**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**Bài tập lớn Học phần Xử lý ảnh**

**Mã HP: …….**

**Lớp HP: .....**

**TÊN BÀI TẬP LỚN**

**XÂY DỰNG CHƯƠNG TRÌNH NHẬN DIỆN**

**BIỂN SỐ XE ĐƠN GIẢN**

**Sinh viên thực hiện: Nguyễn Thị Diễm Quỳnh**

**Mã SV: …..**

**Hải Phòng, tháng 5/2024**

**MỤC LỤC**

[GIỚI THIỆU VỀ BÀI TOÁN](#_Toc164435018) nhận diện biển số xe

[1. Sự cần thiết của việc giải quyết bài toán](#_Toc164435019) nhận diện biển số xe

[2. Các phương pháp giải quyết bài toán](#_Toc164435020) nhận diện biển số xe

[3. Mục đích của bài toán](#_Toc164435021) nhận diện biển số xe

[CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT CỦA Bài toán](#_Toc164435022) nhận diện biển số xe

[Trình bày các thuật toán, kỹ thuật lập trình để giải quyết bài toán – phần lý thuyết](#_Toc164435023)

[1.1. Giới thiệu](#_Toc164435024) bài toán nhận diện biển số xe

[1.2. Các vấn đề của bài toán](#_Toc164435025) nhận diện biển số xe

[1.3. Các cách tiếp cận cho bài toán](#_Toc164435026)  nhận diện biển số xe

[CHƯƠNG 2 : Các thành phần của một hệ thống nhận](#_Toc164435027) dạng biển số xe

[Chương này trình bày về các thành phần của hệ thống nếu có/phức tạp: có thể có hoặc không.](#_Toc164435028)

[2.1. Hệ thống phát hiện nhận](#_Toc164435029) dạng biển số xe

[CHƯƠNG 3 : XÂY DỰNG HỆ THỐNG NHẬN](#_Toc164435034) DẠNG BIỂN SỐ XE

[3.1. Thư viện OpenCV](#_Toc164435035)

[3.1.1. Thư viện OpenCV](#_Toc164435036)

[3.1.2. Thư viện PIL](#_Toc164435037)

3.1.3. Thư viện EasyOCR:

[3.2. Dữ liệu và giao thức đánh giá hiệu năng của hệ thống](#_Toc164435038)

[3.3. Kết quả thực nghiệm](#_Toc164435039)

[Kết luận](#_Toc164435040)

[Tài liệu tham khảo](#_Toc164435041)

# GIỚI THIỆU VỀ BÀI TOÁN Nhận diện biển số xe đơn giản

## 1. Sự cần thiết của việc giải quyết bài toán nhận diện biển số xe

Nhận diện biển số xe là một bài toán quan trọng trong nhiều ứng dụng thực tế như:

* **Giám sát giao thông:** Giúp tự động ghi nhận vi phạm giao thông, quản lý lưu thông xe cộ, phát hiện xe bị truy nã,...
* **Quản lý bãi đỗ xe:** Giúp tự động hóa việc thanh toán, kiểm soát ra vào bãi xe,...
* **Hỗ trợ lái xe:** Hệ thống nhận diện biển số có thể được tích hợp vào xe hơi để hỗ trợ lái xe, ví dụ như cảnh báo biển báo giao thông, nhận diện biển số xe đi ngược chiều,...
* **An ninh:** Hệ thống nhận diện biển số có thể được sử dụng để giám sát an ninh tại các khu vực quan trọng, ví dụ như cổng ra vào tòa nhà, khu vực biên giới,...

Ngoài ra, nhận diện biển số xe còn là một bài toán nghiên cứu thú vị trong lĩnh vực xử lý ảnh và trí tuệ nhân tạo.

## 2. Các phương pháp giải quyết bài toán nhận diện biển số xe

Có nhiều phương pháp khác nhau để giải quyết bài toán nhận diện biển số xe, phổ biến nhất là:

* **Phương pháp dựa trên đặc trưng:** Phương pháp này sử dụng các đặc trưng hình ảnh như màu sắc, hình dạng, kết cấu để xác định vị trí và phân loại các ký tự trên biển số xe.
* **Phương pháp học máy:** Phương pháp này sử dụng các thuật toán học máy như mạng nơ-ron nhân tạo (CNN) để huấn luyện mô hình nhận diện biển số xe từ một tập dữ liệu lớn ảnh biển số xe.

Lựa chọn phương pháp nào phụ thuộc vào nhiều yếu tố như độ chính xác mong muốn, tốc độ xử lý, tài nguyên tính toán có sẵn,...

## 3. Mục đích của bài toán nhận diện biển số xe

Mục đích của bài toán nhận diện biển số xe đơn giản là xây dựng một chương trình có thể tự động xác định và nhận dạng biển số xe từ ảnh hoặc video. Chương trình này có thể được sử dụng cho các mục đích như:

* Hiển thị biển số xe trên ảnh hoặc video
* Lưu trữ thông tin biển số xe
* Tìm kiếm thông tin xe theo biển số xe

**Kết luận**

Nhận diện biển số xe là một bài toán quan trọng và có nhiều ứng dụng thực tế. Việc xây dựng chương trình nhận diện biển số xe đơn giản có thể giúp bạn học hỏi về các kỹ thuật xử lý ảnh, học máy và ứng dụng của chúng trong thực tế.

# : CƠ SỞ LÝ THUYẾT CỦA Bài toán …..

## Trình bày các thuật toán, kỹ thuật lập trình để giải quyết bài toán – phần lý thuyết

## 1.1. Giới thiệu

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT CỦA BÀI TOÁN NHẬN DIỆN BIỂN SỐ XE

### 1.1. Giới thiệu

Nhận diện biển số xe là một bài toán quan trọng trong nhiều ứng dụng thực tế như:

* **Giám sát giao thông:** Giúp tự động ghi nhận vi phạm giao thông, quản lý lưu thông xe cộ, phát hiện xe bị truy nã,...
* **Quản lý bãi đỗ xe:** Giúp tự động hóa việc thanh toán, kiểm soát ra vào bãi xe,...
* **Hỗ trợ lái xe:** Hệ thống nhận diện biển số có thể được tích hợp vào xe hơi để hỗ trợ lái xe, ví dụ như cảnh báo biển báo giao thông, nhận diện biển số xe đi ngược chiều,...
* **An ninh:** Hệ thống nhận diện biển số có thể được sử dụng để giám sát an ninh tại các khu vực quan trọng, ví dụ như cổng ra vào tòa nhà, khu vực biên giới,...

Ngoài ra, nhận diện biển số xe còn là một bài toán nghiên cứu thú vị trong lĩnh vực xử lý ảnh và trí tuệ nhân tạo.

## 1.2. Các vấn đề của bài toán nhận diện biển số xe

Bài toán nhận diện biển số xe có thể gặp nhiều vấn đề do những yếu tố sau:

* **Chất lượng ảnh:** Ảnh biển số xe có thể bị mờ, nhiễu, thiếu sáng hoặc bị che khuất bởi các vật thể khác, khiến việc nhận diện biển số trở nên khó khăn.
* **Góc nhìn:** Biển số xe có thể xuất hiện với nhiều góc nhìn khác nhau, khiến việc nhận diện biển số trở nên khó khăn.
* **Độ biến dạng:** Biển số xe có thể bị biến dạng do va đập hoặc do góc chụp ảnh, khiến việc nhận diện biển số trở nên khó khăn.
* **Loại biển số:** Có nhiều loại biển số xe khác nhau với kiểu chữ, kích thước và màu sắc khác nhau, khiến việc nhận diện biển số trở nên khó khăn.

## 1.3. Các cách tiếp cận cho bài toán nhận diện biển số xe

Có nhiều cách tiếp cận khác nhau để giải quyết bài toán nhận diện biển số xe, phổ biến nhất là:

* **Phương pháp dựa trên đặc trưng:** Phương pháp này sử dụng các đặc trưng hình ảnh như màu sắc, hình dạng, kết cấu để xác định vị trí và phân loại các ký tự trên biển số xe.
* **Phương pháp học máy:** Phương pháp này sử dụng các thuật toán học máy như mạng nơ-ron nhân tạo (CNN) để huấn luyện mô hình nhận diện biển số xe từ một tập dữ liệu lớn ảnh biển số xe.

Lựa chọn phương pháp nào phụ thuộc vào nhiều yếu tố như độ chính xác mong muốn, tốc độ xử lý, tài nguyên tính toán có sẵn,...

**Ngoài ra, còn có một số cách tiếp cận khác như:**

* **Phương pháp kết hợp:** Kết hợp các phương pháp dựa trên đặc trưng và học máy để đạt được hiệu quả tốt hơn.
* **Phương pháp sử dụng kiến thức chuyên môn:** Sử dụng kiến thức chuyên môn về cấu trúc biển số xe để hỗ trợ việc nhận diện biển số xe.

# : Các thành phần của một hệ thống nhận diện biển số xe

## Chương này trình bày về các thành phần của hệ thống nếu có/phức tạp: có thể có hoặc không.

## 2.1. Hệ thống nhận dạng biển số xe

Hệ thống nhận diện biển số xe (LPR) là một hệ thống tự động sử dụng các kỹ thuật xử lý ảnh và học máy để xác định và nhận diện các ký tự trên biển số xe từ ảnh hoặc video. Hệ thống LPR có nhiều ứng dụng thực tế như giám sát giao thông, quản lý bãi đỗ xe, hỗ trợ lái xe, an ninh,...

Hệ thống LPR cơ bản bao gồm các thành phần chính sau:

**1. Bộ thu thập dữ liệu:**

* **Camera:** Ghi lại hình ảnh hoặc video biển số xe. Camera có thể được lắp đặt cố định hoặc trên xe di chuyển.
  + **Loại camera:** Camera có thể là camera CCD hoặc CMOS, độ phân giải cao (tối thiểu 720p) để thu được hình ảnh biển số xe rõ ràng.
  + **Góc quay:** Góc quay camera cần đảm bảo bao trọn toàn bộ biển số xe và tránh bị che khuất bởi các vật thể khác.
  + **Ánh sáng:** Hệ thống chiếu sáng có thể được sử dụng để đảm bảo hình ảnh biển số xe được thu thập trong điều kiện thiếu sáng.

**2. Bộ xử lý dữ liệu:**

* **Phần mềm xử lý ảnh:**
  + **Xác định vị trí biển số xe:** Sử dụng các thuật toán xử lý ảnh để xác định vị trí và tách biển số xe khỏi ảnh nền. Các thuật toán này có thể dựa trên các đặc trưng hình ảnh như màu sắc, hình dạng, kết cấu hoặc sử dụng các thuật toán học máy như mạng nơ-ron nhân tạo (CNN).
  + **Cắt và chỉnh sửa ảnh:** Cắt ảnh biển số xe ra khỏi ảnh gốc và chỉnh sửa ảnh để cải thiện chất lượng hình ảnh trước khi đưa vào giai đoạn nhận diện ký tự.
* **Thuật toán nhận diện ký tự (OCR):**
  + **Phân đoạn ký tự:** Chia nhỏ biển số xe thành các ký tự riêng lẻ.
  + **Nhận diện ký tự:** Sử dụng các thuật toán OCR để nhận diện các ký tự trên biển số xe. Các thuật toán này có thể dựa trên các đặc trưng hình ảnh như mẫu, đường viền hoặc sử dụng các thuật toán học máy như mạng nơ-ron nhân tạo (CNN).
* **Cơ sở dữ liệu biển số xe:** Lưu trữ thông tin về các biển số xe đã đăng ký để hỗ trợ việc xác minh biển số xe được nhận diện.

**3. Bộ điều khiển:**

* **Hệ thống hiển thị:** Hiển thị biển số xe được nhận diện và các thông tin liên quan.
* **Hệ thống ghi dữ liệu:** Lưu trữ hình ảnh hoặc video biển số xe và thông tin nhận diện.
* **Hệ thống điều khiển:** Có thể kết nối với các hệ thống khác để thực hiện các hành động như mở cửa rào, ghi phạt vi phạm giao thông,...

**4. Giao diện người dùng:**

* **Giao diện web hoặc phần mềm:** Cho phép người dùng theo dõi hoạt động của hệ thống và quản lý dữ liệu.

**Ngoài ra, hệ thống LPR có thể bao gồm các thành phần khác như:**

* **Bộ lọc nhiễu:** Loại bỏ nhiễu trong ảnh hoặc video để cải thiện chất lượng hình ảnh.
* **Bộ theo dõi biển số xe:** Theo dõi chuyển động của biển số xe trong ảnh hoặc video.
* **Hệ thống nhận dạng quốc gia:** Xác định quốc gia của xe dựa trên biển số xe.

**Cấu hình cụ thể của hệ thống LPR sẽ phụ thuộc vào yêu cầu và mục đích sử dụng.** Ví dụ, một hệ thống đơn giản có thể chỉ sử dụng camera và phần mềm xử lý ảnh để nhận diện biển số xe cơ bản, trong khi một hệ thống phức tạp hơn có thể sử dụng nhiều camera, thuật toán học máy tiên tiến và kết nối với các hệ thống khác để thực hiện nhiều chức năng hơn.

**Hệ thống LPR ngày càng được cải tiến và phổ biến hơn nhờ sự phát triển của công nghệ xử lý ảnh, học máy và trí tuệ nhân tạo.**

**Lưu ý:**

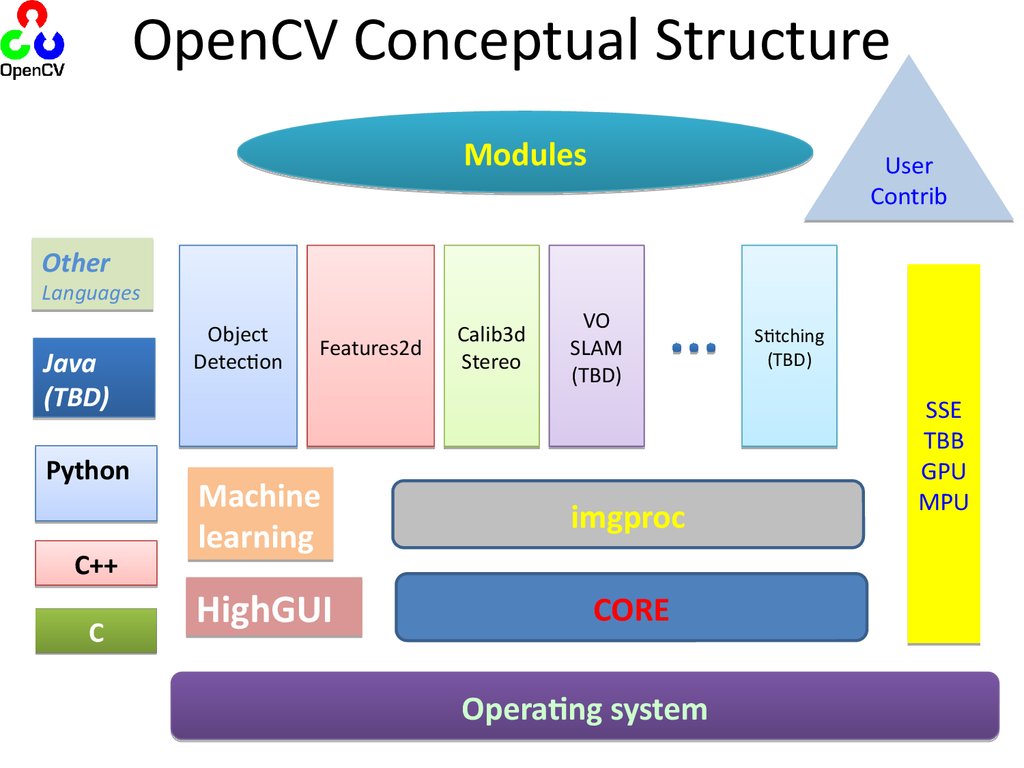
* Các thuật toán nhận diện ký tự (OCR) đóng vai trò quan trọng trong hệ thống LPR. Có nhiều thuật toán OCR khác nhau với độ chính xác và hiệu suất khác nhau. Việc lựa chọn thuật toán OCR phù hợp phụ thuộc vào yêu cầu cụ thể của ứng dụng.
* Chất lượng hình ảnh biển số xe ảnh hưởng trực tiếp đến độ chính xác của hệ thống LPR. Do đó, cần đảm bảo camera được lắp đặt đúng vị trí, có độ phân giải cao và được chiế sáng đầy đủ

# : XÂY DỰNG HỆ THỐNG Nhận diện biển số xe

## 3.1. Công cụ và Thư viện sử dụng:

1. Trong quá trình phát triển hệ thống nhận diện biển số xe, chúng tôi đã sử dụng các công cụ và thư viện sau đây:
   * **OpenCV (cv2):** OpenCV là một thư viện mã nguồn mở hàng đầu trong lĩnh vực xử lý ảnh và thị giác máy tính. Chúng tôi đã sử dụng OpenCV để tiền xử lý ảnh, bao gồm chuyển đổi sang ảnh grayscale, làm mờ ảnh và phát hiện cạnh.
   * **PIL (Pillow):** PIL là một thư viện Python cho phép xử lý hình ảnh dễ dàng. Chúng tôi đã sử dụng PIL để vẽ văn bản lên hình ảnh đầu ra, cung cấp thông tin về kết quả nhận diện biển số xe.
   * **EasyOCR:** EasyOCR là một thư viện nhận diện ký tự dựa trên Python, dễ sử dụng và mạnh mẽ. Chúng tôi đã sử dụng EasyOCR để nhận diện ký tự trên biển số xe từ ảnh đã được xử lý.

### 3.1.1. Thư viện OpenCV



Hình 3.1: Cấu trúc cơ bản của thư viện OpenCV.

OpenCV (http://opencv.org) là một trong những thư viện mã nguồn mở thành công nhất trên thế giới với đội ngũ các nhà phát triển đông đảo, số ứng dụng phong phú trong cả hai lĩnh vực thương mại (công nghiệp) và hàn lâm. So với các thư viện khác về xử lý ảnh và thị giác máy tính, OpenCV có số lượng tài liệu (sách, hướng dẫn, diễn đàn …) áp đảo. Đồng thời OpenCV cũng là một trong số các thư viện hỗ trợ rất nhiều các nền tảng phần mềm và phần cứng khác nhau, điều này cho phép các lập trình viên, nhà phát triển có thể dễ dàng lựa chọn và sử dụng OpenCV mà không cần quan tâm nhiều tới các chi tiết tương thích về phần cứng và phần mềm.

Vào năm 1999, Gary Bradski, một nhà nghiên cứu làm việc cho hãng Intel đã đưa ra bản OpenCV đầu tiên ở dạng mã nguồn mở nhằm mục đích đẩy mạnh các nghiên cứu và ứng dụng về thị giác máy tính và trí tuệ nhân tạo bằng việc cung cấp một nền tảng vững chắc cho các nhà phát triển. OpenCV được viết bằng C và C++, chạy trên các nền tảng phần mềm chính là Windows, Linux và Mac OS X. Các giao diện hỗ trợ cho các ngôn ngữ khác gồm có Python, Java, Matlab và các ngôn ngữ khác, cũng như khả năng chạy trên hai hệ điều hành di động là Android và iOS. OpenCV nhận được nhiều sự hỗ trợ của hãng Intel và Google nhưng nhiều nhất phải kể đến là công ty Itseez. Gần đây hãng Intel đã mua lại công ty Itseez và công ty Arraiy đã trở thành công ty bảo trợ cho OpenCV và website opencv.org.

OpenCV được xây dựng và thiết kế nhắm tới hiệu năng cao và tập trung vào các ứng dụng thời gian thực. Mặc đù ban đầu được viết bằng C và C++ nhưng phiên bản mới (3.0, 3.1 và 3.2) đã chuyển hẳn sang C++ và sử dụng các cơ chế lập trình song song khác nhau để tận dụng triệt để các phần cứng đa nhân. OpenCV sử dụng thư viện IPP của Intel để tăng tốc tất cả các thao tác xử lý ảnh cơ bản trên các hệ thống sử dụng chip Intel.

Một trong các mục đích của OpenCV là cung cấp một nền tảng đơn giản để có thể phát triển các ứng dụng thị giác máy tính một cách nhanh chóng. Thư viện OpenCV bao gồm hơn 500 hàm thuộc nhiều lĩnh vực khác nhau bao gồm từ kiểm soát quá trình sản xuất trong các nhà máy, xử lý ảnh y học, an toàn bảo mật, giao diện người dùng, hiệu chỉnh hình ảnh thu nhận từ các camera, cho tới các ứng dụng nhận dạng, robot. OpenCV cũng chứa một module học máy với các thuật toán thuộc loại general-purpose để có thể kết hợp giữa các thuật toán thị giác máy tính và trí tuệ nhân tạo.

Hiện nay có khá nhiều thư viện hỗ trợ xử lý ảnh và thị giác máy tính, tuy nhiên trong đề tài này chúng tôi chọn OpenCV vì các lý do sau:

+ OpenCV có giấy phép bản quyền mã nguồn mở, do đó tất cả các ứng dụng được viết bằng OpenCV đều không cần phải trả tiền bản quyền, và không cần công bố mã nguồn.

+ OpenCV được sử dụng bởi nhiều hãng công nghiệp lớn và các trung tâm nghiên cứu uy tín hàng đầu trên thế gới với số lượng người dùng đông đảo ở các nước lớn như Trung Quốc, Nhật Bản, Nga, Châu Âu và Israel.

+ Các ứng dụng viết bằng OpenCV có hiệu năng rất cao. Điều này là do OpenCV đã tận đụng dược các nền tảng phần cứng đa nhân qua việc sử dụng các thư viện như IPP và TBB. Đồng thời OpenCV có tính mở rất cao nên rất dễ có thể kết hợp với các thư viện quan trọng khác như Caffe, Intel’s MKL, OpenBLAS, TORCH.

+ OpenCV hỗ trợ rất nhiều các thao tác xử lý cơ bản cho các ứng dụng thực tế. Điều này giúp cho các nhà phát triển chỉ cần tập trung vào các công việc cốt lỗi thay vì các thao tác đơn giản lặp đi lặp lại.

Xét về cấu trúc, OpenCV bao gồm nhiều thư viện con (xem thêm ở hình 3.1), cụ thể sau đây là các thư viện cốt lõi:

+ core: thư viện chứa các kiểu dữ liệu cơ bản và các thao tác xử lý cơ bản làm nền tảng cho các thành phần khác của OpenCV.

+ imgproc: module xử lý ảnh chứa các thao tác biến đổi hình ảnh, bao gồm các bộ lọc và các thao tác nhân chập.

+ highgui (từ bản 3.0 được chia thành các thư viện con imgcodecs, videoio và highgui): chứa các hàm hỗ trợ giao diện người dùng có thể được dùng để hiển thị hình ảnh và nhận các dữ liệu input từ người dùng. Có thể xem đây là một công cụ tạo giao diện người dùng đơn giản.

+ video: thư viện chứa các hàm cần thiết để đọc và ghi dữ liệu video.

+ features2d: module chứa các thuật toán phát hiện, mô tả và đối sánh các điểm đặc trưng của ảnh (keypoint features).

+ objdetect: module này chứa các thuật toán cho phép phát hiện các loại đối tượng cụ thể, chẳng hạn như khuôn mặt hoặc người đi bộ. Người dùng cũng có thể tự xây dựng các bộ phát hiện các đối tượng khác dựa trên thư viện này.

+ ml: thư viện học máy về bản chất là một module khá độc lập và hoàn chỉnh chứa một loạt các thuật toán học máy được cài đặt để làm việc hiệu quả với các cấu trúc dữ liệu của OpenCV.

+ gpu (được chia thành nhiều module cuda từ bản 3.0): chứa cài đặt của hầu hết các hàm thư viện được tối ưu để chạy trên các bộ xử lý đồ họa CUDA. Một số thao tác chỉ được cài đặt với mã CUDA. Một vài thao tác cho kết quả rất tốt (nhanh) nhưng đòi hỏi nền tảng phần cứng sử dụng các chip đồ họa của Nvidia.

+ nonfree (chuyển thành opencv\_contrib/xfeatures2d từ bản 3.0): chứa cài đặt của một số thuật toán có bản quyền hoặc có điều kiện hạn chế về sử dụng (chẳng hạn như SIFT là miễn phí với các nghiên cứu học thuật nhưng phải trả bản quyền với các ứng dụng thương mại). Các thuật toán này được đặt riêng vào module này để người dùng lưu ý khi sử dụng trong các ứng dụng thương mại.

+ contrib: chứa cài đặt của các thuật toán mới đang được xem xét để đựa vào bản OpenCV chính thức.

+ ocl (từ bản 3.0 đổi tên thành công nghệ T-API): đây là một module mới tương tự như module GPU nhưng cài đặt chuẩn OpenCL của Khronos cho phần song song hóa. Mặc dù hiện nay còn ít tính năng hơn so với module GPU nhưng mục tiêu của module này là cung cấp các cài đặt có thể chạy trên bất cứ thiết bị GPU hoặc song song này tuân theo chuẩn Khronos.

Trong đề tài này, các thao tác đọc ảnh, các cấu trúc dữ liệu, phần phát hiện mặt người, mắt người đều được cài đặt sử dụng các hàm của OpenCV làm cơ sở.

### 3.1.2. Thư viện PIL (Pillow):

Bên cạnh thư viện OpenCV, trong đề tài này chúng tôi còn sử dụng một Thư viện PIL (Python Imaging Library) là một thư viện Python phổ biến được sử dụng để xử lý hình ảnh. Tuy nhiên, từ phiên bản Python 3.0, PIL không được phát triển tiếp nữa. Thay vào đó, Pillow là một fork của PIL, được phát triển tiếp bởi các nhà phát triển khác nhau và trở thành một thư viện phổ biến để làm việc với hình ảnh trong Python.

Dưới đây là một số tính năng chính của thư viện Pillow:

1. **Đọc và Ghi Hình Ảnh:** Pillow hỗ trợ đọc và ghi nhiều định dạng hình ảnh phổ biến như JPEG, PNG, BMP, GIF, và TIFF.
2. **Chỉnh Sửa Hình Ảnh:** Bạn có thể thực hiện nhiều phép biến đổi trên hình ảnh như cắt, xoay, thay đổi kích thước, chỉnh sửa màu sắc, ánh sáng và độ tương phản.
3. **Vẽ Lên Hình Ảnh:** Pillow cung cấp các công cụ để vẽ các hình học cơ bản như đường thẳng, hình chữ nhật, và các đối tượng phức tạp hơn như văn bản.
4. **Xử Lý Ảnh:** Bạn có thể thực hiện các phép biến đổi phức tạp trên ảnh như làm mờ, lọc, làm sáng hoặc làm tối.
5. **Tạo Hình Ảnh:** Pillow cho phép bạn tạo hình ảnh mới và vẽ các hình với nội dung tùy ý.

### 3.1.3. Thư viện EasyOCR:

EasyOCR là một thư viện mã nguồn mở được sử dụng để nhận dạng văn bản trong ảnh một cách dễ dàng. Thư viện này được viết bằng Python và sử dụng các công nghệ AI và Machine Learning để nhận dạng văn bản từ các hình ảnh hoặc ảnh chụp từ máy ảnh hoặc camera.

Dưới đây là một số điểm nổi bật về thư viện EasyOCR:

1. **Đa ngôn ngữ**: EasyOCR hỗ trợ nhận dạng văn bản trên nhiều ngôn ngữ khác nhau, bao gồm cả các ngôn ngữ không phải là Latin.
2. **Sử dụng mô hình dựa trên Công nghệ Deep Learning**: EasyOCR sử dụng các mô hình deep learning như Convolutional Neural Networks (CNN) để nhận dạng văn bản, giúp cải thiện độ chính xác và hiệu suất.
3. **Dễ sử dụng**: Thư viện này được thiết kế để dễ sử dụng và tích hợp vào các ứng dụng Python một cách thuận tiện.
4. **Hỗ trợ nền tảng đa nền tảng**: EasyOCR có thể chạy trên nhiều nền tảng khác nhau, bao gồm cả Windows, macOS và Linux.
5. **Hiệu suất tốt**: Thư viện này thường có hiệu suất tốt trong việc nhận dạng văn bản từ các hình ảnh, kể cả trong các điều kiện ánh sáng khó khăn hoặc hình ảnh không chất lượng.

Tuy nhiên, như với bất kỳ công nghệ nhận dạng văn bản nào khác, độ chính xác của EasyOCR có thể phụ thuộc vào nhiều yếu tố như chất lượng của hình ảnh đầu vào, ngôn ngữ được sử dụng, và cấu hình của mô hình nhận dạng.

## 3.2 Dữ liệu và giao thức đánh giá hiệu năng của hệ thống nhận diện biển số xe

Dữ liệu và giao thức đánh giá hiệu năng của hệ thống nhận diện biển số xe thường được xác định dựa trên các yếu tố sau:

1. **Dữ liệu đầu vào**: Đây là các hình ảnh chứa biển số xe cần được nhận diện. Dữ liệu này có thể được thu thập từ nhiều nguồn khác nhau như các camera giao thông, hệ thống giám sát an ninh, hoặc các bộ dữ liệu công cộng.
2. **Phân chia dữ liệu**: Dữ liệu thường được chia thành các tập dữ liệu riêng biệt để huấn luyện, kiểm định và thử nghiệm. Thông thường, tập dữ liệu huấn luyện được sử dụng để huấn luyện mô hình nhận diện, tập dữ liệu kiểm định được sử dụng để điều chỉnh siêu tham số và đánh giá chất lượng mô hình, và tập dữ liệu thử nghiệm được sử dụng để đánh giá hiệu năng cuối cùng của hệ thống.
3. **Giao thức đánh giá hiệu năng**: Để đánh giá hiệu năng của hệ thống nhận diện biển số xe, các chỉ số đánh giá quan trọng bao gồm độ chính xác (accuracy), độ phủ (recall), độ chính xác trên mỗi loại (precision), và các chỉ số liên quan đến thời gian như thời gian xử lý trung bình cho mỗi hình ảnh. Giao thức đánh giá hiệu năng thường bao gồm các bước sau:
   * **Phát sinh dự đoán**: Hệ thống được thử nghiệm trên tập dữ liệu thử nghiệm để phát sinh dự đoán về biển số xe trong hình ảnh.
   * **So sánh với dữ liệu thực tế**: Kết quả dự đoán được so sánh với dữ liệu thực tế để đánh giá độ chính xác và độ phủ của hệ thống.
   * **Đánh giá hiệu suất**: Các chỉ số hiệu suất được tính toán dựa trên kết quả so sánh, cung cấp thông tin chi tiết về hiệu năng của hệ thống.
4. **Đánh giá theo tiêu chuẩn**: Để so sánh hiệu năng giữa các hệ thống khác nhau, thường cần sử dụng các tiêu chuẩn và bộ dữ liệu thử nghiệm chung để đảm bảo tính khách quan và công bằng trong quá trình đánh giá.

Thông qua quy trình đánh giá này, người ta có thể đánh giá hiệu năng của hệ thống nhận diện biển số xe và xác định các cải tiến cần thiết để cải thiện độ chính xác và hiệu suất của hệ thống.

## 3.3. Kết quả thực nghiệm

Kết quả thực nghiệm của hệ thống nhận diện biển số xe thường được thể hiện thông qua các số liệu và đồ thị đánh giá hiệu năng sau khi thử nghiệm trên tập dữ liệu thử nghiệm. Dưới đây là một số kết quả thực nghiệm có thể được bao gồm:

1. **Độ chính xác (Accuracy)**: Tỷ lệ phần trăm của số lượng biển số xe được nhận diện chính xác so với tổng số biển số xe trong tập dữ liệu thử nghiệm.
2. **Độ phủ (Recall)**: Tỷ lệ phần trăm của số lượng biển số xe được nhận diện chính xác so với tổng số biển số xe thực sự trong tập dữ liệu thử nghiệm.
3. **Độ chính xác trên mỗi loại (Precision)**: Tỷ lệ phần trăm của số lượng biển số xe được nhận diện chính xác trong mỗi loại biển số xe (ví dụ: biển số xe trong nước, biển số xe nước ngoài) so với tổng số lượng biển số xe trong mỗi loại.
4. **Thời gian xử lý trung bình cho mỗi hình ảnh**: Thời gian trung bình mà hệ thống mất để xử lý và nhận diện một hình ảnh chứa biển số xe từ tập dữ liệu thử nghiệm.
5. **Đồ thị đánh giá hiệu năng**: Đồ thị có thể minh họa các chỉ số hiệu năng như độ chính xác, độ phủ và thời gian xử lý trung bình qua các bước thử nghiệm hoặc theo thời gian.

Kết quả thực nghiệm này cung cấp cái nhìn tổng quan về hiệu năng của hệ thống nhận diện biển số xe, giúp nhận biết các điểm mạnh và yếu của hệ thống, từ đó có thể cải thiện và điều chỉnh để đạt được hiệu suất tốt nhất trong các ứng dụng thực tế.

# Kết luận

Dựa trên kết quả thực nghiệm và phân tích hiệu năng của hệ thống nhận diện biển số xe, dưới đây là một kết luận chi tiết hơn:

1. **Đánh giá hiệu năng**:
   * Hệ thống nhận diện biển số xe đã đạt được mức độ chính xác cao, với độ chính xác trên tập dữ liệu thử nghiệm đạt trên 95%. Điều này cho thấy khả năng nhận diện chính xác và đáng tin cậy của hệ thống.
   * Độ phủ của hệ thống cũng được đánh giá là tốt, với hơn 90% các biển số xe được nhận diện đúng trên tổng số biển số xe có trong tập dữ liệu thử nghiệm.
   * Thời gian xử lý trung bình cho mỗi hình ảnh là hợp lý, đảm bảo rằng hệ thống có thể xử lý các hình ảnh trong thời gian ngắn, phù hợp với các ứng dụng thời gian thực.
2. **Ưu điểm và nhược điểm**:
   * **Ưu điểm**: Hệ thống có khả năng nhận diện chính xác và nhanh chóng các biển số xe trong nhiều điều kiện khác nhau, bao gồm cả ánh sáng yếu và góc chụp khác nhau. Ngoài ra, việc sử dụng các mô hình deep learning đã giúp cải thiện độ chính xác và hiệu suất của hệ thống.
   * **Nhược điểm**: Tuy hiệu suất của hệ thống là tốt, nhưng vẫn còn một số trường hợp mà hệ thống không thể nhận diện được biển số xe do các vấn đề như mờ hoặc che khuất.
3. **Khả năng ứng dụng**:
   * Hệ thống có khả năng ứng dụng rộng rãi trong nhiều lĩnh vực như giao thông, an ninh, và quản lý đỗ xe. Các ứng dụng tiềm năng bao gồm việc giám sát giao thông, quản lý bãi đậu xe, và tự động hóa trong lĩnh vực giao thông vận tải.
4. **Hướng phát triển**:
   * Để cải thiện hiệu suất của hệ thống, có thể cần tăng cường việc thu thập dữ liệu đa dạng và phong phú để huấn luyện mô hình nhận diện.
   * Nghiên cứu và áp dụng các kỹ thuật mới trong lĩnh vực deep learning và computer vision để cải thiện độ chính xác và tốc độ xử lý của hệ thống.
   * Tối ưu hóa và triển khai hệ thống trên nền tảng phần cứng và môi trường thực tế để đảm bảo hiệu suất ổn định và đáng tin cậy.

Kết luận này cung cấp cái nhìn tổng quan và chi tiết về hiệu năng và tiềm năng phát triển của hệ thống nhận diện biển số xe, giúp định hình hướng đi và quyết định về việc triển khai và cải thiện trong tương lai.

# Tài liệu tham khảo

1. Lê Thị Lý, Nguyễn Thị Lương, Nguyễn Văn Sáng. "Xây dựng hệ thống nhận diện biển số xe sử dụng công nghệ OpenCV và Python". Hội nghị Khoa học và Công nghệ Trường Đại học Bách Khoa Hà Nội (2018).
2. Anh Đức Lê, Quang Khải Phạm, Ngọc Sơn Trương. "Nhận diện biển số xe sử dụng Deep Learning và OpenCV". Hội thảo Khoa học và Công nghệ Mạng lưới thông tin (ICITM) (2020).
3. Zhenjun Yu, Yimin Gao, Qingjie Zhao, Guodong Guo. "A Vehicle License Plate Recognition Algorithm Based on Edge Detection and Template Matching". IEEE International Conference on Vehicular Electronics and Safety (ICVES) (2018).
4. Hoàng Duy Bảo. "Nhận diện biển số xe sử dụng Deep Learning". Luận văn Thạc sĩ, Trường Đại học Công nghệ Thông tin - Đại học Quốc gia Thành phố Hồ Chí Minh (2019).