**TRƯỜNG ĐẠI HỌC VINH**

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**

**Logo

Description automatically generated**

**BÁO CÁO HỌC PHẦN**

**THỊ GIÁC MÁY TÍNH**

**ĐỒ ÁN TRÍCH XUẤT THÔNG TIN THẺ SINH VIÊN**

|  |  |
| --- | --- |
| GVHD: | Th.S. Nguyễn Thị Minh Tâm |
| Nhóm SVTH: | **Nhóm: 06** |
|  | Nguyễn Văn Hiếu, MSV: 215748020110144 |
|  | Nguyễn Thị Ngân, MSV: 215748020110182 |
|  | Phạm Thuỳ Linh, MSV: 215748020110454 |
|  |

**Nghệ An – 202****4**

**MỤC LỤC**

[I. GIỚI THIỆU BÀI TOÁN 2](#_Toc181721570)

[1.1. Giới thiệu bài toán 2](#_Toc181721571)

[1.2. Mục đích đề tài 3](#_Toc181721572)

[II. CÁC THUẬT TOÁN HIỆN NAY 4](#_Toc181721573)

[2.1. Các thuật toán (phương pháp) có trong bài toán 4](#_Toc181721574)

[2.1.1. Xử lý ảnh và phát hiện cạnh với OpenCV 4](#_Toc181721575)

[2.1.2. Nhận diện ký tự bằng EasyOCR 4](#_Toc181721576)

[2.1.3. Trích xuất và cắt vùng ảnh 5](#_Toc181721577)

[2.1.4. Giao diện người dùng (GUI) với Tkinter 5](#_Toc181721578)

[2.1.5. Xử lý lỗi và quản lý tài nguyên 5](#_Toc181721579)

[2.2. Ưu điểm 6](#_Toc181721580)

[2.2.1. Tự động hóa quá trình trích xuất thông tin 6](#_Toc181721581)

[2.2.2. Giao diện thân thiện và dễ sử dụng 6](#_Toc181721582)

[2.2.3. Tích hợp xử lý ảnh và nhận diện ký tự 6](#_Toc181721583)

[III. TRIỂN KHAI 7](#_Toc181721585)

[3.1. Quy trình thực hiện 7](#_Toc181721586)

[3.2. Giao diện chương trình 11](#_Toc181721587)

[IV. KẾT LUẬN 12](#_Toc181721588)

[4.1. Kết quả đạt được 12](#_Toc181721589)

[4.2. Những tồn tại và hạn chế 12](#_Toc181721590)

[4.3. Hướng phát triển 13](#_Toc181721591)

[V. PHỤ LỤC 14](#_Toc181721592)

[5.1. Các bước chạy dự án 14](#_Toc181721593)

[5.2. Video trình bày sản phẩm 16](#_Toc181721594)

# **I. GIỚI THIỆU BÀI TOÁN**

## **1.1. Giới thiệu bài toán**

Trong thời đại 4.0 hiện nay, công nghệ ngày càng phát triển mạnh mẽ, đặt ra nhiều thách thức và yêu cầu mới trong việc quản lý dữ liệu, đặc biệt là dữ liệu sinh viên. Với sự gia tăng nhanh chóng về số lượng sinh viên tại các trường đại học, việc quản lý và trích xuất thông tin của sinh viên trở nên ngày càng phức tạp và quan trọng hơn bao giờ hết. Các phương pháp lưu trữ và quản lý thông tin truyền thống dưới dạng văn bản không chỉ bị giới hạn về dung lượng mà còn gặp khó khăn trong việc xử lý khối lượng dữ liệu lớn, dẫn đến khó khăn trong việc tìm kiếm và cập nhật thông tin. Điều này tạo ra nhu cầu cấp thiết về một giải pháp thông minh, thuận tiện và hiệu quả cho việc quản lý và trích xuất thông tin sinh viên từ thẻ sinh viên.

Bài toán mà chúng tôi đặt ra là làm thế nào để xây dựng một phần mềm có khả năng nhận dạng và trích xuất thông tin từ thẻ sinh viên một cách nhanh chóng, chính xác và tự động. Mục tiêu của chúng tôi không chỉ dừng lại ở việc trích xuất thông tin đơn thuần mà còn hướng tới việc tối ưu hóa quy trình quản lý dữ liệu sinh viên, giúp giảm thiểu thời gian và công sức cho các phòng ban, đồng thời hạn chế các lỗi phát sinh do nhập liệu thủ công.

Chúng tôi đã chọn ứng dụng ý tưởng này vào lĩnh vực quản lý thông tin sinh viên, tận dụng tiềm năng của công nghệ nhận diện được áp dụng rộng rãi trong các lĩnh vực khác như nhận dạng thông tin trên căn cước công dân, hộ chiếu, hay bằng lái xe. Thông qua việc sử dụng phần mềm có khả năng quét và nhận diện các thông tin quan trọng từ thẻ sinh viên như mã số sinh viên, tên, ngày sinh, và các thông tin khác một cách tự động. Đây là một bước đột phá giúp thay thế các phương pháp nhập liệu thủ công, cải thiện đáng kể tốc độ xử lý thông tin, đồng thời đảm bảo độ chính xác và nhất quán của dữ liệu.

### **1.2. Mục đích đề tài**

Mục đích chính của đề tài là xây dựng một phần mềm trích xuất thông tin dành cho sinh viên, nhằm giải quyết những khó khăn hiện có trong việc quản lý và trích xuất dữ liệu sinh viên. Phần mềm này hướng đến việc cung cấp một giải pháp công nghệ hiệu quả, tối ưu hóa quy trình xử lý thông tin và mang lại lợi ích thiết thực cho các tổ chức giáo dục. Cụ thể, phần mềm có các mục tiêu cụ thể như sau:

- Hiệu quả và nhanh chóng: Phần mềm được thiết kế để mang lại hiệu quả cao và thực hiện thao tác một cách nhanh chóng.

- Nhận diện tự động: Một trong những ưu điểm chính của phần mềm là khả năng nhận diện thông tin tự động khi người dùng đưa thẻ sinh viên vào hệ thống. Thông qua việc ứng dụng công nghệ nhận dạng ký tự quang học (OCR), phần mềm có thể tự động quét và nhận dạng thông tin mà không yêu cầu người dùng phải thao tác nhiều.

- Trích xuất thông tin chính xác: Phần mềm đảm bảo khả năng trích xuất chính xác các thông tin quan trọng từ thẻ sinh viên, bao gồm họ tên, mã sinh viên, ngành học, và khóa học.

- Ứng dụng rộng rãi: Phần mềm này không chỉ giới hạn trong phạm vi quản lý thông tin sinh viên mà còn có thể được ứng dụng rộng rãi trong nhiều lĩnh vực khác như tuyển dụng, quản lý nhân sự, và quản lý học sinh, sinh viên ở các cấp học khác nhau.

Mục tiêu của chúng tôi là không chỉ đáp ứng nhu cầu chính của người dùng mà còn tạo ra một phần mềm linh hoạt, dễ mở rộng để đáp ứng các yêu cầu và thách thức trong tương lai. Đồng thời, chúng tôi cam kết giảm thiểu sự cố lỗi gây ra do người sử dụng, tạo ra một trải nghiệm người dùng tối ưu.

# **II. CÁC THUẬT TOÁN HIỆN NAY**

## **2.1. Các thuật toán (phương pháp) có trong bài toán**

### *2.1.1. Xử lý ảnh và phát hiện cạnh với OpenCV*

Gaussian Blur (cv2.GaussianBlur): Đây là phương pháp làm mờ ảnh bằng bộ lọc Gaussian. Phương pháp này giúp giảm nhiễu trong ảnh, làm mượt các chi tiết, và chuẩn bị cho quá trình phát hiện cạnh. Làm mờ ảnh giúp loại bỏ các điểm nhiễu không cần thiết, để các đường viền của đối tượng (ở đây là thẻ sinh viên) trở nên rõ nét hơn.

Canny Edge Detection (cv2.Canny): Thuật toán Canny là một phương pháp phát hiện cạnh phổ biến trong xử lý ảnh. Nó hoạt động qua bốn bước: làm mờ ảnh, tìm gradient của ảnh, áp dụng non-maximum suppression (giữ lại điểm cạnh mạnh nhất), và áp dụng double threshold để xác định các cạnh chắc chắn và cạnh yếu. Kết quả là một ảnh đen trắng, trong đó các cạnh được đánh dấu là các đường trắng.

Contour Detection (cv2.findContours): Sau khi phát hiện cạnh, thuật toán tìm các đường viền (contours) trong ảnh. Các contours này đại diện cho các đối tượng có biên rõ ràng trong ảnh, ở đây là thẻ sinh viên. Sau khi tìm được contour, cv2.boundingRect sẽ tính toán hình chữ nhật bao quanh contour này, giúp xác định vùng chứa thẻ sinh viên.

### *2.1.2. Nhận diện ký tự bằng EasyOCR*

Optical Character Recognition (OCR) : OCR là phương pháp giúp nhận diện và trích xuất văn bản từ ảnh. Trong mã này, easyocr. Reader khởi tạo mô hình nhận dạng ký tự với ngôn ngữ hỗ trợ là tiếng Việt và tiếng Anh. Phương pháp OCR sử dụng mạng nơ-ron sâu để phân tích các ký tự trong từng vùng ảnh và trích xuất văn bản.

Region of Interest (ROI) : Các vùng chứa thông tin như họ tên, ngành, khoa, khóa, và mã số sinh viên (MSSV) được định nghĩa trước với tọa độ cụ thể. Sau khi tìm được vùng chứa thẻ sinh viên, ứng dụng sẽ sử dụng EasyOCR để nhận diện văn bản trong các vùng này.

### *2.1.3. Trích xuất và cắt vùng ảnh*

Bounding Box và Resize: Sau khi tìm được contour lớn nhất (được cho là thẻ sinh viên), chương trình cắt và hiển thị vùng ảnh này. Ảnh được cắt từ thẻ sinh viên sẽ được resize để đảm bảo chất lượng và kích thước ổn định. Điều này hỗ trợ OCR trong việc nhận diện ký tự một cách chính xác hơn.

### *2.1.4. Giao diện người dùng (GUI) với Tkinter*

Canvas để hiển thị video: Tkinter sử dụng Canvas để hiển thị luồng video từ camera và vùng ảnh đã cắt của thẻ sinh viên. Các ảnh này sẽ được cập nhật liên tục để tạo hiệu ứng xem video theo thời gian thực.

Label để hiển thị thông tin trích xuất: Sau khi OCR trích xuất thông tin, các trường thông tin (họ tên, ngành, khoa, khóa, và MSSV) được hiển thị trong các nhãn (Label) trên giao diện Tkinter.

Button điều khiển: Nút bấm được sử dụng để gọi hàm extract\_info, giúp người dùng dễ dàng thao tác để trích xuất thông tin từ thẻ sinh viên.

### *2.1.5. Xử lý lỗi và quản lý tài nguyên*

Kiểm tra và xử lý lỗi: Nếu không có ảnh thẻ sinh viên để trích xuất, chương trình hiển thị thông báo lỗi bằng messagebox. Đây là một phương pháp giúp tránh các lỗi gây ra khi không tìm được thẻ trong ảnh.

Quản lý camera: cap.release() đảm bảo rằng camera được giải phóng khi đóng ứng dụng, tránh việc chiếm dụng tài nguyên không cần thiết.

## **2.2. Ưu điểm**

### *2.2.1. Tự động hóa quá trình trích xuất thông tin*

- Phần mềm giúp tự động hóa việc trích xuất thông tin từ thẻ sinh viên, tiết kiệm thời gian và công sức so với nhập thủ công.

- Giảm thiểu lỗi con người khi nhập liệu và cải thiện độ chính xác trong quản lý thông tin sinh viên.

### *2.2.2. Giao diện thân thiện và dễ sử dụng*

- Giao diện đơn giản, trực quan được xây dựng bằng Tkinter giúp người dùng dễ dàng thao tác, kể cả với những người không am hiểu kỹ thuật.

- Các nhãn và nút điều khiển được sắp xếp hợp lý, dễ nhìn và thân thiện với người dùng.

### *2.2.3. Tích hợp xử lý ảnh và nhận diện ký tự*

- Sử dụng OpenCV để xử lý ảnh giúp chương trình dễ dàng xác định và cắt vùng chứa thẻ sinh viên từ khung hình camera.

- EasyOCR hỗ trợ nhận diện ký tự tiếng Việt, giúp phần mềm trích xuất được thông tin chính xác từ ảnh thẻ sinh viên, đặc biệt với các tên tiếng Việt có dấu.

### *2.2.4. Hiệu quả chi phí và dễ mở rộng*

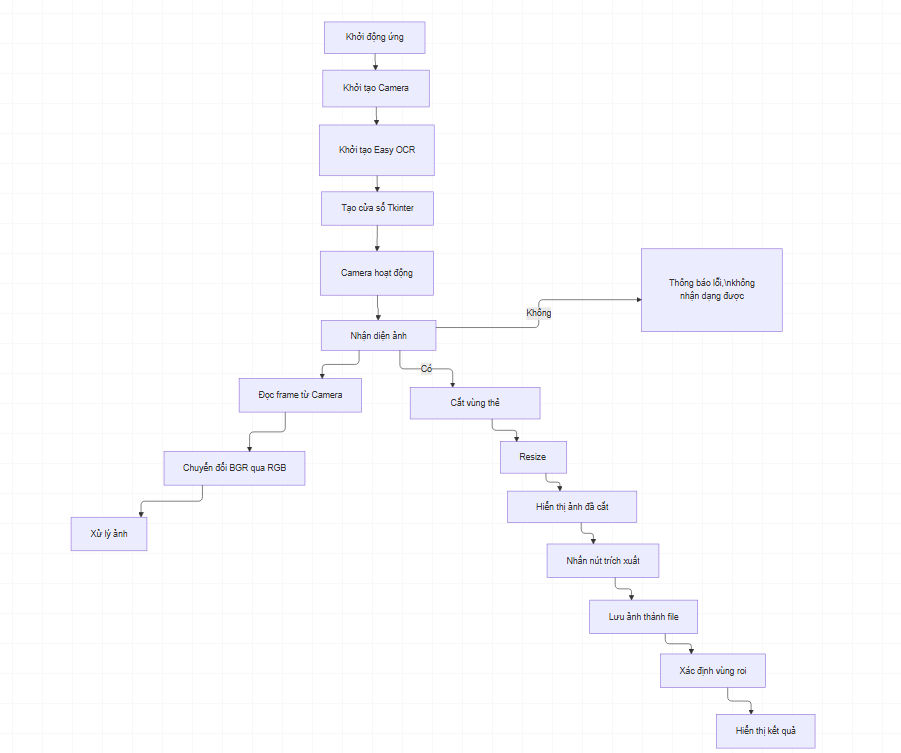
- Phần mềm không yêu cầu phần cứng đặc biệt, có thể chạy trên máy tính có webcam thông thường.

- Code dễ tùy chỉnh và có thể mở rộng để nhận diện các loại thông tin khác trên thẻ sinh viên hoặc ứng dụng cho các loại thẻ khác.

# **III. TRIỂN KHAI**

## **3.1. Quy trình thực hiện**

- Sơ đồ dưới đây tóm tắt các bước truy xuất thông tin thẻ sinh viên :



**Hình 3.1.** Sơ đồ tóm tắt các bước

Đầu tiên, nó khởi tạo các thư viện cần thiết như OpenCV để xử lý hình ảnh, Tkinter để tạo giao diện người dùng, Pillow để xử lý hình ảnh và EasyOCR để nhận diện ký tự. Khi ứng dụng chạy, một cửa sổ giao diện sẽ xuất hiện với các phần hiển thị video từ camera, vùng ảnh cắt từ thẻ sinh viên và các nhãn để hiển thị thông tin trích xuất. Camera sẽ liên tục ghi lại hình ảnh và hiển thị trên giao diện. Khi người dùng đặt thẻ sinh viên trước camera, ứng dụng sẽ tự động tìm và cắt vùng chứa thẻ sinh viên. Sau đó, người dùng nhấn nút “Trích xuất thông tin” để bắt đầu quá trình nhận diện ký tự từ vùng ảnh cắt. EasyOCR sẽ đọc các ký tự từ ảnh và hiển thị thông tin như họ tên, ngành, khoa, khóa và mã số sinh viên trên giao diện. Nếu không tìm thấy ảnh sinh viên hoặc có lỗi, ứng dụng sẽ hiển thị thông báo lỗi. Khi người dùng đóng cửa sổ, ứng dụng sẽ giải phóng camera và kết thúc.

**A. Lọc nhiễu, chuyển ảnh xám**

Đầu tiên, ảnh từ camera được làm mờ bằng bộ lọc Gaussian Blur, một kỹ thuật làm mịn ảnh giúp giảm nhiễu và loại bỏ các chi tiết nhỏ không cần thiết. Cụ thể, đoạn mã sử dụng hàm cv2.GaussianBlur với kích thước kernel là (5, 5) để làm mờ ảnh, giúp làm mịn các cạnh và chi tiết nhỏ trong ảnh. Sau khi làm mờ, ảnh được chuyển sang ảnh xám bằng cách sử dụng phương pháp phát hiện cạnh Canny. Hàm cv2.Canny được sử dụng để phát hiện các cạnh trong ảnh, với các ngưỡng là 75 và 150. Quá trình này giúp xác định rõ ràng các đường biên của thẻ sinh viên, từ đó dễ dàng tìm và cắt vùng chứa thẻ. Việc kết hợp giữa làm mờ ảnh và phát hiện cạnh giúp tăng độ chính xác trong việc nhận diện và trích xuất thông tin từ thẻ sinh viên.

**B. Canny**

Đầu tiên, ảnh từ camera được làm mờ bằng bộ lọc Gaussian Blur với kích thước kernel là (5, 5). Quá trình này giúp giảm nhiễu và làm mịn các chi tiết nhỏ trong ảnh, tạo điều kiện thuận lợi cho việc phát hiện cạnh. Sau khi làm mờ, ảnh được chuyển sang ảnh xám và áp dụng phương pháp phát hiện cạnh Canny. Phương pháp này sử dụng hai ngưỡng (75 và 150) để xác định các cạnh yếu và mạnh trong ảnh. Các cạnh mạnh được giữ lại, trong khi các cạnh yếu chỉ được giữ lại nếu chúng kết nối với các cạnh mạnh, tạo ra một ảnh nhị phân với các cạnh được phát hiện rõ ràng. Tiếp theo, các contour trong ảnh nhị phân được tìm kiếm bằng hàm cv2.findContours, giúp xác định các đường biên của các đối tượng trong ảnh, bao gồm cả thẻ sinh viên.

Đoạn mã duyệt qua các contour và sử dụng hàm cv2.boundingRect để xác định hình chữ nhật bao quanh mỗi contour. Nếu kích thước của hình chữ nhật đủ lớn (rộng và cao đều lớn hơn 100), vùng chứa thẻ sinh viên sẽ được cắt và hiển thị trên giao diện. Quá trình này đảm bảo rằng chỉ những vùng có kích thước phù hợp mới được chọn, giúp tăng độ chính xác trong việc nhận diện và trích xuất thông tin từ thẻ sinh viên. Phương pháp phát hiện cạnh Canny, kết hợp với các bước xử lý ảnh khác, tạo nên một quy trình mạnh mẽ và hiệu quả để xử lý và trích xuất thông tin từ hình ảnh.

**C. Xử lý ảnh để tìm contour của thẻ sinh viên**

Phát hiện contour là một phương pháp quan trọng trong xử lý ảnh để xác định và biểu diễn các đường biên của đối tượng trong ảnh. Quá trình phát hiện contour:Sử dụng hàm như cv2.findContours() trong OpenCV để tìm các contour trong ảnh. Hàm này trả về danh sách các contour và một biến hierarchy (cấu trúc của contour). Vẽ contour: Sau khi có danh sách contour, chúng tôi có thể vẽ mọi thứ lên ảnh gốc bằng cách sử dụng hàm như cv2.drawContours().

**D. Trích xuất thông tin**

Quá trình trích xuất thông tin từ thẻ sinh viên trong đoạn mã của bạn diễn ra qua một chuỗi các bước cụ thể. Khi người dùng nhấn nút “Trích xuất thông tin”, hàm extract\_info được gọi để kiểm tra sự tồn tại của vùng ảnh chứa thẻ sinh viên. Nếu vùng ảnh tồn tại, ảnh sẽ được lưu thành file “oopp.jpg”. Tiếp theo, các vùng quan tâm (ROI) cho từng trường thông tin trên thẻ sinh viên được định nghĩa trước, bao gồm họ tên, ngành, khoa, khóa và mã số sinh viên. Mỗi vùng được xác định bằng tọa độ y1, y2, x1, x2. Hàm extract\_info sau đó duyệt qua từng vùng quan tâm, cắt vùng ảnh tương ứng và sử dụng EasyOCR để nhận diện ký tự. Kết quả nhận diện được ghép lại thành chuỗi văn bản và hiển thị trên các nhãn tương ứng trong giao diện Tkinter. Nếu vùng ảnh cắt vượt quá giới hạn của ảnh gốc, quá trình trích xuất cho vùng đó sẽ bị bỏ qua. Thông tin nhận diện được hiển thị rõ ràng trên giao diện, giúp người dùng dễ dàng theo dõi kết quả. Quá trình này đảm bảo rằng thông tin từ thẻ sinh viên được trích xuất một cách chính xác và hiệu quả.

**Cách thức hoạt động:**

EasyOCR hoạt động theo các bước sau:

- Chuyển đổi ảnh và tiền xử lý: EasyOCR sử dụng các kỹ thuật xử lý ảnh để làm nổi bật các chi tiết văn bản trong ảnh đầu vào. Quá trình này có thể bao gồm chuyển đổi ảnh sang ảnh xám và làm mờ ảnh để giảm nhiễu.

- Phát hiện và trích xuất khối văn bản: Hệ thống sử dụng mạng nơ-ron để phát hiện các khối văn bản, xác định vị trí của chúng trong ảnh và phân biệt văn bản với các phần không liên quan.

- Nhận diện ký tự: Sau khi phát hiện các vùng văn bản, EasyOCR áp dụng các mạng nơ-ron sâu (DNN) được huấn luyện trên nhiều ngôn ngữ và ký tự để nhận diện từng ký tự trong các vùng này.

- Ghép ký tự thành từ và câu: EasyOCR ghép các ký tự đã nhận diện để tạo thành từ và câu hoàn chỉnh. Nó cũng tích hợp các thuật toán xử lý ngôn ngữ tự nhiên để cải thiện độ chính xác trong việc ghép từ.

- Xuất văn bản: Văn bản nhận diện từ ảnh được xuất ra dưới dạng văn bản số, sẵn sàng để lưu trữ, phân tích, hoặc tích hợp vào các ứng dụng khác.

**E. Thiết kế giao diện**

Thiết kế giao diện của ứng dụng trích xuất thông tin từ thẻ sinh viên sử dụng Tkinter để tạo một cửa sổ đồ họa thân thiện và dễ sử dụng.

Tkinter, viết tắt của "Tk interface", là tiêu chuẩn của Python để xây dựng GUI và được bao gồm trong Thư viện tiêu chuẩn Python. Nó là một ràng buộc với bộ công cụ Tk GUI, một thư viện mã nguồn mở miễn phí của các tiện ích GUI có thể được sử dụng để xây dựng giao diện đồ họa bằng nhiều ngôn ngữ lập trình.

## **3.2. Giao diện chương trình**

****

**Hình 3.2.** Giao diện khi chạy ứng dụng

# **IV. KẾT LUẬN**

## **4.1. Kết quả đạt được**

Chúng tôi đã thành công trong việc xây dựng một hệ thống thông tin giúp trích xuất thông tin từ thẻ sinh viên một cách hiệu quả. Qua quá trình triển khai, chúng tôi đã thực hiện một chu trình rõ ràng, từ phân tích và thiết kế đến xây dựng hệ thống, đặt vai trò và vị trí mỗi yếu tố trong quá trình này.

Tiến trình phân tích và thiết kế hệ thống của chúng tôi đã đảm bảo tính toàn vẹn và hiệu quả của quá trình trích xuất thông tin. Chúng tôi đã không chỉ trình bày kết quả mà còn thảo luận về vai trò và vị trí của hệ thống trong môi trường thông tin phức tạp ngày nay.

Chúng tôi cũng đã phát triển kỹ năng giao tiếp, phối hợp và giải quyết vấn đề kỹ thuật trong quá trình làm việc nhóm. Điều này là quan trọng để đảm bảo sự hiệu quả của dự án và sự hài lòng của người sử dụng cuối cùng.

## **4.2. Những tồn tại và hạn chế**

Mặc dù đã đạt được những thành công, nhưng chúng tôi nhận ra rằng phạm vi của chúng tôi còn hạn chế và chưa bao quát được toàn bộ các khía cạnh của đề tài. Điều này mở ra cơ hội để nghiên cứu và mở rộng hệ thống trong tương lai.

Chúng tôi cũng thừa nhận rằng còn thiếu kinh nghiệm trong việc phân tích nghiệp vụ. Điều này là một bài học quý báu, và chúng tôi sẽ áp dụng kiến thức này để cải thiện quy trình làm việc của chúng tôi và đảm bảo tính toàn vẹn của thông tin trích xuất.

## **4.3. Hướng phát triển**

Chúng tôi cam kết tiếp tục phát triển và cải thiện hệ thống trích xuất thông tin từ thẻ sinh viên. Cùng với sự phát triển của công nghệ, chúng tôi sẽ liên tục nghiên cứu và tích hợp những tiến bộ mới vào hệ thống của mình.

Hướng phát triển chủ yếu tập trung vào việc mở rộng phạm vi ứng dụng và tối ưu hóa thuật toán nhận diện. Chúng tôi sẽ liên tục thu thập thông tin và đánh giá hiệu suất để đảm bảo rằng hệ thống của chúng tôi có thể hoạt động chính xác và hiệu quả trong mọi điều kiện.

Chúng tôi hy vọng rằng hệ thống trích xuất thông tin này sẽ đóng góp tích cực vào việc quản lý thông tin sinh viên và có ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau.

# **V. PHỤ LỤC**

## **5.1. Các bước chạy dự án**

- Cài đặt các thư viện: Open CV, Tkinter, Pillow (PIL), EasyOCR.

- Thực hiện chạy phần mã nguồn sau:

A computer screen shot of text

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

## **5.2. Video trình bày sản phẩm**

[**https://www.youtube.com/watch?v=1BjC9t70ZVY**](https://www.youtube.com/watch?v=1BjC9t70ZVY&fbclid=IwZXh0bgNhZW0CMTAAAR11sNb1KHKWRgDHmi16LuZVlyQGmA-aH4hs4-NyyGz0_O1q9-fLU6yNGp0_aem_6UosYI9F0iShtS-OSEjpFQ)