**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 10: Xây dựng hệ thống nhận diện biển số xe tự động với OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210893 | Phạm Mạnh Cương | DCCNTT12.10.3 |
| **2** | 20210763 | Nguyễn Huy Hoàng | DCCNTT12.10.3 |
| **3** | 20210909 | Nguyễn Văn Tuấn | DCCNTT12.10.3 |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 10: Xây dựng hệ thống nhận diện biển số xe tự động với OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210893 | Phạm Mạnh Cương | DCCNTT12.10.3 |
| **2** | 20210763 | Nguyễn Huy Hoàng | DCCNTT12.10.3 |
| **3** | 20210909 | Nguyễn Văn Tuấn | DCCNTT12.10.3 |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 10**  **Tên học phần: Xử lý ảnh và thị giác máy tính**  **Lớp Tín chỉ: DCCNTT12.10.3** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Phạm Mạnh Cương** | **Nguyễn Huy Hoàng** | **Nguyễn Văn Tuấn** |
| --- | --- | --- | --- | --- | --- |
| 20210893 | 20210763 | 20210909 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |

# **LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành đến cô Lương Thị Hồng Lan, người đã tận tình chỉ bảo, định hướng và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài. Những ý kiến đóng góp quý báu của thầy/cô không chỉ giúp chúng em hiểu rõ hơn về bài toán mà còn là nguồn động lực lớn để chúng em hoàn thành tốt báo cáo này.

Chúng em cũng xin cảm ơn Nhà trường và Khoa Công nghệ Thông tin đã tạo điều kiện thuận lợi, cung cấp các nguồn tài liệu và môi trường học tập lý tưởng, giúp chúng em tiếp cận với các công nghệ hiện đại phục vụ nghiên cứu. Bên cạnh đó, sự động viên và hỗ trợ từ gia đình, bạn bè luôn là điểm tựa tinh thần vững chắc để chúng em vượt qua mọi khó khăn trong quá trình thực hiện đề tài.

Cuối cùng, chúng em chân thành cảm ơn cộng đồng phát triển mã nguồn mở, đặc biệt là các tác giả của OpenCV và EasyOCR, những công cụ quan trọng đã giúp chúng em triển khai hệ thống một cách hiệu quả. Chúng em mong rằng với sự nỗ lực của mình, đề tài này sẽ mang lại giá trị thiết thực và nhận được những ý kiến đóng góp quý báu từ thầy cô và các bạn.

Chúng em xin chân thành cảm ơn!

# **LỜI NÓI ĐẦU**

Trong thời đại công nghệ 4.0, việc ứng dụng trí tuệ nhân tạo và xử lý ảnh vào đời sống ngày càng trở nên phổ biến và đóng vai trò quan trọng trong nhiều lĩnh vực như giao thông, y tế, và quản lý thông tin. Một trong những ứng dụng nổi bật là hệ thống nhận diện biển số xe tự động, giúp nâng cao hiệu quả và tính chính xác trong quản lý phương tiện giao thông.

Đề tài *"Xây dựng hệ thống nhận diện biển số xe tự động với OpenCV"* không chỉ mang tính thực tiễn cao mà còn giúp chúng em hiểu rõ hơn về các kỹ thuật xử lý ảnh và nhận diện ký tự. Báo cáo này được thực hiện nhằm trình bày toàn bộ quá trình phân tích, thiết kế, và triển khai hệ thống, cũng như đánh giá kết quả thực nghiệm trên bộ dữ liệu thực tế.

Chúng em hy vọng rằng báo cáo này sẽ mang lại một cái nhìn rõ ràng và chi tiết về quy trình xây dựng một hệ thống nhận diện biển số xe, đồng thời là một tài liệu tham khảo hữu ích cho những ai quan tâm đến lĩnh vực xử lý ảnh và thị giác máy tính.

# **MỤC LỤC**

[LỜI CẢM ƠN 6](#_Toc184483721)

[LỜI NÓI ĐẦU 7](#_Toc184483722)

[MỤC LỤC 8](#_Toc184483723)

[DANH MỤC HÌNH ẢNH 9](#_Toc184483724)

[CHƯƠNG I: CƠ SỞ LÝ THUYẾT 10](#_Toc184483725)

[1.1 Tổng quan về xử lý ảnh và thị giác máy tính 10](#_Toc184483726)

[1.1.1 Giới thiệu xử lý ảnh và thị giác máy tính 10](#_Toc184483727)

[1.1.2 Ứng dụng thực tế của xử lý ảnh trong nhận diện biển số xe 11](#_Toc184483728)

[1.2. Tổng quan về OpenCV 11](#_Toc184483729)

[1.2.1. Giới thiệu về OpenCV 11](#_Toc184483730)

[1.2.2. Các tính năng chính 12](#_Toc184483731)

[1.2.3. Khái niệm cơ bản trong xử lý ảnh số 12](#_Toc184483732)

[1.2.4. Ứng dụng 13](#_Toc184483733)

[CHƯƠNG II: XÂY DỰNG HỆ THỐNG 14](#_Toc184483734)

[2.1 Bài toán nói chung 14](#_Toc184483735)

[2.2 Các kỹ thuật sử dụng trong bài toán nhận dạng biển số xe 16](#_Toc184483736)

[2.3. Xây dựng hệ thống nhận diện biển số xe 20](#_Toc184483737)

[2.3.1. Tiền xử lý ảnh 20](#_Toc184483738)

[2.3.2. Phát hiện biển số xe bằng OpenCV 20](#_Toc184483739)

[CHƯƠNG III: THỰC NGHIỆM 23](#_Toc184483740)

[5.1. Ngôn ngữ lập trình và các thư viện 23](#_Toc184483741)

[5.1.1. Ngôn ngữ lập trình Python 23](#_Toc184483742)

[5.1.2. Các thư viện được sử dụng 23](#_Toc184483743)

[5.2. Dữ liệu sử dụng 25](#_Toc184483744)

[5.3. Kết quả thực nghiệm 26](#_Toc184483745)

[KẾT LUẬN 30](#_Toc184483746)

[TÀI LIỆU THAM KHẢO 31](#_Toc184483747)

# **DANH MỤC HÌNH ẢNH**

[Hình 1: Các bước cơ bản trong xử lý ảnh 12](#_Toc184461681)

[Hình 2: Ảnh minh họa chương trình nhận diện biển số xe 15](#_Toc184461682)

[Hình 3: Ảnh minh họa ảnh gốc 17](#_Toc184461683)

[Hình 4: Ảnh minh họa chuyển đổi sang thang xám 18](#_Toc184461684)

[Hình 5: Ảnh minh họa sau khi phát hiện biên 18](#_Toc184461685)

[Hình 6: Ảnh minh họa phát hiện vùng biển số và khaonh vùng 19](#_Toc184461686)

[Hình 7: Ảnh minh họa kết quả nhận dạng ký tự từ biển số 20](#_Toc184461687)

[Hình 8: Ảnh biển số rõ nét 27](#_Toc184461688)

[Hình 9: Ảnh biển số mờ hoặc điều kiện ánh sáng yếu 27](#_Toc184461689)

[Hình 10: Kết quả mã tỉnh/thành phố đầy đủ ký tự 28](#_Toc184461690)

[Hình 11: Kết quả khi mã tỉnh/thành phố mất đi 1 phần ký tự 28](#_Toc184461691)

[Hình 12: Ảnh đầu vào 29](#_Toc184461692)

[Hình 13: Ảnh đã chọn ROI 29](#_Toc184461693)

[Hình 14: kết quả nhận diện. 29](#_Toc184461694)

# **CHƯƠNG I: CƠ SỞ LÝ THUYẾT**

## **1.1 Tổng quan về xử lý ảnh và thị giác máy tính**

### **1.1.1 Giới thiệu xử lý ảnh và thị giác máy tính**

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác. Đầu tiên phải kể đến Xử lý tín hiệu số là một môn học hết sức cơ bản cho xử lý tín hiệu chung, các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn… Thứ hai, các công cụ toán như Đại số tuyến tính, Sác xuất, thống kê. Một số kiến thứ cần thiết như Trí tuệ nhân tao, Mạng nơ ron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh.

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh sô thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.

Để dễ tưởng tượng, xét các bước cần thiết trong xử lý ảnh. Đầu tiên, ảnh tự nhiên từ thế giới ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kiểu CCIR). Gần đây, với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. (Máy ảnh số hiện nay là một thí dụ gần gũi). Mặt khác, ảnh cũng có thể tiếp nhận từ vệ tinh; có thể quét từ ảnh chụp bằng máy quét ảnh. Hình 1.1 dưới đây mô tả các bước cơ bản trong xử lý ảnh.

**A diagram of a boat

Description automatically generated**

Hình 1: Các bước cơ bản trong xử lý ảnh

**1.1.2 Ứng dụng thực tế của xử lý ảnh trong nhận diện biển số xe**

Ứng dụng xử lý ảnh trong nhận diện biển số xe đã trở thành một giải pháp quan trọng để tự động hóa nhiều tác vụ liên quan đến giao thông và quản lý phương tiện. Một số ứng dụng chính bao gồm:

* **Quản lý giao thông**: Giám sát, phát hiện vi phạm và tự động gửi phạt nguội nhờ nhận diện biển số từ camera giao thông.
* **Thu phí tự động**: Các hệ thống ETC (Electronic Toll Collection) nhận diện biển số xe để tự động tính phí mà không cần dừng xe.
* **Bãi đỗ xe thông minh**: Quản lý ra/vào bãi đỗ xe dựa trên nhận diện biển số thay vì vé giấy hoặc thẻ từ.
* **Điều tra an ninh**: Xác định và truy vết phương tiện tham gia vào các hoạt động đáng ngờ thông qua phân tích hình ảnh.

## **1.2. Tổng quan về OpenCV**

### **1.2.1. Giới thiệu về OpenCV**

OpenCV, viết tắt của *Open Source Computer Vision Library*, là một thư viện mã nguồn mở được phát triển nhằm cung cấp các công cụ và thuật toán phục vụ xử lý ảnh và thị giác máy tính. Được khởi tạo bởi Intel vào năm 1999 và sau đó phát hành công khai vào năm 2000, OpenCV nhanh chóng trở thành thư viện phổ biến nhất trong cộng đồng nghiên cứu và ứng dụng thị giác máy tính.

Với hơn 2,500 thuật toán được tích hợp sẵn, OpenCV cung cấp khả năng xử lý ảnh thời gian thực, từ các tác vụ đơn giản như chỉnh sửa ảnh đến các bài toán phức tạp như phân tích video và học sâu. Hiện tại, thư viện này hỗ trợ đa nền tảng, bao gồm Windows, macOS, Linux, Android, và iOS.

### **1.2.2. Các tính năng chính**

* **Xử lý ảnh số**: Cung cấp công cụ để xử lý ảnh như thay đổi kích thước, xoay ảnh, lọc ảnh, phát hiện cạnh, và chỉnh sửa độ sáng, tương phản.
* **Xử lý video**: Hỗ trợ đọc, ghi, và phân tích luồng video theo thời gian thực, cho phép thực hiện các tác vụ như phát hiện chuyển động hoặc trích xuất đối tượng từ video.
* **Phát hiện và nhận diện đối tượng**: Bao gồm phát hiện khuôn mặt, mắt, nụ cười, và các đối tượng khác thông qua mô hình học máy hoặc học sâu.
* **Hỗ trợ học máy và học sâu**: OpenCV tích hợp các thuật toán học máy cơ bản và cho phép sử dụng các mô hình học sâu từ các framework phổ biến như TensorFlow và PyTorch.
* **Tương thích đa ngôn ngữ**: OpenCV hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Python, C++, Java, và MATLAB, giúp lập trình viên dễ dàng tích hợp vào ứng dụng.

### **1.2.3. Khái niệm cơ bản trong xử lý ảnh số**

* **Ảnh số và không gian màu**:
  + Ảnh số là tập hợp các pixel, trong đó mỗi pixel mang một giá trị màu hoặc mức xám nhất định.
  + Không gian màu phổ biến gồm:
    - **RGB**: Ba kênh màu đỏ, xanh lá, và xanh dương.
    - **HSV**: Biểu diễn màu sắc theo sắc độ (Hue), độ bão hòa (Saturation), và giá trị ánh sáng (Value).
    - **Grayscale**: Ảnh xám với giá trị pixel nằm trong khoảng từ 0 (đen) đến 255 (trắng).
* **Biến đổi hình học**:
  + **Dịch chuyển (Translation)**: Di chuyển ảnh theo một hướng xác định.
  + **Xoay (Rotation)**: Thay đổi góc của ảnh.
  + **Phóng to/thu nhỏ (Scaling)**: Thay đổi kích thước ảnh.
  + **Cắt ảnh (Cropping)**: Lấy một phần cụ thể của ảnh.
* **Làm mịn và giảm nhiễu**:
  + **Lọc Gaussian**: Làm mịn ảnh bằng cách giảm nhiễu nhưng vẫn giữ được cạnh rõ nét.
  + **Lọc trung vị (Median Filter)**: Loại bỏ nhiễu dạng điểm (salt-and-pepper noise) hiệu quả.
* **Phát hiện cạnh (Edge Detection)**:
  + **Sobel**: Phát hiện biên bằng cách tính gradient theo hướng x và y.
  + **Canny**: Một trong những phương pháp phát hiện biên mạnh mẽ nhất, dựa trên ngưỡng hai mức.

### **1.2.4. Ứng dụng**

OpenCV được sử dụng rộng rãi trong các lĩnh vực sau:

* **Giám sát giao thông**: Nhận diện biển số xe, phát hiện và theo dõi phương tiện.
* **Y tế**: Phân tích hình ảnh y khoa như MRI hoặc CT.
* **Thương mại điện tử**: Hỗ trợ tìm kiếm sản phẩm bằng hình ảnh.
* **Nhận diện khuôn mặt**: Ứng dụng trong bảo mật và xác thực danh tính.

# **CHƯƠNG II: XÂY DỰNG HỆ THỐNG**

## **2.1 Bài toán nói chung**

Nhận diện đối tượng trong ảnh là một lĩnh vực quan trọng trong Xử lý ảnh và Thị giác máy tính, với mục tiêu xác định và phân loại các đối tượng cụ thể trong hình ảnh hoặc video. Đây là một bài toán có ứng dụng rộng rãi, từ giám sát an ninh, y tế, giao thông, đến tự động hóa trong công nghiệp.

Ví dụ minh họa bài toán nhận dạng:

* Nhận diện khuôn mặt người trong hình ảnh chụp đám đông.
* Nhận diện biển số xe
* Phát hiện biển báo giao thông trên đường.
* Xác định các sản phẩm trên băng chuyền trong dây chuyền sản xuất.

Hình ảnh minh họa:  
A screenshot of a computer

Description automatically generated

Hình 2: Ảnh minh họa chương trình nhận diện biển số xe

**Các công việc chính trong bài toán nhận dạng đối tượng trong ảnh**

Để nhận dạng một đối tượng trong ảnh, hệ thống cần thực hiện các bước cơ bản sau:

* **Thu thập dữ liệu (Data Collection):**
  + Thu thập các hình ảnh hoặc video chứa đối tượng cần nhận diện từ nhiều nguồn khác nhau (camera, internet, v.v.).
  + Đảm bảo dữ liệu đủ đa dạng về ánh sáng, góc độ, kích thước và môi trường.
* **Tiền xử lý ảnh (Image Preprocessing):**
  + Xử lý dữ liệu ảnh để chuẩn bị cho các bước tiếp theo, bao gồm:
    - Chuyển đổi ảnh về dạng xám (Grayscale).
    - Cân bằng độ sáng, tương phản (Histogram Equalization).
    - Lọc nhiễu (Noise Reduction).
    - Cắt ảnh (Cropping) để tập trung vào đối tượng chính.
* **Phát hiện đối tượng (Object Detection):**
  + Sử dụng các kỹ thuật phát hiện đối tượng để xác định vị trí đối tượng trong ảnh.
  + Ví dụ:
    - Sử dụng các phương pháp dựa trên ngưỡng (Thresholding).
    - Sử dụng mô hình học sâu như YOLO, Faster R-CNN, hoặc SSD.
* **Trích xuất đặc trưng (Feature Extraction):**
  + Trích xuất các đặc trưng quan trọng từ đối tượng, ví dụ:
    - Đường biên (Edges).
    - Đặc điểm hình dạng (Shape).
    - Văn bản (Text) hoặc ký tự nếu có.
* **Nhận dạng (Recognition):**
  + Áp dụng các thuật toán nhận dạng để phân loại đối tượng hoặc giải mã thông tin.
  + Với biển số xe, bước này bao gồm nhận diện ký tự và chuyển chúng thành chuỗi văn bản.
* **Hậu xử lý và hiển thị kết quả (Post-Processing and Result Display):**
  + Xử lý kết quả nhận dạng để đưa ra thông tin hữu ích.
  + Ví dụ: Hiển thị chuỗi biển số xe trên màn hình hoặc lưu kết quả vào cơ sở dữ liệu.

## **2.2 Các kỹ thuật sử dụng trong bài toán nhận dạng biển số xe**

Bài toán nhận dạng biển số xe sử dụng một số kỹ thuật trong xử lý ảnh và nhận dạng ký tự. Dưới đây là mô tả chi tiết từng kỹ thuật, bao gồm tư tưởng, minh họa, và ưu nhược điểm:

**1. Xử lý ảnh (Preprocessing)**

* **Tư tưởng kỹ thuật:** Xử lý ảnh nhằm chuẩn hóa và làm nổi bật các đặc điểm quan trọng để nhận diện biển số dễ dàng hơn. Gồm các bước:
  + **Chuyển đổi sang thang xám (Grayscale):** Loại bỏ thông tin màu, chỉ giữ lại cường độ sáng, giảm kích thước dữ liệu.
  + **Làm mờ (Blurring):** Dùng Gaussian Blur để làm mượt ảnh, giảm nhiễu, giúp xác định viền tốt hơn.
  + **Phát hiện biên (Edge Detection):** Dùng thuật toán Canny để tìm các đường biên, làm nổi bật vùng có khả năng chứa biển số xe.
* **Hình minh họa:**
  + Ảnh gốc:

A license plate on a motorcycle

Description automatically generated

Hình 3: Ảnh minh họa ảnh gốc

* Ảnh thang xám:

A license plate on a motorcycle

Description automatically generated

Hình 4: Ảnh minh họa chuyển đổi sang thang xám

* Ảnh sau khi phát hiện biên:

A black and white image of a sign

Description automatically generated

Hình 5: Ảnh minh họa sau khi phát hiện biên

* **Ưu điểm:**
  + Giảm nhiễu, làm nổi bật các đặc điểm quan trọng.
  + Chuẩn bị dữ liệu tốt hơn cho các bước xử lý sau.
* **Nhược điểm:**
  + Độ chính xác bị ảnh hưởng nếu ảnh đầu vào có nhiều nhiễu.
  + Phụ thuộc vào tham số của các thuật toán (như Gaussian Blur và Canny).

**2. Phát hiện vùng biển số xe (Region Detection)**

* **Tư tưởng kỹ thuật:** Dùng thuật toán tìm đường viền (**Contours**) để phát hiện các vùng hình chữ nhật, từ đó xác định khu vực chứa biển số xe.
  + Tìm các đường viền lớn nhất trong ảnh.
  + Kiểm tra các đường viền có hình dạng là tứ giác (biển số xe thường có dạng hình chữ nhật).
* **Hình minh họa:**
  + Vùng biển số được phát hiện và khoanh vùng:

A license plate on a motorcycle

Description automatically generated

Hình 6: Ảnh minh họa phát hiện vùng biển số và khaonh vùng

* **Ưu điểm:**
  + Hiệu quả cao khi ảnh đầu vào có độ tương phản tốt.
  + Không cần dữ liệu huấn luyện.
* **Nhược điểm:**
  + Độ chính xác giảm nếu ảnh bị che khuất, biển số bị méo.
  + Phụ thuộc nhiều vào chất lượng ảnh.

**3. Nhận dạng ký tự (Optical Character Recognition - OCR)**

* **Tư tưởng kỹ thuật:** Sử dụng thư viện **EasyOCR** để nhận diện các ký tự trên vùng biển số đã được cắt.
  + OCR dựa trên các mô hình học sâu để nhận diện ký tự trong hình ảnh.
* **Hình minh họa:**
  + Kết quả nhận dạng ký tự từ biển số:

A screenshot of a computer

Description automatically generated

Hình 7: Ảnh minh họa kết quả nhận dạng ký tự từ biển số

* **Ưu điểm:**
  + Dễ sử dụng, hỗ trợ nhiều ngôn ngữ.
  + Độ chính xác cao với ảnh rõ ràng.
* **Nhược điểm:**
  + Hiệu suất giảm với ảnh có chất lượng thấp hoặc biển số bị che khuất.
  + Tốn tài nguyên tính toán hơn các phương pháp truyền thống.

**4. Tra cứu tỉnh/thành phố từ mã biển số**

* **Tư tưởng kỹ thuật:** Lấy hai chữ số đầu tiên từ kết quả nhận diện ký tự (biển số xe), sau đó tra cứu trong từ điển mã tỉnh/thành phố để xác định địa phương.
* **Ưu điểm:**
  + Dễ triển khai, không cần thuật toán phức tạp.
  + Tăng tính trực quan trong ứng dụng.
* **Nhược điểm:**
  + Phụ thuộc hoàn toàn vào độ chính xác của OCR.
  + Không nhận diện được nếu biển số bị sai ký tự hoặc không đầy đủ.

## **2.3. Xây dựng hệ thống nhận diện biển số xe**

### **2.3.1. Tiền xử lý ảnh**

Trong phần này, trình bày các bước xử lý ảnh trước khi đưa vào các thuật toán phát hiện và nhận diện. Các bước cơ bản bao gồm:

**1. Đọc và chuyển đổi định dạng ảnh**

* + Ảnh được đọc bằng OpenCV (cv2.imread) ở định dạng BGR (Blue-Green-Red).
  + Chuyển đổi từ BGR sang RGB để phù hợp với định dạng hiển thị trong giao diện Qt (cv2.cvtColor).

**2. Chọn vùng ROI (Region of Interest)**

* + Người dùng có thể chọn vùng chứa biển số xe thủ công bằng cách sử dụng (cv2.selectROI). Điều này giúp tập trung vào vùng quan trọng, giảm nhiễu từ các phần khác của ảnh.

**3. Chuẩn bị dữ liệu ảnh**

* + Kích thước ảnh có thể được điều chỉnh sao cho phù hợp với mô hình OCR hoặc các bước nhận diện tiếp theo.
  + Chuyển ảnh từ định dạng NumPy array sang định dạng Qt để hiển thị trong giao diện người dùng.

**4. Tiền xử lý ảnh trong vùng ROI**

* + Sử dụng ảnh cắt từ ROI làm đầu vào cho OCR.
  + (Có thể mở rộng) Thêm các bước như chuyển đổi ảnh sang grayscale, làm mịn ảnh, hoặc tăng cường độ tương phản nếu cần thiết.

### **2.3.2. Phát hiện biển số xe bằng OpenCV**

Để thực hiện nhận diện biển số xe từ hình ảnh, chương trình được xây dựng với các bước như sau:

**1. Chuẩn bị dữ liệu và giao diện:**

* + **Giao diện người dùng (UI):** Tệp trangchu.ui được sử dụng để thiết kế giao diện với các thành phần như:
* lbl\_anh: Hiển thị hình ảnh được chọn.
* btn\_anh: Nút chọn ảnh để thực hiện xử lý.
* txt\_bienso: Hiển thị văn bản biển số xe nhận diện được.
* txt\_tinh\_tp: Hiển thị thông tin tỉnh/thành phố dựa trên mã biển số.
  + **Dữ liệu mã tỉnh/thành phố:** Một từ điển province\_dict được tạo để ánh xạ mã biển số (hai chữ số đầu) với tên tỉnh/thành phố tương ứng.

**2. Các bước xử lý chính:**

**a. Chọn và hiển thị hình ảnh:**

* + Người dùng chọn một tệp hình ảnh thông qua hộp thoại (QFileDialog).
  + Hình ảnh được chuyển từ định dạng BGR (OpenCV) sang RGB (Qt) để tương thích với việc hiển thị trên giao diện.
  + Hình ảnh sau đó được hiển thị trên thành phần lbl\_anh với tỉ lệ giữ nguyên.

**b. Nhận diện biển số tự động:**

* + Sử dụng thư viện **EasyOCR** để đọc văn bản từ hình ảnh.
  + Dựa vào kết quả nhận diện, chương trình:
* Lấy 2 ký tự đầu của biển số làm mã tỉnh/thành phố.
* So khớp mã này với từ điển province\_dict để hiển thị tên tỉnh/thành phố.

**c. Lựa chọn và nhận diện vùng ROI:**

* + Người dùng có thể chọn vùng quan tâm (ROI) từ ảnh gốc bằng công cụ (cv2.selectROI).
  + Sau khi chọn vùng ROI, chương trình thực hiện nhận diện văn bản trong vùng đó.
  + Nếu nhận diện được, biển số và tên tỉnh/thành phố tương ứng sẽ được hiển thị.

**3. Kết quả xử lý:**

* + Biển số xe được hiển thị tại txt\_bienso.
  + Tên tỉnh/thành phố dựa trên mã biển số được hiển thị tại txt\_tinh\_tp.
  + Nếu không nhận diện được văn bản, chương trình sẽ thông báo "Không nhận diện được biển số xe" hoặc "Không nhận diện được tỉnh thành."

# **CHƯƠNG III: THỰC NGHIỆM**

## **5.1. Ngôn ngữ lập trình và các thư viện**

### **5.1.1. Ngôn ngữ lập trình Python**

**Python** là một ngôn ngữ lập trình hướng đối tượng, cấp cao và vô cùng mạnh mẽ. Điều đặc biệt ở ngôn ngữ này là sự đa năng, đa công dụng. Mang các tính năng ưu việt như tạo kiểu dữ liệu tự động và cơ chế cung cấp bộ nhớ tự động. Ngoài ra, Python còn được coi là một ngôn ngữ lập trình có cấu trúc dữ liệu cao cấp và mạnh mẽ nhất.

Các tính năng chính thể hiện sự ưu việt của Python có thể liệt kê như:

* **Cực kỳ đơn giản và dễ tiếp thu** cũng như thực hành: sự đơn giản của ngôn ngữ này thể hiện ở các câu lệnh dễ học, dễ hiểu. Python biến những “cú pháp code khó nhằn” trở nên dễ dàng hơn bằng những giải pháp.
* **Hoàn toàn miễn phí và là mã nguồn mở**: nhà lập trình có thể thoải mái di chuyển Python thậm chí sử dụng nó với mục đích thương mại hóa. Với đặc trưng là một ngôn ngữ lập trình mã nguồn mở. Bạn hoàn toàn có thể thay đổi các mã nguồn mở của Python ở bất cứ các ứng dụng phần mềm nào.
* **Là một ngôn ngữ di động:** di động có nghĩa là nó rất dễ di chuyển từ nền tảng này sang các nền tảng khác. Đồng thời, bạn có thể chạy chương trình của Python ở mọi nền tảng mà không sợ có bất kỳ những thay đổi nào ảnh hưởng.
* **Python là ngôn ngữ phiên dịch cao cấp:**khi chạy chương trình Python trên bất cứ hệ điều hành hay mạng Internet nào. Python có thể hoàn toàn tự động dịch và chạy theo đúng chương trình dữ liệu của máy tính đó.

### **5.1.2. Các thư viện được sử dụng**

**5.1.2. Thư viên OpenCV**

**OpenCV** (Open Source Computer Vision Library) là một thư viện mã nguồn mở được phát triển để hỗ trợ xử lý và phân tích hình ảnh cũng như video trong lĩnh vực thị giác máy tính và máy học. Được ra đời vào năm 2000 bởi Intel, OpenCV ngày nay là một trong những công cụ phổ biến nhất trong cộng đồng lập trình viên và nhà nghiên cứu liên quan đến xử lý ảnh.

**Các tính năng chính của OpenCV**

* **Xử lý ảnh**:
  + Chuyển đổi ảnh giữa các không gian màu (RGB, HSV, Grayscale,...).
  + Lọc ảnh: Làm mịn, làm sắc nét, giảm nhiễu.
  + Phát hiện biên cạnh, phát hiện đường thẳng, đường cong.
* **Thị giác máy tính**:
  + Phát hiện và nhận diện khuôn mặt, mắt, cử chỉ.
  + Nhận diện và theo dõi vật thể trong video.
  + Xử lý video: Theo dõi chuyển động, phân tích khung hình.
* **Học máy (Machine Learning)**:
  + Tích hợp các thuật toán như SVM, k-NN, và các bộ phân loại.
  + Hỗ trợ nhận diện chữ viết tay, nhận diện ký tự quang học (OCR).
* **Hỗ trợ phần cứng**:
  + Tăng tốc xử lý bằng GPU (hỗ trợ CUDA của NVIDIA).
  + Tối ưu hóa cho các thiết bị nhúng như Raspberry Pi, Jetson Nano.

**Ưu điểm của OpenCV**

* **Mã nguồn mở và miễn phí**: Thích hợp cho nghiên cứu, phát triển và ứng dụng.
* **Hỗ trợ đa nền tảng**: Chạy trên Windows, Linux, macOS, Android, iOS, và các thiết bị nhúng.
* **Tích hợp nhiều ngôn ngữ lập trình**: Python, C++, Java, MATLAB,...
* **Thư viện phong phú**: Tích hợp nhiều thuật toán tối ưu.

**Ứng dụng thực tế của OpenCV**

* **Nhận diện khuôn mặt và cử chỉ**: Dùng trong các hệ thống bảo mật, camera thông minh.
* **Phân tích y tế**: Xử lý ảnh MRI, X-quang để phát hiện bệnh.
* **Xử lý hình ảnh trong công nghiệp**: Kiểm tra chất lượng sản phẩm, robot tự động.
* **Phát triển xe tự hành**: Xử lý và phân tích hình ảnh từ camera của xe.
* **Ứng dụng thực tế tăng cường (AR)**: Tích hợp hình ảnh ảo vào môi trường thực.

**5.1.2.2. Thư viên EasyOCR**

**EasyOCR** là một thư viện mã nguồn mở được phát triển để nhận diện ký tự quang học (OCR - Optical Character Recognition) trong các hình ảnh và tài liệu. Được xây dựng trên các framework học sâu như **PyTorch**, EasyOCR nổi bật với sự đơn giản, độ chính xác cao và khả năng hỗ trợ nhiều ngôn ngữ khác nhau.

**Tính năng chính của EasyOCR**

* **Hỗ trợ đa ngôn ngữ**:
  + EasyOCR hỗ trợ hơn **80 ngôn ngữ**, bao gồm tiếng Anh, tiếng Việt, Trung Quốc, Nhật Bản, Hàn Quốc và nhiều ngôn ngữ khác.
  + Dễ dàng thêm ngôn ngữ mới nếu cần.
* **Nhận diện ký tự từ nhiều định dạng**:
  + Có thể nhận diện chữ viết từ các hình ảnh, tài liệu quét, ảnh chụp màn hình hoặc ảnh trong tự nhiên.
* **Độ chính xác cao**:
  + Sử dụng mô hình học sâu (Deep Learning) để đạt độ chính xác vượt trội, đặc biệt trong các trường hợp có font chữ đa dạng hoặc nền phức tạp.
* **Dễ sử dụng**:
  + API đơn giản, dễ tích hợp vào các ứng dụng Python.
* **Hỗ trợ nhiều kiểu chữ**:
  + Nhận diện được chữ in, chữ viết tay (trong một số trường hợp), và chữ có phong cách đặc biệt.

**Ưu điểm của EasyOCR**

* **Đơn giản và nhanh chóng**: API dễ sử dụng, không cần quá nhiều cấu hình.
* **Độ chính xác cao**: Hiệu quả tốt ngay cả trên nền phức tạp.
* **Hỗ trợ đa ngôn ngữ**: Rất hữu ích cho các ứng dụng quốc tế.

**Nhược điểm**

* Hiệu suất có thể thấp hơn so với các dịch vụ OCR thương mại như Google Vision API hoặc Tesseract trong một số trường hợp.
* Yêu cầu tài nguyên mạnh mẽ để chạy mô hình học sâu, đặc biệt trên CPU.

## **5.2. Dữ liệu sử dụng**

Dữ liệu sử dụng trong bài toán nhận diện biển số xe bao gồm các hình ảnh chụp biển số xe từ nhiều nguồn khác nhau, đảm bảo đa dạng về điều kiện ánh sáng, góc chụp và chất lượng hình ảnh.

Các ảnh bảo gồm:

- Biển số xe có chất lượng cao, rõ ràng.

- Biển số xe bị mờ, bị nhiễu hoặc chụp trong điều kiện ánh sáng yếu.

- Biển số xe có nhiều định dạng khác nhau (1 hàng hoặc 2 hàng).

Ngoài ra, bài toán sử dụng thư viện EasyOCR để hỗ trợ nhận diện ký tự từ hình ảnh và một từ điển mã tỉnh/thành phố để xác định vị trí của biển số.

## **5.3. Kết quả thực nghiệm**

**Nhận diện biển số:**

Hệ thống cho kết quả nhận diện tốt với các biển số rõ ràng, nhưng giảm hiệu suất với các biển số bị mở hoặc có ánh sáng không đồng đều. Ví dụ:

- Ảnh rõ nét: Độ chính xác đạt trên 90%:

A screenshot of a white and black license plate

Description automatically generated

Hình 8: Ảnh biển số rõ nét

- Ảnh mờ hoặc điều kiện ánh sáng yếu: Độ chính xác giảm xuống khoảng 60-70%:

A close up of a car

Description automatically generated

Hình 9: Ảnh biển số mờ hoặc điều kiện ánh sáng yếu

**Nhận diện tỉnh/thành phố:**

Hệ thống dựa trên mã biển số xe để xác định tỉnh/thành phố. Kết quả cho thấy:

- Mã tỉnh/thành phố nhận diện chính xác đối với các biển số đầy đủ ký tự:

A motorcycle with a license plate

Description automatically generated

Hình 10: Kết quả mã tỉnh/thành phố đầy đủ ký tự

- Trong một số trường hợp biển số bị mờ hoặc mất một phần ký tự, mã tỉnh/thành phố không nhận diện được:

A license plate on a motorcycle

Description automatically generated

Hình 11: Kết quả khi mã tỉnh/thành phố mất đi 1 phần ký tự

**Minh họa kết quả:**

- Ảnh đầu vào:

The front of a car

Description automatically generated

Hình 12: Ảnh đầu vào

- Ảnh đã chọn ROI (Region of Interest):

A close up of a license plate

Description automatically generated

Hình 13: Ảnh đã chọn ROI

- Kết quả nhận diện:

A white car parked in a parking lot

Description automatically generated

Hình 14: kết quả nhận diện.

Hệ thống hoạt động hiệu quả nhất trong điều kiện ảnh đầu vào rõ ràng và được căn chỉnh đúng tỷ lệ.

# **KẾT LUẬN**

Hệ thống nhận diện biển số xe tự động sử dụng OpenCV đã được triển khai thành công, đáp ứng các mục tiêu đặt ra trong bài tập. Qua quá trình nghiên cứu và xây dựng hệ thống, các kết quả chính có thể được tổng kết như sau:3

1. **Tính khả thi và ứng dụng thực tiễn:**  
   Hệ thống đã chứng minh tính khả thi trong việc áp dụng xử lý ảnh và nhận diện ký tự để tự động phát hiện và trích xuất thông tin từ biển số xe. Đây là nền tảng quan trọng để phát triển các ứng dụng thực tế như quản lý giao thông, kiểm soát ra vào tại bãi đỗ xe, hoặc giám sát an ninh.
2. **Hiệu quả của OpenCV và EasyOCR:**

* **OpenCV** đã thể hiện vai trò vượt trội trong việc xử lý ảnh, phát hiện vùng biển số xe, và tiền xử lý ảnh để tăng chất lượng nhận diện.
* **EasyOCR** cho thấy độ chính xác cao trong việc nhận diện ký tự, ngay cả với các ký tự tiếng Việt và biển số xe có cấu trúc đặc biệt.

1. **Hạn chế và thách thức:**  
   Dù đạt được nhiều kết quả tích cực, hệ thống vẫn gặp một số hạn chế:

* Chất lượng ảnh đầu vào kém (mờ, nhiễu, hoặc ánh sáng không đồng đều) có thể làm giảm độ chính xác của nhận diện.
* Hệ thống cần được tối ưu hơn để xử lý nhanh trong trường hợp áp dụng cho luồng dữ liệu lớn hoặc thời gian thực.

**Hướng phát triển:**

* **Nâng cao độ chính xác:** Áp dụng các kỹ thuật học sâu tiên tiến hơn để cải thiện khả năng nhận diện trong các điều kiện phức tạp.
* **Xử lý thời gian thực:** Tích hợp hệ thống với camera để nhận diện trực tiếp và tối ưu tốc độ xử lý.
* **Mở rộng ứng dụng:** Xây dựng thêm các tính năng như phân loại phương tiện, giám sát hành trình, hoặc liên kết với cơ sở dữ liệu giao thông.

# **TÀI LIỆU THAM KHẢO**

[1]. Lê Mỹ Hà(2019), *Giáo trình thị giác máy tính và ứng dụng,* Nhà xuất bản Đại học quốc gia thành phố Hồ Chí Minh.

[2]. Nguyễn Quang Hoan(2006), *Xử lý ảnh*, Nhà xuất bản Học viện Bưu chính viễn thông.