**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**

**VIỆN ĐÀO TẠO CHẤT LƯỢNG CAO**



**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN : XỬ LÝ ẢNH**

**XÂY DỰNG CHƯƠNG TRÌNH NHẬN DẠNG BIỂN SỐ XE**

**Ô TÔ ĐƠN GIẢN.**

**Họ và tên: Nguyễn Tuấn Minh**

**Mã sinh viên : 90504**

**GIẢNG VIÊN HƯỚNG DẪN: Ts.NGUYỄN HỮU TUÂN**

**HẢI PHÒNG 2024**

Mục lục

[LỜI MỞ ĐẦU 3](#_Toc165903656)

[CHƯƠNG I: GIỚI THIỆU 4](#_Toc165903657)

[1.1. Tổng quan về xử lý ảnh 4](#_Toc165903658)

[1.1.1. Quá trình xử lý ảnh 5](#_Toc165903659)

[1.1.2. Phạm vi ứng dụng xử lý ảnh 8](#_Toc165903660)

[1.2. Mục đích của đề tài 9](#_Toc165903661)

[1.3. Phương hướng xây dựng 9](#_Toc165903662)

[1.4. Kết quả dự kiến 9](#_Toc165903663)

[CHƯƠNG II :CHƯƠNG TRÌNH NHẬN DIỆN BIỂN SỐ XE ĐƠN GIẢN 11](#_Toc165903664)

[2.1. Tổng quan bài toán 11](#_Toc165903665)

[2.2. Yêu cầu và thách thức của chương trình 11](#_Toc165903666)

[2.3. Phương pháp thực hiện và các công nghệ hỗ trợ 12](#_Toc165903667)

[CHƯƠNG III : CÀI ĐẶT VÀ THỬ NGHIỆM 14](#_Toc165903668)

[3.1.Công cụ sử dụng 14](#_Toc165903669)

[3.1.1.Cách cài đặt Pycharm 15](#_Toc165903670)

[3.1.2. Thư viện sử dụng đi kèm 15](#_Toc165903671)

[3.2. Viết chương trình 17](#_Toc165903672)

3.2.1: Sơ đồ hệ thống ................................................................................ 23

3.2.2: Chạy chương trình .......................................................................... 23

Kết Luận ........................................................................................................... 25

Tài Liệu Tham Khảo ......................................................................................... 25

# LỜI MỞ ĐẦU

Trải qua hơn một thế kỷ, từ thời điểm ảnh số xuất hiện vào thập kỷ đầu của thế kỷ 19, lĩnh vực xử lý ảnh đã trải qua những bước tiến vượt bậc. Đã có những đóng góp to lớn cho đời sống hàng ngày từ các phương diện khác nhau. Trong thời đại hiện đại ngày nay, với sự tiến bộ không ngừng của công nghệ và máy tính, chúng ta đang chứng kiến một thời kỳ mới của sự tiện lợi và hiệu suất cao trong việc xử lý ảnh.

Một trong những ứng dụng quan trọng nhất của lĩnh vực này là việc nhận dạng biển số xe. Từ việc quản lý giao thông đến giám sát an ninh, khả năng này đóng vai trò không thể phủ nhận trong việc tăng cường sự an toàn và quản lý hiệu quả. Với mục tiêu này, em xin giới thiệu chương trình nhận dạng biển số xe đơn giản của mình.

Bằng việc kết hợp các thuật toán xử lý ảnh tiên tiến, chương trình này có khả năng nhận dạng và phân tích các biển số xe một cách nhanh chóng và chính xác. Mọi người đều có thể tiếp cận và sử dụng chương trình này với Pycharm, từ các cơ quan chính phủ đến doanh nghiệp và cá nhân chỉ bằng việc thay thế đường dẫn ảnh ở trong chương trình.

Không chỉ giúp tiết kiệm thời gian và công sức, chương trình nhận dạng biển số xe còn mang lại sự linh hoạt và chính xác trong việc xử lý dữ liệu. Em cam kết mang đến giải pháp hiệu quả và tiện ích cho người dùng. Em tin rằng chương trình nhận dạng biển số xe đơn giản của em sẽ trở thành một công cụ tuyệt vời và không thể thiếu trong giao thông hàng ngày. Hãy cùng em khám phá và trải nghiệm sức mạnh của công nghệ xử lý ảnh ngày nay!

### 

# CHƯƠNG I: GIỚI THIỆU

## 1.1. Tổng quan về xử lý ảnh

Xử lý ảnh (tiếng Anh: digital image processing) hoặc xử lý ảnh kỹ thuật số là việc sử dụng máy tính số để xử lý các ảnh kỹ thuật số thông qua các thuật toán. Là một phân ngành của xử lý tín hiệu số, xử lý ảnh kỹ thuật số có nhiều lợi thế so với xử lý ảnh analog. Nó cho phép áp dụng nhiều thuật toán hơn cho dữ liệu đầu vào và có thể tránh được các vấn đề như nhiễu và méo hình trong quá trình xử lý. Vì ảnh được định nghĩa trên hai chiều (hoặc nhiều hơn), xử lý ảnh kỹ thuật số có thể được mô hình hóa dưới dạng hệ thống đa chiều. Sự phát triển của xử lý ảnh kỹ thuật số chủ yếu bị ảnh hưởng bởi ba yếu tố: đầu tiên là sự phát triển của máy tính; thứ hai là sự phát triển của toán học (đặc biệt là sáng tạo và cải tiến lý thuyết toán rời rạc); thứ ba là nhu cầu ứng dụng rộng rãi trong môi trường, nông nghiệp, quân sự, công nghiệp và khoa học y tế ngày càng tăng.

Xử lý ảnh đóng vai trò quan trọng trong lĩnh vực thị giác máy tính, nơi mà em tập trung vào việc chuyển đổi hình ảnh từ dạng ban đầu thành dạng mới, với các thuộc tính có thể được điều chỉnh theo mong muốn của người dùng. Quá trình này bao gồm nhiều phương pháp như phân tích, phân loại các đối tượng trong hình ảnh, cải thiện chất lượng, phân đoạn và tách cạnh, cũng như gắn nhãn cho các vùng quan trọng và dịch thông tin từ hình ảnh.

Tương tự như việc xử lý dữ liệu thông qua đồ họa, xử lý ảnh số là một lĩnh vực quan trọng của tin học ứng dụng. Nó tập trung vào việc xử lý ảnh tự nhiên và các phương pháp biến đổi dữ liệu để truyền tải hoặc mã hóa hình ảnh. Mục tiêu của xử lý ảnh bao gồm:

Nâng cao chất lượng hình ảnh: Cải thiện độ rõ nét, tăng độ phân giải, giảm nhiễu và cải thiện các yếu tố khác để tạo ra hình ảnh rõ ràng và dễ đọc hơn.

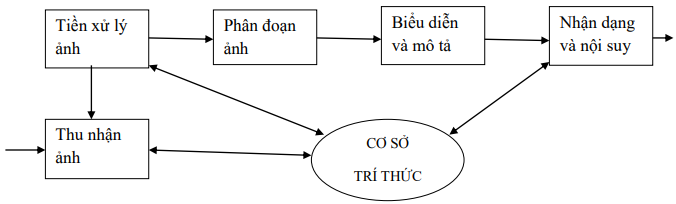
Tự động nhận dạng và đánh giá: Phân tích hình ảnh để nhận dạng đối tượng, dự đoán nội dung và đánh giá các yếu tố quan trọng trong hình ảnh.

Công nghệ xử lý ảnh đóng vai trò quan trọng trong nhiều lĩnh vực như y học (trong việc xử lý hình ảnh y khoa, nhận diện các cấu trúc tế bào), nhận dạng ký tự trong văn bản và nhiều ứng dụng khác đòi hỏi khả năng phân tích và xử lý hình ảnh một cách hiệu quả.

Trong dự án này, em sẽ tập trung vào việc áp dụng các kỹ thuật và phương pháp cơ bản của xử lý ảnh để xây dựng một chương trình nhận diện biển số xe. Mục tiêu của em là sử dụng các thuật toán xử lý ảnh để tự động phát hiện và nhận dạng biển số xe từ các bức ảnh hoặc video.

Chương trình này sẽ không chỉ giúp tăng cường khả năng tự động hóa trong việc nhận diện biển số xe mà còn mở ra nhiều tiềm năng trong việc ứng dụng xử lý ảnh vào các lĩnh vực thực tế khác như giao thông, an ninh, và quản lý đô thị.

### 1.1.1. Quá trình xử lý ảnh



Hình 1: Các quá trình xử lý ảnh

* Thu nhận ảnh:

Công đoạn đầu tiên và quan trọng nhất trong quá trình xử lý ảnh là thu nhận ảnh từ các thiết bị như máy ảnh màu, máy ảnh trắng đen, máy quét ảnh, hoặc máy quay. Chất lượng của một bức ảnh thu được phụ thuộc vào thiết bị thu và điều kiện môi trường như ánh sáng và phong cảnh. Sau khi thu nhận, ảnh được chuyển đổi từ tín hiệu analog sang tín hiệu số thông qua quá trình chuyển đổi ADC (Analog to Digital Converter). Quá trình này chính là quá trình số hóa ảnh, giúp biến ảnh từ dạng analog thành dạng số để có thể được xử lý bởi các thiết bị và phần mềm máy tính.

* Tiền xử lý:

Quá trình tiền xử lý ảnh đóng vai trò quan trọng trong việc cải thiện chất lượng của hình ảnh. Với khả năng ảnh bị nhiễu và có độ tương phản thấp, bước này trở nên cực kỳ cần thiết để tối ưu hóa ảnh trước khi thực hiện các xử lý phức tạp hơn.

Chức năng chính của bộ tiền xử lý ảnh là loại bỏ nhiễu và nâng cao độ tương phản để làm cho ảnh trở nên rõ ràng và nét hơn. Các công đoạn trong quá trình này bao gồm cải thiện độ tương phản, khử nhiễu, khôi phục ảnh, và điều chỉnh các yếu tố khác như khử bóng và khử độ lệch. Mục tiêu của bước tiền xử lý là chuẩn bị cho các bước xử lý phức tạp hơn trong quá trình xử lý ảnh.

Thường thì, các bộ lọc được sử dụng để thực hiện các công việc này, giúp làm sáng tỏ và làm nổi bật các đặc điểm quan trọng trong ảnh trước khi đưa chúng vào các giai đoạn xử lý tiếp theo.

* Phân đoạn ảnh:

Phân đoạn ảnh là giai đoạn then chốt trong quy trình xử lý ảnh, nơi ảnh được chia thành các thành phần có cùng tính chất dựa trên biên hoặc các vùng liên thông. Các tiêu chuẩn để xác định các vùng liên thông có thể bao gồm cùng màu sắc, cùng mức xám, và các yếu tố khác. Mục đích của quá trình này là tạo ra một miêu tả tổng hợp về các phần tử cấu tạo lên ảnh, giúp giảm lượng thông tin và tạo ra các đặc trưng cần thiết cho các bước xử lý tiếp theo.

Phân đoạn ảnh là việc tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích và nhận dạng ảnh. Ví dụ, để nhận dạng chữ trên một phong bì thư, cần chia các câu, từ, chữ thành các phần riêng biệt để nhận dạng. Đây thường là phần phức tạp và khó khăn nhất trong xử lý ảnh, có thể gây ra các lỗi và ảnh hưởng đến độ chính xác của quá trình nhận dạng. Kết quả của quá trình phân đoạn ảnh thường là các dữ liệu điểm ảnh thô hoặc hàm chứa biên của một vùng ảnh.

* Biểu diễn và mô tả

Quá trình biểu diễn và mô tả ảnh trong xử lý ảnh là quá trình quan trọng để chuyển đổi thông tin hình ảnh thành dạng có thể được sử dụng và hiểu bởi máy tính. Nó bao gồm hai phần chính: biểu diễn và mô tả.

**Biểu diễn ảnh:**

Trong quá trình này, ảnh được biểu diễn dưới dạng số để máy tính có thể hiểu được. Phương pháp phổ biến nhất là biểu diễn ảnh dưới dạng ma trận số, trong đó mỗi phần tử của ma trận tương ứng với một pixel trong ảnh.

Mỗi pixel có thể được biểu diễn bằng giá trị mức xám (cho ảnh xám) hoặc bằng ba giá trị mức xám (cho ảnh màu), tương ứng với các thành phần màu RGB (đỏ, xanh lá cây, và lam).

Ngoài ra, có các phương pháp biểu diễn khác như Fourier, Wavelet, hoặc biểu diễn vector.

**Mô tả ảnh:**

Sau khi ảnh đã được biểu diễn dưới dạng số, quá trình mô tả ảnh giúp trích xuất thông tin quan trọng và đặc trưng từ ảnh.

Điều này có thể bao gồm việc tìm kiếm cạnh, phát hiện điểm đặc trưng, hay mô tả đặc điểm hình học của các đối tượng trong ảnh.

Mô tả ảnh thường sử dụng các kỹ thuật như bộ lọc, biến đổi, và các thuật toán máy học để nhận diện và trích xuất các đặc trưng quan trọng.

Quá trình biểu diễn và mô tả ảnh là bước quan trọng trong xử lý ảnh, đặc biệt trong các ứng dụng như nhận dạng ảnh, phân loại, và phát hiện đối tượng. Đảm bảo rằng biểu diễn và mô tả chính xác của ảnh sẽ là cơ sở cho các bước xử lý và phân tích ảnh tiếp theo.

* Nhận dạng và nội suy:

Quá trình nhận dạng ảnh là quá trình xác định và phân loại các đối tượng trong ảnh, thường được thực hiện bằng cách so sánh với các mẫu chuẩn đã được lưu trữ trước đó. Nội suy là phần của quá trình này, thực hiện dự đoán ý nghĩa của các đối tượng đã nhận dạng dựa trên ngữ cảnh hoặc thông tin có sẵn.

Có hai phương pháp cơ bản trong quá trình nhận dạng ảnh:

Nhận dạng theo tham số: Xác định các đối tượng dựa trên các thuộc tính và đặc điểm cụ thể của chúng.

Nhận dạng theo cấu trúc: Xác định các đối tượng dựa trên cấu trúc tổ chức hoặc mối quan hệ giữa các thành phần trong ảnh.

Các đối tượng phổ biến được nhận dạng hiện nay bao gồm ký tự (in hoặc viết tay), văn bản, vân tay, mã vạch, và mặt người. Quá trình nhận dạng và nội suy ảnh là bước cuối cùng trong quá trình xử lý ảnh, nhằm gán nhãn và giải thích ý nghĩa của các đối tượng đã được nhận dạng. Điều này thường được áp dụng trong các hệ thống tự động hoặc bán tự động để rút trích thông tin quan trọng từ ảnh.

### 1.1.2. Phạm vi ứng dụng xử lý ảnh

Trong thời đại công nghệ hiện đại, xử lý ảnh không chỉ là một lĩnh vực kỹ thuật mà đã trở thành một phần quan trọng trong cuộc sống hàng ngày cũng như trong nhiều lĩnh vực khoa học và công nghiệp. Sự phát triển của công nghệ đã mở ra một loạt các ứng dụng đa dạng cho xử lý ảnh, từ việc cải thiện ảnh cá nhân đến ứng dụng trong y học, robot học, an ninh, và nhiều lĩnh vực khác. Bài luận này sẽ thảo luận về phạm vi ứng dụng của xử lý ảnh và tiềm năng của nó.

Một trong những phạm vi ứng dụng chính của xử lý ảnh là trong lĩnh vực y học. Các hệ thống xử lý ảnh được sử dụng để chẩn đoán bệnh lý từ hình ảnh y khoa như tia X, MRI, và siêu âm. Chúng giúp các chuyên gia y tế nhận dạng các bất thường và vấn đề sức khỏe từ hình ảnh, từ đó cải thiện quá trình chẩn đoán và điều trị. Ngoài ra, xử lý ảnh cũng được áp dụng trong việc phục hồi và cải thiện chất lượng hình ảnh y học, giúp tạo ra các hình ảnh rõ ràng và chi tiết hơn.

Trong lĩnh vực robot học và tự động hóa, xử lý ảnh chơi một vai trò quan trọng trong việc giúp robot nhận dạng và tương tác với môi trường xung quanh. Hệ thống thị giác máy tính được sử dụng để nhận diện và phản ứng với các đối tượng và tình huống trong thời gian thực. Điều này giúp robot thực hiện các nhiệm vụ tự động hoặc hợp tác với con người trong nhiều ứng dụng, từ sản xuất công nghiệp đến dịch vụ khách hàng.

Trong lĩnh vực an ninh và giám sát, xử lý ảnh được sử dụng để phát hiện và nhận dạng các đối tượng và hành vi không mong muốn. Hệ thống giám sát sử dụng xử lý ảnh để nhận diện và theo dõi các đối tượng trong thời gian thực, từ người đi lại đến hành vi xâm nhập. Điều này giúp cải thiện an ninh và quản lý rủi ro trong các khu vực quan trọng như công cộng, doanh nghiệp, và nhà máy.

Ngoài ra, xử lý ảnh cũng có nhiều ứng dụng trong việc phát triển ứng dụng và trò chơi điện tử. Thị giác máy tính được sử dụng để tạo ra trải nghiệm thú vị và tương tác trong thế giới ảo hoặc thực tế tăng cường. Các tính năng như nhận diện cử động và nhận diện khuôn mặt được tích hợp để cải thiện trải nghiệm người dùng trong các ứng dụng di động và trò chơi điện tử.

Trong tương lai, xử lý ảnh có tiềm năng phát triển mạnh mẽ hơn trong nhiều lĩnh vực khác nhau. Với sự tiến bộ của công nghệ, các ứng dụng mới có thể được phát triển để đáp ứng nhu cầu ngày càng tăng của xã hội và công nghiệp. Với sự đa dạng và tiềm năng không giới hạn, xử lý ảnh đang trở thành một lĩnh vực quan trọng và hứa hẹn cho tương lai.

## 1.2. Mục đích của đề tài

Mục tiêu của đề tài về "Chương trình Nhận diện Biển số Xe Đơn giản" không chỉ nằm trong tầm nhìn mở rộng của nghiên cứu về trí tuệ nhân tạo và thị giác máy tính mà còn là sự hướng đến một giải pháp thực tiễn và linh hoạt. Nghiên cứu này tập trung vào việc phát triển một hệ thống tự động có khả năng nhận diện biển số xe và phân tích thông tin liên quan, nhằm ứng dụng rộng rãi trong các lĩnh vực như giao thông, an ninh và quản lý đô thị.

Bằng cách kết hợp các phương pháp tiên tiến trong xử lý ảnh và máy học, đề tài này đặt ra mục tiêu tối ưu hóa quá trình nhận diện biển số xe, giảm thiểu thời gian và công sức cần thiết cho việc thu thập dữ liệu, đồng thời tăng cường độ chính xác và hiệu suất của quy trình này.

Mục tiêu cụ thể là xây dựng một hệ thống linh hoạt và có khả năng thích ứng với nhiều điều kiện ánh sáng và môi trường khác nhau, từ đó cung cấp thông tin chính xác và đáng tin cậy về các phương tiện di chuyển. Điều này không chỉ hỗ trợ cho việc quản lý giao thông và an ninh mà còn góp phần vào sự phát triển bền vững của các đô thị hiện đại, thông qua việc áp dụng công nghệ vào các lĩnh vực quan trọng như giao thông và an ninh đô thị.

## 1.3. Phương hướng xây dựng

Thu thập dữ liệu: Bước đầu tiên là thu thập dữ liệu ảnh chứa biển số xe từ nhiều nguồn khác nhau. Dữ liệu này sẽ bao gồm các điều kiện ánh sáng, góc chụp và môi trường đa dạng để đảm bảo tính linh hoạt và đa dạng của hệ thống.

Tiền xử lý dữ liệu: Dữ liệu thu thập được sẽ được tiền xử lý để loại bỏ nhiễu, cân bằng ánh sáng và tăng cường chất lượng hình ảnh. Điều này sẽ giúp tăng độ chính xác của quá trình nhận diện sau này.

Phát triển mô hình nhận diện: Sử dụng các kỹ thuật xử lý ảnh em sẽ xây dựng một mô hình có khả năng nhận diện và phân tích biển số xe từ ảnh.

## 1.4. Kết quả dự kiến

Kết quả dự kiến của đề tài này là xây dựng thành công một hệ thống nhận diện biển số xe đơn giản nhưng mạnh mẽ và hiệu quả. Dưới đây là những kết quả mà em mong đợi:

**Độ chính xác cao:** Hệ thống nhận diện biển số xe được kỳ vọng có độ chính xác cao, đảm bảo rằng biển số xe được nhận diện và phân tích chính xác trong nhiều điều kiện ánh sáng và môi trường khác nhau.

**Tính linh hoạt:** Hệ thống được thiết kế để có khả năng thích ứng với nhiều loại biển số xe từ nhiều quốc gia khác nhau, đồng thời hoạt động ổn định và hiệu quả trên các nền tảng và môi trường thực tế.

**Tính ổn định:** Hệ thống sẽ được kiểm tra và đảm bảo tính ổn định, không bị ảnh hưởng quá nhiều bởi nhiễu hoặc biến đổi về điều kiện môi trường.

**Tính thời gian thực:** Hệ thống sẽ hoạt động trong thời gian thực, đảm bảo rằng việc nhận diện và phân tích biển số xe diễn ra một cách nhanh chóng và hiệu quả.

**Độ tin cậy:** Cuối cùng, kết quả mong đợi là một hệ thống nhận diện biển số xe tin cậy và đáng tin cậy, giúp cải thiện quản lý giao thông, an ninh và các ứng dụng khác trong thực tế.

Với những kết quả này, đề tài sẽ đóng góp vào sự tiến bộ của lĩnh vực trí tuệ nhân tạo và thị giác máy tính, đồng thời cung cấp các giải pháp hiệu quả cho các vấn đề thực tiễn trong xã hội.

**CHƯƠNG II :CHƯƠNG TRÌNH NHẬN DIỆN BIỂN SỐ XE ĐƠN GIẢN**

**2.1. Tổng quan bài toán**

Bài toán nhận diện biển số xe đóng vai trò không thể phủ nhận trong nhiều lĩnh vực ứng dụng thực tiễn như giao thông, an ninh và quản lý đô thị. Ý nghĩa của việc nhận diện biển số xe là cung cấp thông tin về các phương tiện di chuyển, từ đó hỗ trợ quản lý và giám sát hiệu quả trong các hệ thống giao thông, an ninh và quản lý đô thị.

Trong lĩnh vực giao thông, nhận diện biển số xe giúp theo dõi và kiểm soát luồng xe, hỗ trợ việc quản lý tuyến đường, điều chỉnh giao thông và dự báo tình trạng kẹt xe. Nó cũng hỗ trợ trong việc xác định vi phạm giao thông và áp dụng các biện pháp quản lý an toàn giao thông.

Trong lĩnh vực an ninh, nhận diện biển số xe có thể giúp phát hiện và theo dõi các phương tiện đối tượng, từ việc giám sát biên giới đến giảm tội phạm trên đường phố. Nó cũng có thể được sử dụng để phát hiện và ngăn chặn các hoạt động tội phạm như trộm cắp xe hoặc vận chuyển hàng hóa bất hợp pháp.

Trong lĩnh vực quản lý đô thị, thông tin từ việc nhận diện biển số xe có thể được sử dụng để quản lý các dịch vụ công cộng như đậu xe, thu phí tự động và quản lý vận tải công cộng. Nó cũng có thể hỗ trợ trong việc nghiên cứu và đánh giá các chính sách và dự án đô thị, từ việc phân tích tải trọng giao thông đến quản lý ô nhiễm không khí và tiếng ồn.

Tóm lại, việc nhận diện biển số xe đóng vai trò quan trọng và không thể thiếu trong việc cải thiện quản lý và an ninh trong các hệ thống giao thông, an ninh và quản lý đô thị.

## 2.2. Yêu cầu và thách thức của chương trình

Trong việc nhận diện biển số xe, có một số yêu cầu và thách thức đáng chú ý cần được xem xét. Đầu tiên, điều kiện ánh sáng đa dạng là một thách thức lớn. Biển số xe có thể được chụp trong điều kiện ánh sáng ban ngày, ánh sáng mờ, hoặc thậm chí là ánh sáng yếu trong điều kiện thiên nhiên khó khăn. Điều này đòi hỏi hệ thống nhận diện phải có khả năng thích ứng với các điều kiện ánh sáng khác nhau để đảm bảo độ chính xác.

Thách thức tiếp theo là góc chụp và biến dạng của biển số xe. Các hình ảnh có thể được chụp từ nhiều góc độ và khoảng cách khác nhau, từ đó tạo ra biến dạng và biến thể trong hình dạng của biển số xe. Điều này đặt ra yêu cầu về tính linh hoạt và độ chính xác cao cho hệ thống nhận diện biển số xe.

Cuối cùng, nhiễu trong hình ảnh cũng là một thách thức đáng kể. Nhiễu có thể bao gồm các yếu tố như mờ, độ nhiễu, hoặc các vật thể khác che khuất phần của biển số xe. Hệ thống nhận diện cần có khả năng phát hiện và loại bỏ nhiễu để đảm bảo độ chính xác trong việc nhận diện.

Tổng cộng, việc đối mặt với các yêu cầu và thách thức này yêu cầu hệ thống nhận diện biển số xe phải có tính linh hoạt, độ chính xác cao và khả năng thích ứng với các điều kiện thực tế khác nhau để đảm bảo hiệu suất và độ tin cậy của nó.

## 2.3. Phương pháp thực hiện và các công nghệ hỗ trợ

Trong việc phát hiện biển số xe, em có thể sử dụng một loạt các phương pháp và công nghệ để đạt được kết quả chính xác và hiệu quả. Dưới đây là một số phương pháp và công nghệ phổ biến mà em có thể sử dụng:

**Chuyển ảnh sang ảnh xám (Convert to Grayscale):**

Chuyển đổi hình ảnh sang không gian màu xám giúp giảm bớt thông tin màu sắc không cần thiết và tăng tốc độ xử lý ảnh. Trong không gian màu xám, mỗi điểm ảnh chỉ có một giá trị độ sáng duy nhất thay vì ba giá trị (R, G, B) như trong không gian màu RGB, giảm chi phí tính toán cho việc xử lý ảnh.

**Áp dụng Gaussian Blur:**

Gaussian Blur là một kỹ thuật làm mờ hình ảnh bằng cách áp dụng bộ lọc Gaussian. Bằng cách này, ta giảm được nhiễu và các chi tiết không mong muốn trong ảnh. Việc làm mờ giúp loại bỏ nhiễu và làm cho các biên trở nên mượt mà, làm cho quá trình phát hiện biên trở nên dễ dàng hơn.

**Phát hiện biên Canny (Canny Edge Detection):**

Phát hiện biên Canny là một trong những phương pháp phổ biến nhất để tìm biên trong hình ảnh. Nó hoạt động bằng cách sử dụng gradient của độ sáng để xác định các biên. Phát hiện biên Canny tạo ra một hình ảnh chỉ có các điểm biên, làm giảm độ phức tạp của dữ liệu và tạo ra một hình ảnh dễ dàng phân tích hơn cho các bước tiếp theo.

**Tìm các đường viền (Find Contours):**

Sau khi đã có hình ảnh chỉ chứa các biên, ta sử dụng hàm findContours của OpenCV để tìm các đường viền trong hình ảnh. Điều này giúp xác định các vùng chứa biển số xe bằng cách tìm các đường viền đóng bao quanh các vật thể có cường độ độ sáng khác biệt với nền.

Vẽ các đường viền lên ảnh (Draw Contours on Image):

Ta vẽ các đường viền đã tìm được lên bản sao của ảnh gốc. Việc này giúp trực quan hóa kết quả của quá trình phát hiện biển số xe và kiểm tra hiệu suất của thuật toán.

Hiển thị kết quả tìm được :

Sau khi tắt cửa sổ hiển thị biển sổ với viền được bao xung quanh, kết quả sẽ hiển trên một cửa sổ Tkinter.

# CHƯƠNG III : CÀI ĐẶT VÀ THỬ NGHIỆM

## 3.1.Công cụ sử dụng

### PyCharm, một môi trường phát triển tích hợp (IDE) được JetBrains thiết kế đặc biệt cho việc phát triển ứng dụng Python, đem đến cho lập trình viên một loạt các tính năng và công cụ giúp tạo ra các ứng dụng Python một cách nhanh chóng và hiệu quả.

### Một trong những điểm mạnh của PyCharm là giao diện người dùng thân thiện và dễ sử dụng. Giao diện được thiết kế một cách thông minh, cho phép người dùng dễ dàng tìm kiếm và truy cập các tính năng, công cụ và tài liệu cần thiết. PyCharm cung cấp các công cụ tự động hoàn chỉnh cho việc viết mã, gỡ lỗi, kiểm tra và triển khai ứng dụng Python.

### Một tính năng nổi bật của PyCharm là khả năng hỗ trợ nhiều ngôn ngữ lập trình. Ngoài Python, PyCharm cũng hỗ trợ các ngôn ngữ như HTML, CSS, JavaScript và SQL. Điều này cho phép lập trình viên làm việc với nhiều loại tệp tin và dự án khác nhau trong một môi trường phát triển duy nhất.

### PyCharm cung cấp các tính năng mạnh mẽ cho việc quản lý dự án. Lập trình viên có thể dễ dàng tạo, mở, chỉnh sửa và xóa các tệp tin và thư mục trong dự án của mình. Hơn nữa, PyCharm cũng hỗ trợ việc quản lý các phiên bản của mã nguồn sử dụng các hệ thống quản lý phiên bản như Git và SVN.

### PyCharm cung cấp một loạt các tính năng hỗ trợ lập trình viên tối ưu hóa hiệu suất làm việc của họ. Công cụ tự động hoàn chỉnh mã, gợi ý mã, và kiểm tra lỗi giúp giảm thiểu thời gian và công sức cần thiết cho việc viết mã. Hơn nữa, PyCharm cũng tích hợp các công cụ gỡ lỗi mạnh mẽ, cho phép lập trình viên dễ dàng xác định và sửa lỗi trong mã của họ.

### Một tính năng đặc biệt của PyCharm là khả năng tích hợp với các công cụ và dịch vụ phổ biến khác như Docker, Vagrant và Anaconda. Điều này cho phép lập trình viên phát triển và triển khai các ứng dụng Python trên các môi trường phát triển và triển khai đa dạng một cách dễ dàng.

### PyCharm là một môi trường phát triển tích hợp mạnh mẽ và linh hoạt cho việc phát triển ứng dụng Python. Với giao diện người dùng thân thiện, các tính năng mạnh mẽ và tích hợp với các công cụ và dịch vụ phổ biến, PyCharm là lựa chọn lý tưởng cho lập trình viên Python muốn tạo ra các ứng dụng chất lượng và hiệu quả.

### 3.1.1.Cách cài đặt Pycharm

Để cài đặt PyCharm, em thực hiện các bước sau:

Tải xuống trình cài đặt PyCharm từ trang web chính thức của JetBrains.  
https://www.jetbrains.com/pycharm/download/?section=windows

Chạy tệp cài đặt và làm theo hướng dẫn trên màn hình để hoàn thành quá trình cài đặt.

Sau khi cài đặt hoàn tất, mở PyCharm và cấu hình môi trường làm việc.

### 3.1.2. Thư viện sử dụng đi kèm

**OpenCV**

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở hàng đầu trong lĩnh vực thị giác máy tính và xử lý ảnh. Được phát triển bởi Intel vào năm 1999, OpenCV đã trở thành một công cụ không thể thiếu cho các nhà nghiên cứu, nhà phát triển phần mềm và kỹ sư làm việc trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo.

Mục tiêu chính của OpenCV là cung cấp một nền tảng toàn diện cho việc phát triển các ứng dụng xử lý ảnh và thị giác máy tính. Thư viện này cung cấp một loạt các tính năng và công cụ để xử lý hình ảnh số, từ việc đọc và ghi ảnh, xử lý ảnh, phát hiện biên, nhận diện đối tượng, đến việc phân tích ký tự và thực hiện các tác vụ phức tạp như nhận diện khuôn mặt và theo dõi vật thể.

OpenCV được viết bằng ngôn ngữ lập trình C/C++ nhưng hỗ trợ các ngôn ngữ khác như Python, Java và MATLAB thông qua các giao diện lập trình ứng dụng (API). Điều này giúp OpenCV trở nên linh hoạt và dễ dàng tích hợp vào các dự án và ứng dụng khác nhau trên nhiều nền tảng.

Một trong những điểm mạnh của OpenCV là tính di động và đa nền tảng. Thư viện này được tối ưu hóa để hoạt động trên nhiều hệ điều hành khác nhau như Windows, Linux, macOS và cả các thiết bị nhúng. Điều này giúp cho việc phát triển và triển khai các ứng dụng thị giác máy tính trở nên dễ dàng và linh hoạt hơn.

Ngoài ra, OpenCV còn có một cộng đồng lớn và tích cực, với hàng nghìn thành viên đóng góp vào việc phát triển và cải thiện thư viện. Cộng đồng này cung cấp các tài liệu, ví dụ mã nguồn mở, và hỗ trợ kỹ thuật, giúp cho việc học và sử dụng OpenCV trở nên dễ dàng hơn đối với mọi người.

Tóm lại, OpenCV không chỉ là một thư viện mã nguồn mở phổ biến trong lĩnh vực thị giác máy tính và xử lý ảnh, mà còn là một công cụ mạnh mẽ và linh hoạt được sử dụng rộng rãi trong các ứng dụng từ nghiên cứu đến công nghiệp. Được hỗ trợ bởi một cộng đồng lớn và tích cực, OpenCV tiếp tục phát triển và trở thành một công cụ quan trọng trong nhiều lĩnh vực công nghệ hiện đại.

**NumPy**

NumPy là một thư viện mã nguồn mở cực kỳ mạnh mẽ và phổ biến trong ngôn ngữ lập trình Python, được thiết kế để hỗ trợ việc thao tác với mảng và ma trận số học một cách hiệu quả. Được phát triển lần đầu bởi Travis Oliphant vào năm 2005, NumPy đã trở thành một trong những công cụ cơ bản không thể thiếu trong nhiều ứng dụng khoa học và toán học, cũng như trong lĩnh vực máy học và học sâu.

Một trong những ưu điểm lớn nhất của NumPy là khả năng xử lý mảng và ma trận số lớn một cách nhanh chóng và hiệu quả. Thư viện này cung cấp một loạt các hàm và phương thức cho phép người dùng thực hiện các phép toán số học cơ bản như cộng, trừ, nhân, chia, cũng như các phép toán trên ma trận như nhân ma trận, chuyển vị ma trận, tính định thức và giải phương trình tuyến tính.

Ngoài ra, NumPy còn cung cấp một loạt các hàm và phương thức cho việc thao tác, chuyển đổi và truy cập dữ liệu trong mảng và ma trận. Điều này bao gồm việc trích xuất phần tử, cắt, ghép, lấy độ lớn, tính toán tổng, trung bình, phương sai và nhiều tính toán thống kê khác.

NumPy cũng là nền tảng cho nhiều thư viện và công cụ khác trong hệ sinh thái Python cho khoa học dữ liệu và máy học. Ví dụ, Pandas, một thư viện phổ biến cho phân tích dữ liệu, được xây dựng trên NumPy và sử dụng nó làm một trong những cấu trúc dữ liệu chính. Scikit-learn, một thư viện mạnh mẽ cho máy học, cũng sử dụng NumPy để thực hiện các tính toán số học và xử lý dữ liệu.

Một trong những điểm mạnh khác của NumPy là khả năng tương thích với nhiều thư viện và công nghệ khác, bao gồm C/C++, Fortran và MATLAB. Điều này làm cho NumPy trở thành một công cụ lý tưởng cho việc tích hợp và sử dụng cùng với các dự án có sẵn hoặc các mã nguồn mở khác.

Tóm lại, NumPy là một thư viện cực kỳ quan trọng và hữu ích trong việc thực hiện tính toán số học và xử lý dữ liệu trong Python. Khả năng linh hoạt, hiệu suất cao và tích hợp tốt với các công nghệ khác làm cho NumPy trở thành công cụ không thể thiếu đối với các nhà khoa học dữ liệu, nhà nghiên cứu và lập trình viên làm việc trong lĩnh vực khoa học máy tính và toán học.

**PIL**

PIL (Python Imaging Library) là một thư viện mã nguồn mở cho phép người dùng thực hiện các thao tác xử lý ảnh cơ bản như mở, lưu, chỉnh sửa, và chuyển đổi định dạng của hình ảnh. PIL cung cấp các công cụ cho việc thay đổi kích thước, cắt, xoay, lật và áp dụng các hiệu ứng đồ họa đơn giản cho hình ảnh. Mặc dù PIL không còn được phát triển tích cực từ năm 2011, nhưng nó vẫn là một lựa chọn phổ biến cho các tác vụ xử lý ảnh đơn giản trong Python.

**EasyOCR**

EasyOCR là một thư viện mã nguồn mở mạnh mẽ cho nhận diện ký tự quang học (OCR), sử dụng công nghệ mạng nơ-ron sâu để trích xuất văn bản từ hình ảnh với độ chính xác cao. Được phát triển bởi Jaided AI, EasyOCR đã nhanh chóng trở thành công cụ ưu việt trong lĩnh vực xử lý hình ảnh và trích xuất văn bản. Thư viện này hỗ trợ hơn 80 ngôn ngữ, bao gồm các ngôn ngữ phổ biến như tiếng Anh, tiếng Trung, tiếng Nhật, và nhiều ngôn ngữ khác. Điều này giúp EasyOCR trở thành một giải pháp lý tưởng cho các ứng dụng quốc tế hóa, nơi việc xử lý văn bản từ nhiều ngôn ngữ khác nhau là cần thiết.

EasyOCR nổi bật với khả năng cài đặt và sử dụng đơn giản, thân thiện với người dùng. Chỉ với vài dòng lệnh, người dùng có thể dễ dàng bắt đầu trích xuất văn bản từ hình ảnh. Thư viện này cũng có khả năng xử lý nhiều định dạng hình ảnh phổ biến như JPEG, PNG và TIFF, mang lại sự linh hoạt trong việc lựa chọn nguồn dữ liệu đầu vào. Khả năng tích hợp dễ dàng với các ứng dụng Python khác, bao gồm các framework như Flask và Django, giúp các nhà phát triển nhanh chóng thêm chức năng nhận diện văn bản vào các ứng dụng web hoặc các dự án AI của họ.

Trong thực tế, EasyOCR có rất nhiều ứng dụng. Nó có thể được sử dụng để nhận diện biển số xe trong các hệ thống giám sát giao thông hoặc các ứng dụng quản lý bãi đỗ xe, giúp tự động hóa quá trình thu thập và xử lý thông tin. Trong các doanh nghiệp, EasyOCR hỗ trợ việc số hóa và quản lý tài liệu in, tự động trích xuất thông tin từ các hợp đồng, hóa đơn, và các biểu mẫu, giúp tiết kiệm thời gian và giảm thiểu sai sót do nhập liệu thủ công. Ngoài ra, EasyOCR còn có thể nhận diện văn bản trong các hình ảnh chụp từ điện thoại hoặc máy ảnh, hữu ích trong các ứng dụng di động hoặc các dự án liên quan đến thị giác máy tính.

Cài đặt EasyOCR rất đơn giản, có thể thực hiện thông qua pip. Thư viện này yêu cầu một số thư viện phụ thuộc như PyTorch, nhưng hướng dẫn cài đặt chi tiết có sẵn trên trang GitHub của EasyOCR. Ví dụ cơ bản cho thấy cách sử dụng EasyOCR để nhận diện văn bản từ một hình ảnh là rất dễ thực hiện. Người dùng chỉ cần tạo một đối tượng Reader với ngôn ngữ mong muốn, sau đó sử dụng phương thức readtext để đọc văn bản từ hình ảnh và in kết quả.

Với khả năng hỗ trợ nhiều ngôn ngữ, độ chính xác cao và dễ dàng tích hợp, EasyOCR là một công cụ không thể thiếu trong bộ công cụ của các nhà phát triển AI và xử lý hình ảnh. Thư viện này không chỉ giúp tiết kiệm thời gian mà còn nâng cao hiệu quả và độ chính xác trong các tác vụ nhận diện văn bản, làm cho nó trở thành lựa chọn hàng đầu cho các dự án yêu cầu trích xuất văn bản từ hình ảnh.

## 3.2. Viết chương trình

Trước hết, để sử dụng OpenCV trong PyCharm, em cần cài đặt thư viện cho môi trường của mình. Em có thể thực hiện điều này dễ dàng thông qua terminal tích hợp trong PyCharm bằng cách thực hiện các bước sau:

Mở terminal trong PyCharm bằng cách nhấn tổ hợp phím Alt + F12.

Trong terminal, nhập lệnh sau:

pip install opencv-python

pip install easyocr

pip install opencv-python==4.5.4.60

pip install opencv-contrib-python==4.5.4.60

Việc này sẽ tải và cài đặt OpenCV cho môi trường Python của em, cho phép em sử dụng thư viện này trong dự án PyCharm của em một cách thuận tiện.

Sau khi cài đặt xong thư viện thì em sẽ code 1 chương trình đơn giản để phát hiện biển số xe  
  
import cv2

from PIL import ImageFont, ImageDraw, Image, ImageTk

from easyocr import Reader

from tkinter import Tk, Label, filedialog, Toplevel

import numpy as np

import os

# Sử dụng tkinter để mở hộp thoại chọn file

root = Tk()

root.withdraw() # Ẩn cửa sổ gốc của tkinter

filetypes = [("Image files", "\*.jpg \*.jpeg \*.png \*.webp"), ("Video files", "\*.mp4 \*.avi \*.mov \*.mkv")]

filenames = filedialog.askopenfilenames(title="Chọn ảnh hoặc video", filetypes=filetypes)

if not filenames:

raise FileNotFoundError("Không có file nào được chọn.")

# Tải font chữ

fontpath = r"C:\Users\tiphu\Downloads\Compressed\arial.ttf-master\arial.ttf-master\arial.ttf"

font = ImageFont.truetype(fontpath, 40)

# Định nghĩa màu chữ (RGB và alpha)

b, g, r, a = 0, 255, 0, 0

# Khởi tạo đối tượng EasyOCR reader

reader = Reader(['en'])

# Hàm xử lý và hiển thị kết quả cho một khung hình

def process\_frame(frame):

img = cv2.resize(frame, (800, 600))

# Chuyển ảnh sang ảnh xám

grayscale = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

blurred = cv2.GaussianBlur(grayscale, (5, 5), 0)

edged = cv2.Canny(blurred, 10, 200)

contours, \_ = cv2.findContours(edged, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

contours = sorted(contours, key=cv2.contourArea, reverse=True)[:5]

number\_plate\_shape = None

for c in contours:

perimeter = cv2.arcLength(c, True)

approximation = cv2.approxPolyDP(c, 0.02 \* perimeter, True)

if len(approximation) == 4: # Hình chữ nhật

number\_plate\_shape = approximation

break

text = "Không tìm thấy biển số xe"

if number\_plate\_shape is not None:

x, y, w, h = cv2.boundingRect(number\_plate\_shape)

number\_plate = grayscale[y:y + h, x:x + w]

detection = reader.readtext(number\_plate)

if len(detection) != 0:

text = f"Biển số: {detection[0][1]}"

# Vẽ khung xung quanh biển số

cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

return img, text

# Hiển thị kết quả video

def show\_video(filename):

# Mở video

cap = cv2.VideoCapture(filename)

if not cap.isOpened():

raise FileNotFoundError("Không thể mở file video được chọn.")

# Tạo cửa sổ hiển thị kết quả

result\_window = Toplevel(root)

result\_window.title(f"Kết quả nhận diện biển số xe - {os.path.basename(filename)}")

# Nhãn để hiển thị ảnh

img\_label = Label(result\_window)

img\_label.pack(side="left")

# Nhãn để hiển thị kết quả nhận diện

result\_label = Label(result\_window, font=("Arial", 40))

result\_label.pack(side="right")

while cap.isOpened():

ret, frame = cap.read()

if not ret:

break

processed\_img, text = process\_frame(frame)

# Resize ảnh để hiển thị nhỏ hơn

display\_img = cv2.resize(processed\_img, (800, 600))

# Hiển thị ảnh

img\_rgb = cv2.cvtColor(display\_img, cv2.COLOR\_BGR2RGB)

img\_pil = Image.fromarray(img\_rgb)

img\_tk = ImageTk.PhotoImage(img\_pil)

img\_label.config(image=img\_tk)

img\_label.image = img\_tk # Giữ tham chiếu để ảnh không bị xóa

# Hiển thị kết quả nhận diện

result\_label.config(text=text)

result\_window.update\_idletasks()

result\_window.update()

cap.release()

cv2.destroyAllWindows()

# Hiển thị kết quả ảnh

def show\_image(filename):

img = cv2.imread(filename)

if img is None:

raise FileNotFoundError("Không thể đọc file ảnh được chọn.")

processed\_img, text = process\_frame(img)

# Tạo cửa sổ hiển thị kết quả

result\_window = Toplevel(root)

result\_window.title(f"Kết quả nhận diện biển số xe - {os.path.basename(filename)}")

# Resize ảnh để hiển thị nhỏ hơn

display\_img = cv2.resize(processed\_img, (800, 400))

# Hiển thị ảnh

img\_rgb = cv2.cvtColor(display\_img, cv2.COLOR\_BGR2RGB)

img\_pil = Image.fromarray(img\_rgb)

img\_tk = ImageTk.PhotoImage(img\_pil)

img\_label = Label(result\_window, image=img\_tk)

img\_label.image = img\_tk # Giữ tham chiếu để ảnh không bị xóa

img\_label.pack(side="left")

# Hiển thị kết quả nhận diện với font lớn hơn

result\_label = Label(result\_window, text=text, font=("Arial", 40))

result\_label.pack(side="right")

result\_window.mainloop()

# Lặp qua từng file và xử lý

for filename in filenames:

minh = os.path.splitext(filename)[1].lower()

if minh in ['.mp4', '.avi', '.mov', '.mkv']:

show\_video(filename)

else:

show\_image(filename)

**3.2.1. Sơ đồ hệ thống**

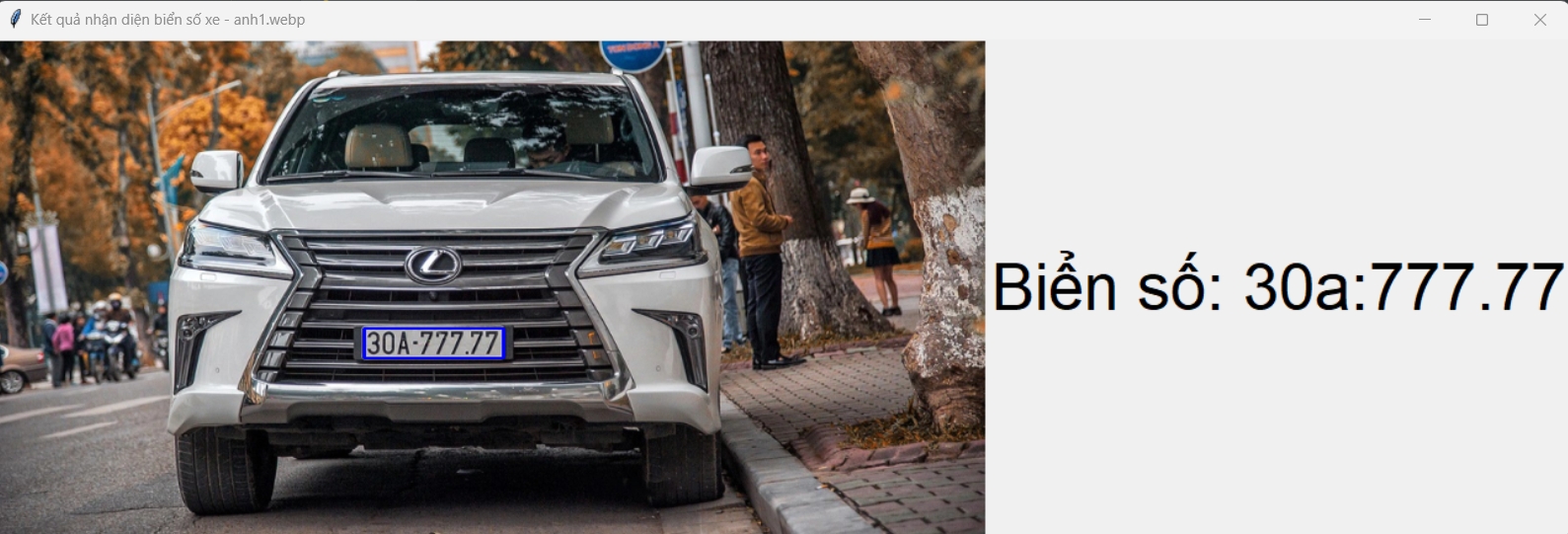
**A diagram of a flowchart

Description automatically generated**

### 3.2.2: Chạy chương trình

Chạy thử chương trình em thu được kết quả như sau:

Hình 2



Hình 3

A close up of a car

Description automatically generated

Hình 4 Trường hợp chưa xác định được

A car on the street

Description automatically generated

# 

A car parked in a parking lot

Description automatically generated

*\*Nhận xét kết quả :* Sau nhiều lần thử nghiệm em rút ra kết luận rằng chương trình này hoạt động tốt với các file input chất lượng cao (ánh sáng, độ nét, góc độ biển số).

*\*Nhược điểm còn tồn tại :* Không nhận diện được với file kém chất lượng, độ nghiêng lớn, ảnh có 2 xe trong cùng khung hình, đọc thiếu thông tin.

# KẾT LUẬN

Trong quá trình thực hiện dự án về phát hiện biển số xe, tôi đã bước vào một hành trình mới đầy thú vị và sâu sắc vào lĩnh vực xử lý ảnh. Đây là một lĩnh vực có sự ứng dụng rộng lớn trong thực tế và tôi đã tận dụng cơ hội này để học hỏi và phát triển kỹ năng cũng như tư duy logic của bản thân.

Qua quá trình nghiên cứu lý thuyết và thực hành, tôi đã đạt được những tiến bộ đáng kể. Việc hoàn thành dự án đúng thời hạn và theo yêu cầu đã đặt ra đối với tôi là một thành tựu đáng tự hào. Tôi đã thành công trong việc triển khai thuật toán phát hiện biển số xe và hiển thị chúng trên màn hình. Điều này không chỉ giúp tôi nắm vững hơn về kỹ thuật mà còn là một bước quan trọng trong việc áp dụng kiến thức vào thực tế.

Tuy nhiên, trong quá trình này, tôi đã phải đối mặt với nhiều thách thức và hạn chế như thời gian và kiến thức hạn chế, cùng với sự thiếu kinh nghiệm. Nhận biết và nhấn mạnh những điểm này sẽ giúp tôi hiểu rõ hơn về những khía cạnh cần cải thiện và phát triển trong tương lai.

Kinh nghiệm từ dự án này đã giúp tôi phát hiện ra sự yêu thích và hiểu biết sâu sắc hơn về lĩnh vực xử lý ảnh. Tôi hy vọng sẽ tiếp tục nhận được sự đóng góp và hỗ trợ từ giáo viên và bạn bè để có thể tiến xa hơn trong hành trình học tập và nghiên cứu của mình.

# TÀI LIỆU THAM KHẢO

1. Nguyễn Hữu Tuân. *Tài liệu học tập XỬ LÝ ẢNH*, NXB Đại học Hàng Hải, 2016.
2. <https://chat.openai.com/>
3. <https://www.youtube.com/watch?v=Zy1FGflg3-0&t=9s>
4. <https://github.com/vantupham/plate-recognition>
5. Cùng các tài liệu khác trên mạng