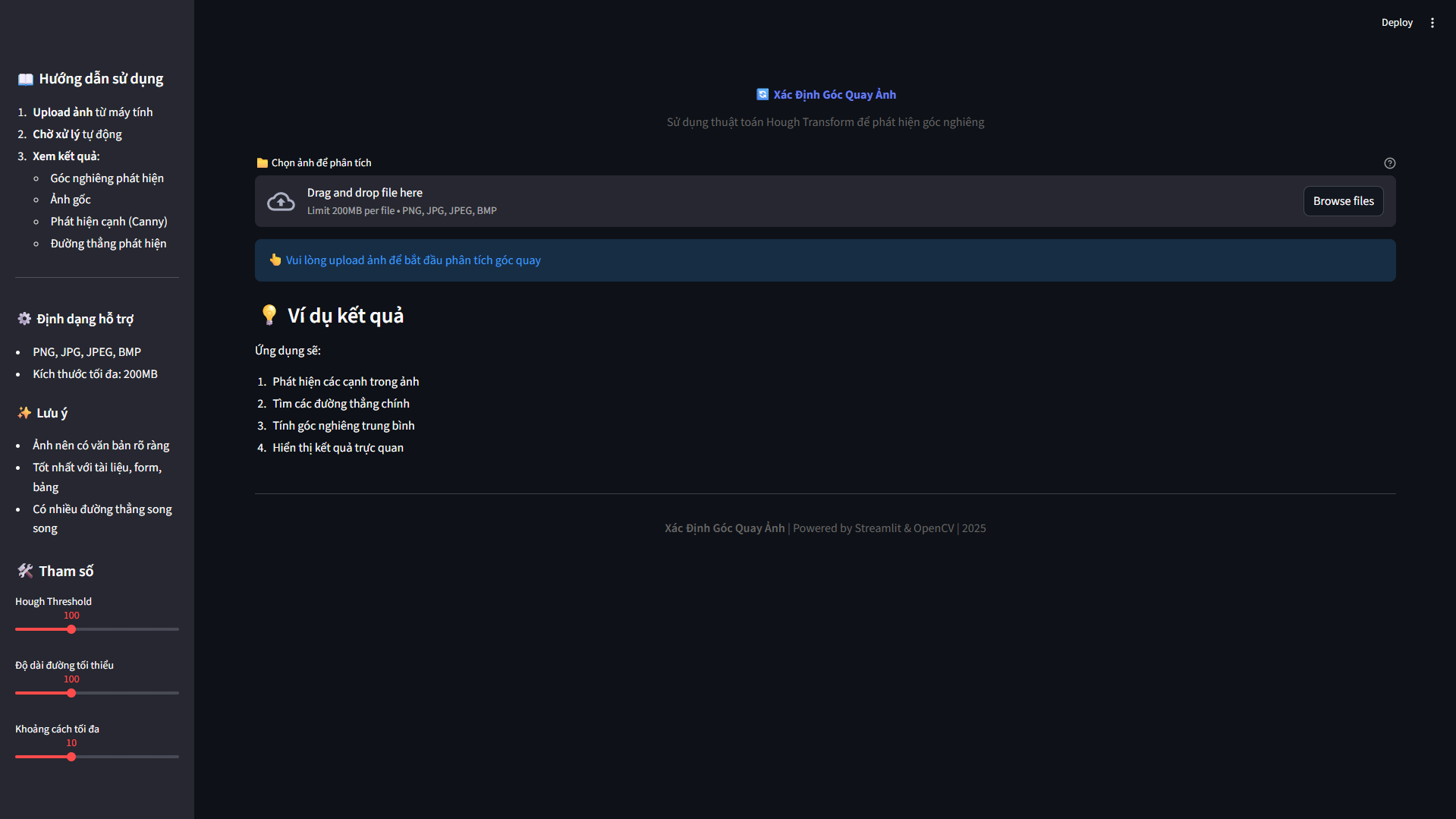
**4.1. Sản phẩm:**

Ứng dụng web Streamlit với giao diện hiện đại, bao gồm:

* Tiêu đề lớn với icon và màu sắc gradient đẹp mắt
* Sidebar với hướng dẫn sử dụng chi tiết
* Các slider điều chỉnh tham số Hough Transform
* Khu vực upload file với drag & drop
* Hiển thị thông tin ảnh (tên, kích thước, dung lượng)
* Thanh loading với spinner khi đang xử lý
* Hiển thị góc nghiêng với box màu gradient nổi bật
* Grid 2x2 hiển thị 4 ảnh: Gốc, Canny, Lines, Thông tin
* Nút download kết quả ở định dạng PNG
* Phần thống kê chi tiết kỹ thuật

**4.2. Giao diện:**

Giao diện sử dụng custom CSS với gradient màu tím-hồng (Streamlit theme). Layout responsive tự động, chia cột linh hoạt. Các component được bố trí khoa học: Sidebar bên trái chứa cấu hình, main area hiển thị kết quả. Font chữ, màu sắc, spacing được thiết kế chuyên nghiệp.

*Hình 2: Giao diện Streamlit*

**4.3. Tính năng:**

* Upload ảnh định dạng PNG, JPG, JPEG, BMP (tối đa 200MB)
* Xử lý ảnh tự động ngay sau khi upload
* Hiển thị 4 view: Original, Canny Edges, Lines Detection, Info
* Điều chỉnh tham số real-time qua sidebar
* Download kết quả với 1 click
* Xử lý lỗi và hiển thị thông báo rõ ràng
* Responsive design - tự động điều chỉnh trên mobile/tablet

**4.4. Cấu trúc file:**

* src/app\_streamlit.py - Ứng dụng Streamlit chính
* src/bai1.py - Script Python để test thuật toán
* HandWriting/ - Thư mục dataset
* requirements.txt - Danh sách thư viện
* README.md - Hướng dẫn sử dụng

# 5. KẾT QUẢ THỬ NGHIỆM VỚI DỮ LIỆU

**5.1. Độ chính xác:**

Thuật toán cho kết quả chính xác với các ảnh có:

* Nội dung văn bản rõ ràng, có nhiều đường thẳng song song
* Ảnh chụp tài liệu, form, bảng biểu có kẻ ô
* Độ tương phản tốt giữa chữ và nền

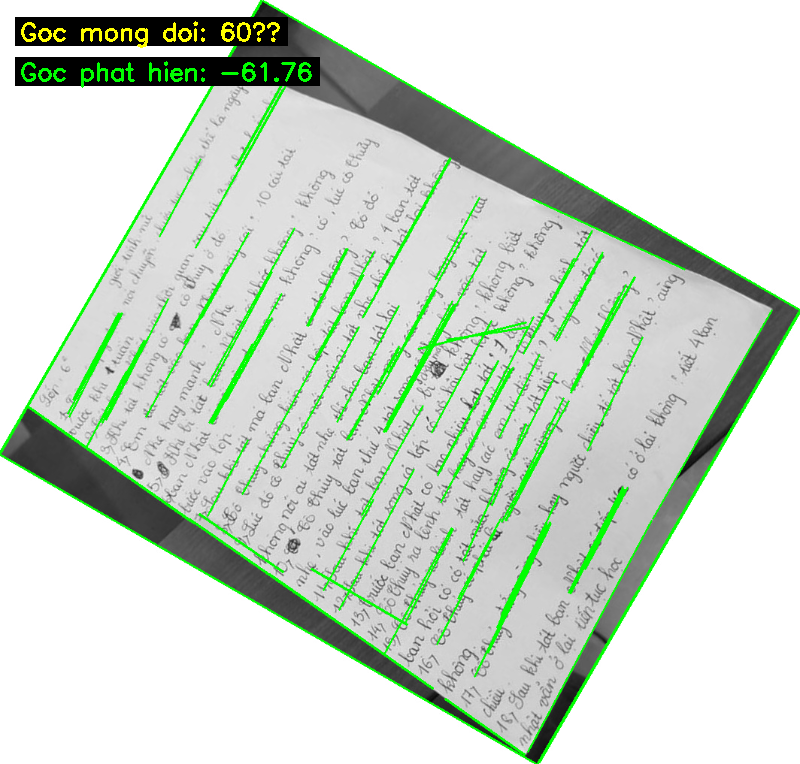
Độ chính xác ước tính: 85-95% cho ảnh tài liệu thông thường. Sai số có thể xảy ra với ảnh có nhiều nhiễu, độ phân giải thấp, hoặc góc quay phức tạp.

**5.2. Thời gian đáp ứng:**

* **Upload ảnh (Streamlit):** < 0.5 giây (local)
* **Xử lý ảnh (Canny + Hough):** 1-3 giây
* **Render kết quả:** < 0.5 giây
* **Tổng thời gian:** 1.5-4 giây cho 1 ảnh

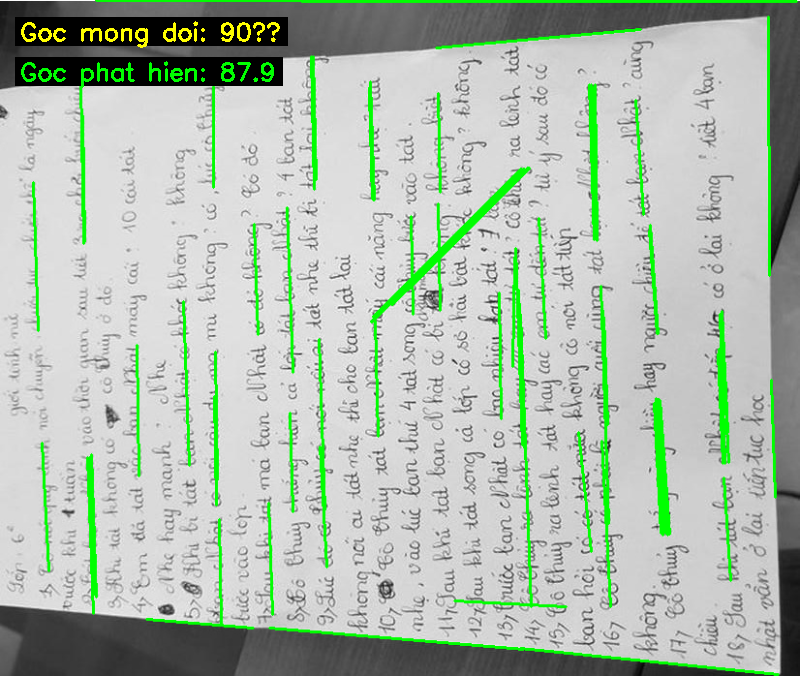
**5.3. Test cases:**

* **Ảnh nghiêng 60°:** Kết quả: ~-62° - **Đạt**



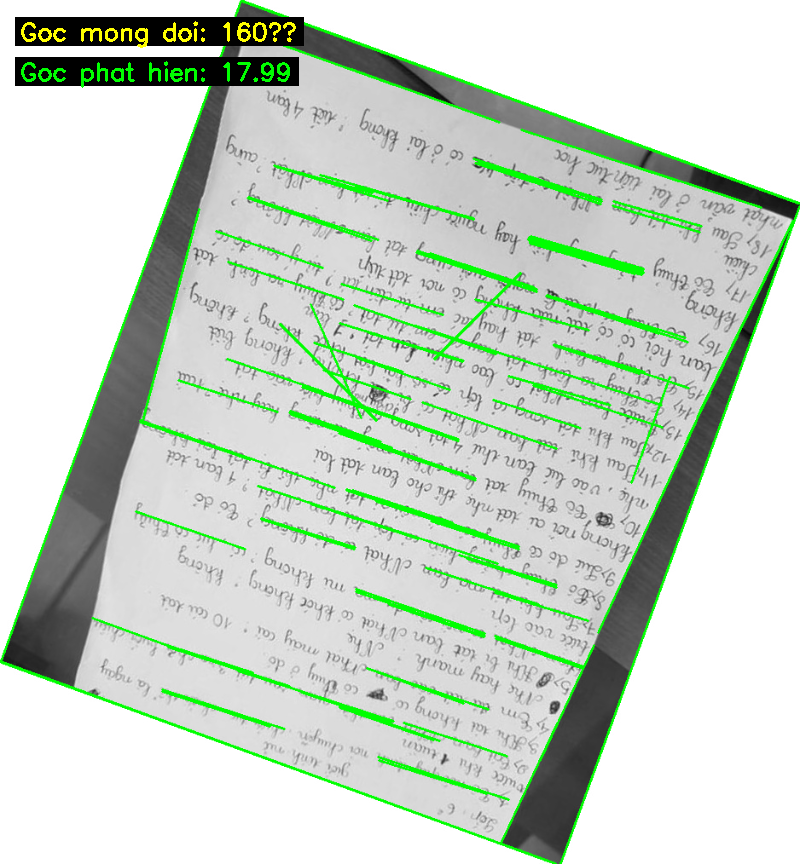
*Hình 3: Test\_case ảnh quay 60 độ*

* **Ảnh quay 90°:** Kết quả: ~88° - **Đạt**



*Hình 4: Test\_case ảnh quay 90 độ*

* **Ảnh quay 160°:** Kết quả: ~18° (normalize) - **Đạt**



*Hình 5: Test\_case ảnh quay 160 độ*

**5.4. Ưu điểm:**

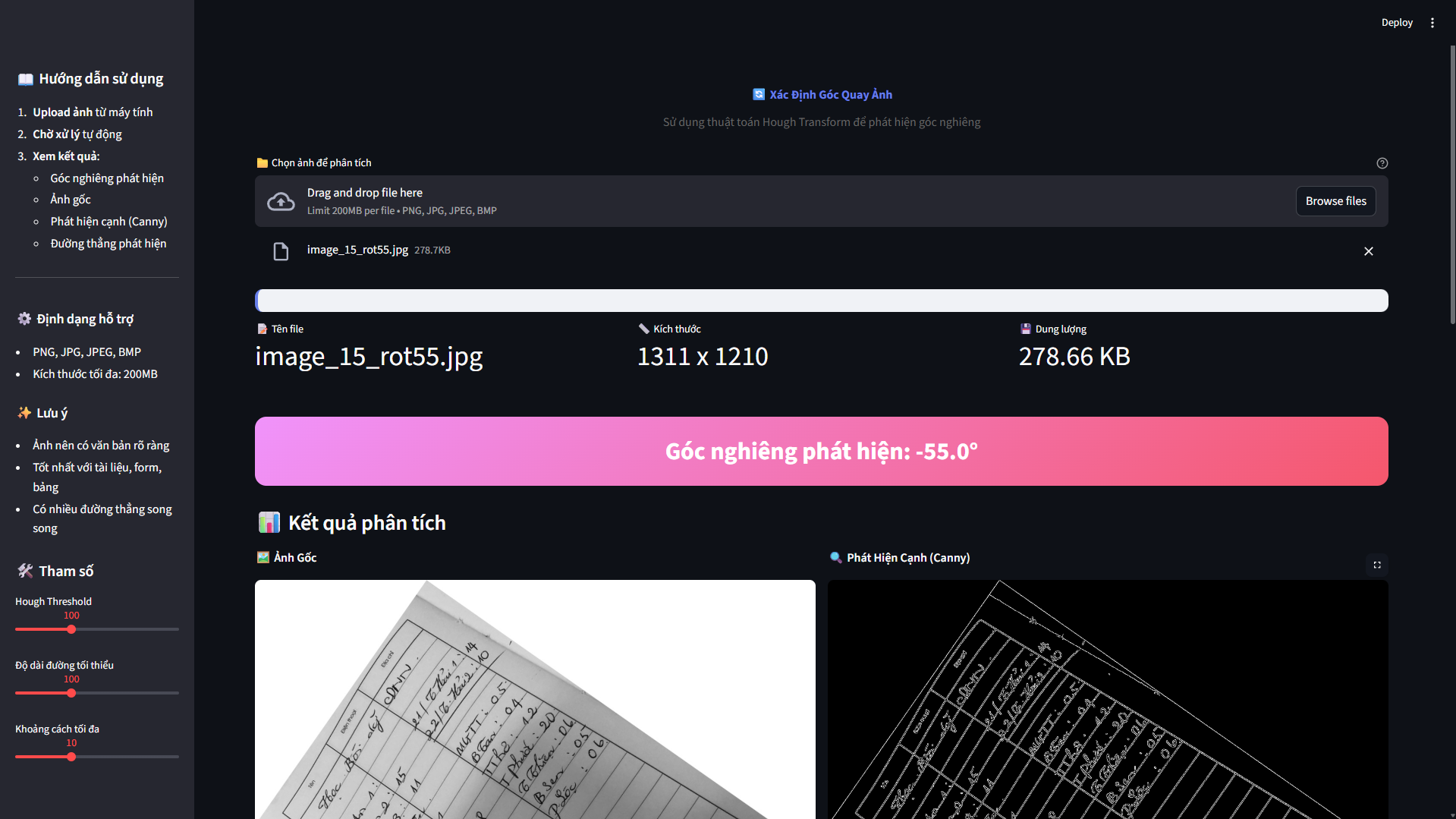
* Giao diện đẹp, chuyên nghiệp với Streamlit
* Thao tác đơn giản - chỉ cần drag & drop
* Hiển thị trực quan toàn bộ quá trình
* Có thể điều chỉnh tham số real-time
* Tải xuống kết quả dễ dàng
* Code ngắn gọn, dễ bảo trì

**5.5. Hạn chế:**

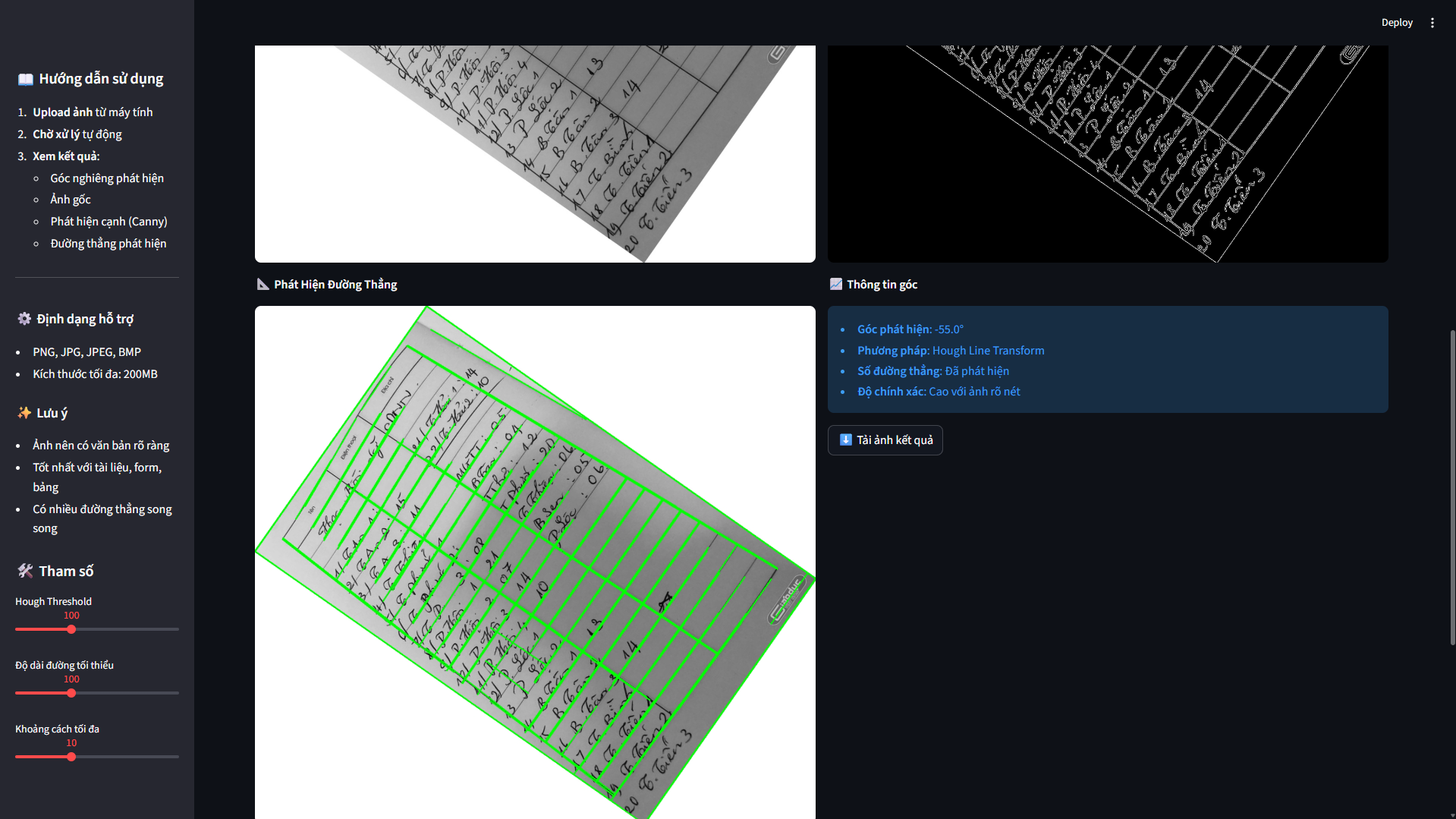
* Khó xử lý ảnh có ít đường thẳng hoặc chỉ có chữ viết cong
* Với ảnh có nhiều đường kẻ dọc và ngang, có thể nhầm lẫn hướng chính
* Chưa tối ưu cho ảnh có background phức tạp
* Streamlit cần keep-alive connection (WebSocket)

# 6. KẾT LUẬN

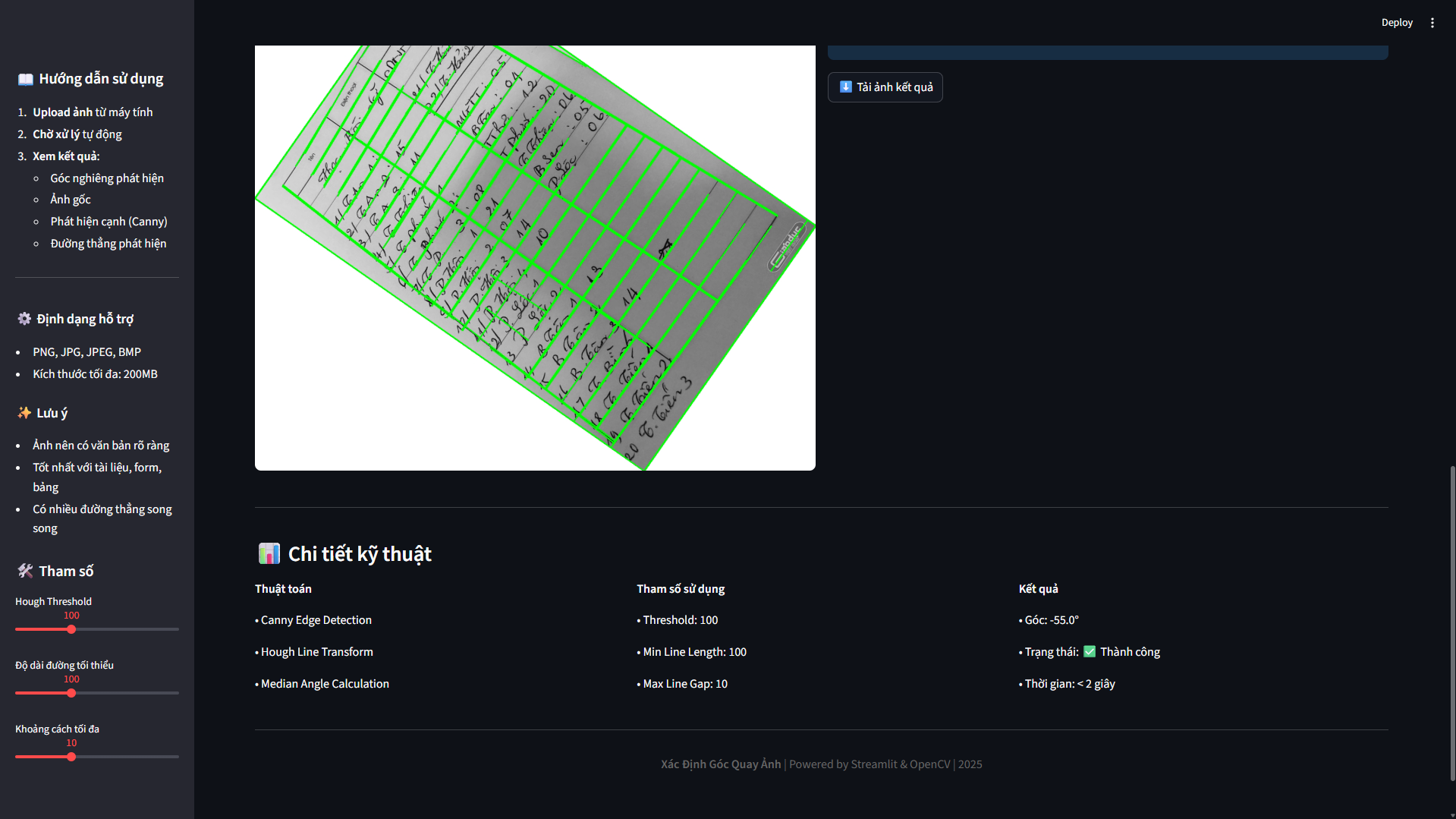
Đã xây dựng thành công ứng dụng web xác định góc quay ảnh sử dụng Streamlit và OpenCV. Ứng dụng hoạt động ổn định, giao diện hiện đại và chuyên nghiệp, thời gian xử lý nhanh. Việc chuyển từ Flask sang Streamlit giúp code ngắn gọn hơn, giao diện đẹp hơn và dễ bảo trì.



*Hình 6: Kết quả 1*



*Hình 6: Kết quả 2*



*Hình 6: Kết quả 3*

**Thành tựu đạt được:**

* Triển khai thành công thuật toán Hough Transform
* Xây dựng giao diện web hiện đại với Streamlit
* Kết quả trực quan, dễ hiểu với 4 view ảnh
* Thời gian xử lý chấp nhận được (1.5-4s)
* Tạo script tự động sinh ảnh minh họa cho báo cáo

**Hướng phát triển:**

* Cải thiện thuật toán để phân biệt đường ngang/dọc chính xác hơn
* Thêm tính năng tự động xoay ảnh về đúng góc
* Hỗ trợ batch processing (xử lý nhiều ảnh cùng lúc)
* Deploy lên Streamlit Cloud để sử dụng online
* Thêm các phương pháp khác: Projection Profile, Deep Learning
* Tích hợp OCR để đọc và xoay tự động văn bản

# 7. TÀI LIỆU/NGUỒN THAM KHẢO

[1] OpenCV Documentation - Hough Line Transform: https://docs.opencv.org/4.x/d9/db0/tutorial\_hough\_lines.html

[2] Streamlit Documentation: https://docs.streamlit.io/

[3] Canny Edge Detection: https://docs.opencv.org/4.x/da/d22/tutorial\_py\_canny.html

[4] NumPy Documentation: https://numpy.org/doc/

[5] Pillow (PIL Fork) Documentation: https://pillow.readthedocs.io/

[6] Computer Vision: Algorithms and Applications - Richard Szeliski

[7] Digital Image Processing - Rafael C. Gonzalez, Richard E. Woods