BAN CƠ YẾU CHÍNH PHỦ

HỌC VIỆN KỸ THUẬT MẬT MÃ

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

ĐỀ TÀI

XÂY DỰNG HỆ THỐNG KIỂM SOÁT VÀO RA TẠI HỌC VIỆN KỸ THUẬT MẬT MÃ

Sinh viên thực hiện: Phạm Thị Phương Anh

Khóa: CT4

Chuyên ngành: Công nghệ thông tin

Mã ngành: 748.02.01

Người hướng dẫn: ThS. Lê Đức Thuận

ThS.Mạc Văn Hải

Hà Nội - 2024

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

ĐỀ TÀI
XÂY DỰNG HỆ THỐNG KIỂM SOÁT VÀO RA TẠI HỌC VIỆN KỸ THUẬT MẬT MÃ

Sinh viên thực hiện: Phạm Thị Phương Anh
Khóa: CT4
Chuyên ngành: Công nghệ thông tin
Mã ngành: 748.02.01
Người hướng dẫn: ThS. Lê Đức Thuận
ThS.Mạc Văn Hải

Hà Nội - 2024

LỜI CẢM ƠN

Trước tiên em xin gửi lời cảm ơn chân thành, sâu sắc tới quý thầy cô trong khoa Công nghệ thông tin nói riêng và toàn thể thầy cô của Học viện Kỹ thuật mật mã nói chung đã truyền đạt cho em những kiến thức, những kinh nghiệm quý báu trong suốt 5 năm học qua.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc tới Th.S Lê Đức Thuận và Th.S Mạc Văn Hải - giảng viên hướng dẫn đồ án tốt nghiệp. Hai thầy đã dành nhiều thời gian của mình để định hướng đề tài, tận tình tư vấn, chỉ bảo hướng dẫn và góp ý trong quá trình làm đồ án cũng như tạo mọi điều kiện thuận lợi để em có thể hoàn thành tốt đồ án của mình.

Em xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp đã định hướng, động viên, hỗ trợ, góp ý và tạo điều kiện thuận lợi để em hoàn thành đồ án.

Mặc dù đã cố gắng, nỗ lực trong quá trình làm đồ án nhưng do kiến thức còn hạn hẹp nên báo cáo không tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến, chỉ bảo của quý thầy cô và các bạn.

Cuối cùng em xin kính chúc thầy cô sức khỏe, đạt được nhiều thành công trong sự nghiệp trồng người cũng như trong nghiên cứu khoa học!

Em xin chân thành cảm ơn!

SINH VIÊN THỰC HIỆN

Phạm Thị Phương Anh

LỜI CAM ĐOAN

Em xin cam đoan đồ án này do em tự nghiên cứu dưới sự hướng dẫn của ThS. Lê Đức Thuận và ThS. Mạc Văn Hải.

Để hoàn thành đồ án này, em chỉ sử dụng những tài liệu đã ghi trong mục tài liệu tham khảo, ngoài ra không sử dụng bất cứ tài liệu nào khác mà không được ghi.

Nếu sai, em xin chịu mọi hình thức kỷ luật theo quy định của Học viện.

Hà Nội, ngày 07 tháng 06 năm 2024

Sinh viên thực hiện

(Ký tên và ghi rõ họ tên)

Phạm Thị Phương Anh

MỤC LỤC

Trang

Lời cảm ơn i
Lời cam đoan ii
Mục lục iii
Danh mục hình vẽ v
Danh mục bảng biểu viii
Danh mục từ viết tắt ix
Mở đầu 1

Chương 1
TỔNG QUAN ĐỀ TÀI

1.1. Thực trạng tại các trường đại học 3
1.1.1. Khảo sát hiện trạng 3
1.1.2. Yêu cầu đề xuất 5
1.2. Công nghệ xử lý nhận diện ảnh 6
1.2.1. Nhận dạng ký tự quang học 6
1.2.2. Mô hình Convolutional Neural Networks 8
1.2.3. Giải thuật Canny 10
1.3. Mô hình YOLOv8 13
1.3.1. Khái quát về hệ thống phát hiện đối tượng 13
1.3.2. Giới thiệu YOLOv8 16
1.3.3. YOLOv8 và sự phát triển từ phiên bản trước 16
1.3.4. Ưu điểm và hạn chế của YOLOv8 17
1.3.5. Các bước huấn luyện và ứng dụng của YOLOv8 18
1.4. Tìm hiểu về các thiết bị nhúng sử dụng trong hệ thống 20
1.4.1. Giới thiệu chung về máy tính nhúng Raspberry 20
1.4.2. Camera USB 22
1.4.3. Động Cơ Servo SG90S 23
1.4.4. Cảm biến vật cản hồng ngoại 23
1.5. Tìm hiểu về Flask Framework 24
1.5.1. Giới thiệu chung về Flask Framework 24
1.5.2. Tính năng của Flask Framework 25

1.5.3. Mjpg-streamer 26

Chương 2
THIẾT KẾ XÂY DỰNG HỆ THỐNG VÀO RA TẠI HỌC VIỆN

2.1. Phân tích chức năng 27
2.1.1. Chức năng của hệ thống 27
2.1.2. Yêu cầu phi chức năng 31
2.1.3. Biểu đồ 31
2.2. Thiết kế hệ thống 35
2.2.1. Phần cứng 35
2.2.2. Phần mềm 36
2.3. Phương pháp nhận diện biển số xe 38
2.3.1. Xác định vị trí trong ảnh 38
2.3.2. Xử lý hình ảnh 46
2.3.3. Sử dụng YOLOv8 để xác định biển số xe 46

Chương 3
THỰC NGHIỆM

3.1. Các chức năng 48
3.1.1. Chức năng đăng nhập 48
3.1.2. Chức năng nhận diện biển số xe và phát video lên hệ thống 49
3.1.3. Chức năng xem lịch sử xe ra vào 50
3.1.4. Chức năng quản lý người dùng 50
3.1.5. Chức năng thống kê 52
3.1.6. Chức năng tải dữ liệu dưới dạng CSV 53
3.2. Hệ thống phần cứng 56
3.3. Đánh giá 57
KẾT LUẬN 60
TÀI LIỆU THAM KHẢO 62
PHỤ LỤC . . . . . .

DANH MỤC HÌNH VẼ

Trang

Hình 1.1. Hệ thống bãi đỗ xe tại trường Đại học Công nghệ Đồng Nai [8] 3
Hình 1.2. Hệ thống bãi đỗ xe tại trường Đại học Bách Khoa Hà Nội . . 4
Hình 1.3. Hệ thống bãi đỗ xe ghi vé giấy 4
Hình 1.4. Hệ thống bãi đỗ xe ghi bằng phấn 5
Hình 1.5. Hoạt động của OCR [16] 6
Hình 1.6. Cấu trúc của mô hình CNN [1] 8
Hình 1.7. Phép Convolution 9
Hình 1.8. Max Pooling [12] 10
Hình 1.9. Average Pooling [12] 10
Hình 1.10. Góc của θ trong thuật toán Canny 13
Hình 1.11. Mô hình YOLOv8 phát hiện và dự đoán hộp giới hạn 14
Hình 1.12. Kiến trúc hệ thống HOG để phát hiện đối tượng. 14
Hình 1.13. Chu trình phát hiện đối tượng với Mạng nơ-ron tích chập dựa trên khu vực (R-CNN) 15
Hình 1.14. Giải thuật Selective Search 15
Hình 1.15. Nhận dạng đối tượng bằng Selective Search 15
Hình 1.16. Minh họa mô hình Faster R-CNN 16
Hình 1.17. Dòng thời gian phát triển của YOLO 17
Hình 1.18. So sánh độ chính xác và hiệu suất của các mô hình YOLO . . 18
Hình 1.19. Máy tính nhúng Raspberry Pi 20
Hình 1.20. Raspberry Pi 3 Model B+ [3] 22
Hình 1.21. Camera USB 22
Hình 1.22. Động Cơ Servo SG90S [13] 23
Hình 1.23. Cảm biến hồng ngoại [9] 24
Hình 1.24. Video stream lên trên trình duyệt 26
Hình 2.1. Biểu đồ Usecase tổng quát 32
Hình 2.2. Biểu đồ hoạt động chức năng đăng nhập 32
Hình 2.3. Biểu đồ hoạt động thêm người dùng 33
Hình 2.4. Biểu đồ hoạt động xem lịch sử ra vào 33
Hình 2.5. Biểu đồ hoạt động chức năng xóa 34
Hình 2.6. Biểu đồ hoạt động chức nhận diện biển số xe 34

Hình 2.7. Biểu đồ cơ sở dữ liệu 35
Hình 2.8. Mô hình đi chân của các thiết bị phần cứng 36
Hình 2.9. Kiến trúc phần mềm hệ thống 36
Hình 2.10. Hoạt động xử lý hình ảnh tại bộ sử lý trung tâm (PC) 37
Hình 2.11. Hoạt động xử lý tại Raspberyy Pi 38
Hình 2.12. Gán nhãn cho vị trí biển số xe trên roboflow 39
Hình 2.13. Gán nhãn cho số trong biển số xe trên roboflow 39
Hình 2.14. Huấn luyện dữ liệu bằng lệnh trên cmd 40
Hình 2.15. Biểu đồ confusion matrix cho nhận dạng vị trí biển số xe . . . 41
Hình 2.16. Biểu đồ confusion matrix cho nhận dạng ký tự trong biển số xe 41
Hình 2.17. Biểu đồ F1-Confidence Curve cho nhận diện vị trí biển số xe . 42
Hình 2.18. Biểu đồ F1-Confidence Curve cho nhận dạng số trong biển số xe 43
Hình 2.19. Biểu đồ Labels Correlogram cho nhận diện vị trí biển số xe . . 43
Hình 2.20. Biểu đồ Labels Correlogram cho nhận diện số trong biển số xe 44
Hình 2.21. Biểu đồ P Curve cho nhận diện vị trí biển số xe 44
Hình 2.22. Biểu đồ P Curve cho nhận diện số trong biển số xe 45
Hình 2.23. Biểu đồ Results cho nhận diện vị trí biển số xe 45
Hình 2.24. Biểu đồ Results cho nhận diện số trong biển số xe 46
Hình 2.25. Ảnh xác định vị trí biển số xe 47
Hình 2.26. Biểu số xe sau khi đã xác định số 47
Hình 3.1. Giao diện đăng nhập của hệ thống 48
Hình 3.2. Giao diện đăng nhập sai 48
Hình 3.3. Giao diện xe đi vào 49
Hình 3.4. Giao diện xe đi ra 49
Hình 3.5. Giao diện xem trạng thái ra vào 50
Hình 3.6. Giao diện thêm người dùng 51
Hình 3.7. Giao diện xóa người dùng 51
Hình 3.8. Giao diện chỉnh sửa người dùng 52
Hình 3.9. Giao diện thống kê theo ngày 52
Hình 3.10. Giao diện thống kê theo tháng 53
Hình 3.11. Giao diện thống kê số ngày làm việc theo tháng 53
Hình 3.12. Giao diện tải dữ liệu dưới dạng CSV 54

Hình 3.13. Mở dữ liệu đã tải về ở dạng CSV bảng thống kê số ngày làm việc trong tháng 54
Hình 3.14. Mở dữ liệu đã tải về ở dạng CSV bảng người dùng 54
Hình 3.15. Mở dữ liệu đã tải về ở dạng CSV bảng thống kê theo ngày . . 55
Hình 3.16. Mở dữ liệu đã tải về ở dạng CSV bảng lịch sử ra vào 55
Hình 3.17. Mở dữ liệu đã tải về ở dạng CSV bảng thống kê theo tháng . . 56
Hình 3.18. Mô hình sản phẩm 56
Hình 3.19. Hình ảnh hiển thị biển số xe lên LCD 57
Hình 3.20. Hình ảnh cảm biến vật cản 57

DANH MỤC BẢNG BIỂU

Trang

Bảng 2.1. Bảng phân tích chức năng đăng nhập 27
Bảng 2.2. Bảng phân tích chức năng thêm người dùng 28
Bảng 2.3. Bảng phân tích chức năng xem lịch sử ra vào 28
Bảng 2.4. Bảng phân tích chức năng sửa thông tin người dùng 29
Bảng 2.5. Bảng phân tích chức năng xóa người dùng 29
Bảng 2.6. Bảng phân tích chức năng thống kê theo ngày, tháng 30
Bảng 2.7. Bảng phân tích chức năng tải dữ liệu thống kê 30
Bảng 2.8. Bảng phân tích chức năng nhận diện biển số xe 31
Bảng 3.1. Kết quả đánh giá chức năng với hệ thống phần cứng 59

DANH MỤC TỪ VIẾT TẮT

Viết tắt	Tiếng anh	Tiếng việt
CNN	Convolutional Neural Networks	Mạng nơ-ron tích chập
GPU	Graphics processing unit	Bộ phận xử lý đồ họa
HOG	Histogram of Oriented Gradients	Phương pháp mô tả đặc trưng
IoT	Internet of Things	Internet vạn vật
IP	Internet Protocol	Giao thức Internet
OCR	Optical Character Recognition	Nhận dạng ký tự quang học
RPN	Region Proposal Network	Mạng đề xuất vùng
YOLO	You Only Look Once	

MỞ ĐẦU

Trong môi trường học tập - nơi học sinh, sinh viên và các cán bộ nhân viên di chuyển hàng ngày với lưu lượng xe vô cùng đông đúc, việc quản lý giữ xe trở nên khó khăn. Đặc biệt, vào những giờ cao điểm, khi hàng loạt sinh viên và nhân viên ra vào trường theo các tiết học, phương pháp truyền thống trong việc giữ xe trở nên không hiệu quả. Những vấn đề như mất vé, tốn nhiều thời gian và chi phí nhân lực, cùng với khả năng xử lý khó khăn khi có sự cố đã gây cho hệ thống này trở nên không linh hoạt.

Vì những thách thức này, nhiều trường đại học hiện nay đã chuyển đổi sang sử dụng hệ thống bãi đỗ xe thông minh. Hệ thống này không chỉ giúp giải quyết các vấn đề cấp thiết ở trên mà còn đem lại nhiều ưu điểm khác như tăng cường an ninh và an toàn cộng đồng học đường. Thay vì phải mất thời gian đáng kể cho quá trình giữ xe, hệ thống sẽ hoạt động hiệu quả và giảm áp lực cho nhân viên, tối ưu hóa việc sử dụng không gian bãi giữ xe.

Hệ thống bãi đỗ xe thông minh không chỉ đơn giản là giải pháp quản lý giữ xe mà còn mang lại hiệu suất cao trong việc tổ chức và điều phối lưu thông xe. Việc tích hợp công nghệ như hệ thống nhận diện biển số xe, quản lý thông tin đăng ký xe, kết nối dữ liệu trực tuyến giúp tối ưu hóa quá trình giữ xe và nâng cao trải nghiệm người dùng. Đồng thời nó còn tăng cường an ninh và giảm thiểu rủi ro xảy ra vấn đề như mất xe.

Do đó, em đã chọn đề tài "Xây dựng hệ thống kiểm soát vào ra tại Học viện Kỹ thuật mật mã". Đề tài trên được thực hiện với 3 chương như sau:

Chương 1: Tổng quan.

Chương mở đầu khảo sát hệ thống bãi đỗ xe tại các trường đại học hiện nay, từ đó đề xuất các yêu cầu cho hệ thống mới. Chương này cũng giới thiệu khái niệm, nghiên cứu về các thư viện và thiết bị cần thiết cho dự án, bao gồm công nghệ như mô hình CNN trong phát hiện đối tượng, YOLOv8, ngôn ngữ lập trình sử dụng và máy tính Raspberry. Qua đó, chương xác định nền tảng và cơ sở lý thuyết cho việc xây dựng hệ thống kiểm soát vào ra tại Học viện Kỹ thuật mật mã.

Chương 2: Thiết kế xây dựng hệ thống quản lý ra vào tại Học viện.

Chương hai trình bày chi tiết về thiết kế và triển khai của hệ thống kiểm soát vào ra. Chương này phân tích các chức năng quan trọng của hệ thống và mô tả quy trình thực hiện nhận diện biển số xe thông minh. Tập trung vào quy trình xây dựng hệ thống quản lý vào ra tại Học viện, chương hai cung cấp cái nhìn rõ ràng về cách giải quyết vấn đề mà bài toán đã đặt ra.

Chương 3: Thực nghiệm.

Chương ba trình bày các chức năng đã được thực hiện trong đề tài, từ đó đưa ra kết quả thực nghiệm, đánh giá về hiệu năng và độ chính xác trong nhận diện biển số xe.

Tuy nhiên, do thời gian thực hiện đồ án có hạn và kiến thức còn nhiều hạn chế, nên đề tài vẫn còn nhiều thiếu sót, nhiều điểm bất cập và các vấn đề chưa thể xử lý triệt để. Em rất mong nhận được sự góp ý của quý thầy cô và các bạn để báo cáo hoàn thiện hơn.

Em xin chân thành cảm ơn!

Chương 1
TỔNG QUAN ĐỀ TÀI

1.1. Thực trạng tại các trường đại học

1.1.1. Khảo sát hiện trạng

Bãi gửi xe tại các trường đại học trong nước ta là một phần không thể thiếu và đóng vai trò quan trọng trong việc giải quyết vấn đề giao thông và đảm bảo an ninh cho sinh viên, giảng viên và nhân viên nhà trường. Tuy nhiên, tình trạng của các bãi gửi xe tại các trường đại học có thể đa dạng và thay đổi tùy theo điều kiện cụ thể của từng trường.

Một số trường có hệ thống bãi gửi xe rộng rãi và hiện đại, với đủ chỗ đỗ và các tiện ích như mái che, hệ thống camera giám sát, và nhân viên quản lý chuyên nghiệp. Những trường này thường đảm bảo được sự an toàn và thuận tiện cho người sử dụng, đồng thời giảm thiểu tắc đường và hỗ trợ việc duy trì môi trường sạch sẽ.

Trường Đại học Công nghệ Đồng Nai là một trong những trường đào tạo nhiều ngành nghề với cơ sở vật chất hiện đại. Song song với đó, nhà trường cũng chú trọng việc xây dựng hệ thống kiểm soát ra vào, kiểm soát an ninh hiện đại, mỗi xe sẽ được cấp một thẻ có mã số riêng được thể hiện như hình 1.1. Quá trình ra vào đều được kiểm tra thông qua đầu đọc thẻ. Đồng thời dữ liệu hiển thị trên màn hình và cho biết xe ra có đúng với xe đăng ký vào hay không [8].

Hình 1.1: Hệ thống bãi đỗ xe tại trường Đại học Công nghệ Đồng Nai [8]

Hệ thống thống bãi đỗ xe tại trường Đại học Bách khoa Hà Nội cũng đã áp dụng phương pháp kiểm soát xe ra vào thông minh bằng thẻ RFID như hình 1.2.

Hình 1.2: Hệ thống bãi đỗ xe tại trường Đại học Bách Khoa Hà Nội

Tuy nhiên, tại một số trường đại học hình thức quản lý bãi gửi xe thông qua vé giấy được như hình 1.3 hoặc hình thức quản lý xe bằng phương pháp ghi phấn như hình 1.4 vẫn phổ biến. Mặc dù đây là phương pháp truyền thống, nhưng nó vẫn gặp phải một số hạn chế. Việc sử dụng vé giấy có thể gây ra sự bất tiện trong quá trình ghi nhận thông tin xe, đồng thời tạo điều kiện cho việc mất kiểm soát và an ninh không gian bãi gửi xe.

Hình 1.3: Hệ thống bãi đỗ xe ghi vé giấy

Hình 1.4: Hệ thống bãi đỗ xe ghi bằng phấn

1.1.2. Yêu cầu đề xuất

Qua khảo sát trên, nhận thấy rằng quản lý xe tự động đang nổi lên như một yếu tố quan trọng, đóng góp vào việc cải thiện hiệu suất và tạo ra những tiện ích đáng kể cho cho sinh viên và nhà trường. Tích hợp các hệ thống tự động hóa trong sắp xếp và quản lý xe không chỉ giúp tối ưu hóa không gian đỗ xe mà còn mang lại nhiều lợi ích đa chiều.

Một trong những ưu điểm nổi bật của việc áp dụng tự động hóa trong quản lý xe tại các trường đại học là tăng hiệu suất và tiết kiệm thời gian. Hệ thống có khả năng quản lý thông tin đăng ký và xác nhận xe một cách hiệu quả. Thông tin như biển số và thời gian vào, ra bãi đỗ được ghi lại để tạo ra một cơ sở dữ liệu đầy đủ và chính xác. Đồng thời, tính an toàn cũng được cải thiện với khả năng giảm thiểu nguy cơ va chạm và tai nạn do con người gây ra trong quá trình di chuyển xe.

Quản lý thông tin và thanh toán cũng trở nên dễ dàng hơn thông qua hệ thống tự động. Dữ liệu về việc đỗ xe, thời gian sử dụng và các quy tắc quản lý có thể được ghi lại tự động. Việc giảm tắc nghẽn và ô nhiễm môi trường cũng là một lợi ích quan trọng của tự động hóa trong quản lý xe. Bằng cách tối ưu hóa lưu thông xe, hệ thống có thể giảm tắc nghẽn giao thông và đồng thời giảm lượng khí thải, đóng góp vào việc cải thiện chất lượng không khí xung quanh. Các dữ liệu liên quan đến việc sử dụng xe và trạng thái kỹ thuật cũng có thể được thu thập để hỗ trợ trong việc lập kế hoạch bảo trì định kỳ và dự đoán nhu cầu sửa chữa.

Ngoài ra, tích hợp với hệ thống thông tin trường đại học cũng là một yếu tố quan trọng. Hệ thống quản lý xe tự động có thể được kết nối với các hệ thống thông tin của trường, cung cấp thông tin đồng bộ và dễ dàng quản lý.

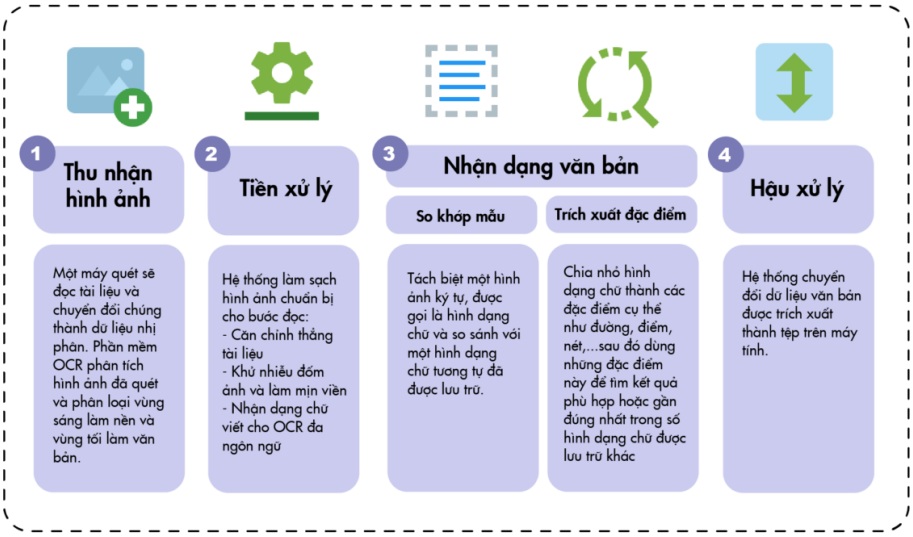
Trong khi đó, trải nghiệm người dùng cũng được cải thiện. Người dùng có thể dễ dàng và nhanh chóng tìm kiếm chỗ đỗ xe thông qua ứng dụng di động hoặc các thiết bị điều khiển, giảm bớt sự phiền toái và thời gian mất mát khi tìm kiếm chỗ đỗ.

Tóm lại, việc áp dụng tự động hóa trong quản lý xe tại các trường đại học ở Việt Nam không chỉ giúp tối ưu hóa các quy trình mà còn mang lại nhiều lợi ích về tiện ích, an toàn và bảo vệ môi trường.

1.2. Công nghệ xử lý nhận diện ảnh

1.2.1. Nhận dạng ký tự quang học

Optical Character Recognition (OCR) [14] là công nghệ nhận diện ký tự được xây dựng theo hệ thống mạng Neuron nhân tạo. Nó là kỹ thuật được sử dụng để chuyển đổi hình ảnh văn bản sang dạng văn bản có thể chỉnh sửa trong máy tính. OCR nhận đầu vào là các ảnh có kí tự riêng rẽ và cho ra kí tự văn bản tương ứng.



Hình 1.5: Hoạt động của OCR [16]

OCR có thể hoạt động [2] với cách thức khác nhau tùy theo điều kiện tích hợp trên hệ thống, tuy nhiên về cơ bản một phần mềm OCR sẽ làm việc theo các bước dưới đây:

Bước 1: Thu nhận hình ảnh

Phần mềm hoạt động như một máy quét đọc tài liệu và chuyển đổi chúng thành dữ liệu nhị phân sau đó công cụ OCR phân tích hình ảnh đã quét và phân loại vùng sáng làm nền và vùng tối làm văn bản.

Bước 2: Tiền xử lý

Hệ thống làm sạch hình ảnh chuẩn bị cho bước đọc tài liệu:

Căn chỉnh thẳng tài liệu đã quét để khắc phục những lỗi nghiêng hình ảnh nếu có Khử nhiễu hoặc loại bỏ những đốm ảnh và làm mịn viền hình ảnh cho văn bản Nhận dạng chữ viết cho OCR đa ngôn ngữ

Bước 3: Nhận dạng văn bản

Hai loại thuật toán được áp dụng để nhận dạng văn bản trên hệ thống là so khớp mẫu và trích xuất đặc điểm.

So khớp mẫu: Tách biệt một hình ảnh ký tự, sau đó so sánh với một hình ảnh ký tự đã được lưu trữ trước đó. Tính năng hoạt động hiệu quả khi hình dạng ký tự đầu vào trùng khớp với phông chữ và tỉ lệ của ký tự được lưu trữ trước đó.

Trích xuất đặc điểm: Chia nhỏ hoặc phân tách hình dạng ký tự thành các đặc điểm cụ thể như: các đường thẳng, các điểm giao, các nét vòng,v.v sau đó so sánh với các hình dạng chữ đã được lưu trữ để tìm kết quả phù hợp gần đúng nhất.

Bước 4: Hậu xử lý

Hệ thống chuyển đổi dữ liệu sau khi trích xuất thành tệp văn bản trên máy tính để phục vụ cho mục đích lưu trữ, tìm kiếm, chia sẻ, v.v

OCR được ứng dụng nhiều trong đời sống và trong nhiều công nghệ khác nhau như [16]:

Tích hợp vào máy scanner: Là ứng dụng thường gặp nhất của OCR trong đời sống, xuất hiện dưới dạng máy scan trong các văn phòng tuy nhiên hiện nay máy scan chỉ có thể quét được hình ảnh và trả chúng về định dạng hình ảnh số.

Hỗ trợ cuộc sống cho người già và khiếm thị: Phần mềm OCR có khả năng nhận dạng hệ thống phông chữ dành cho người khiếm thị từ đó tích hợp với công nghệ tổng hợp giọng nói để đọc ra thành tiếng văn bản được giải mã. Ứng dụng này còn được biết đến như một dạng giọng nói vi tính hóa, giúp cho việc đọc của người cao tuổi và người khiếm thị trở nên đơn giản hơn.

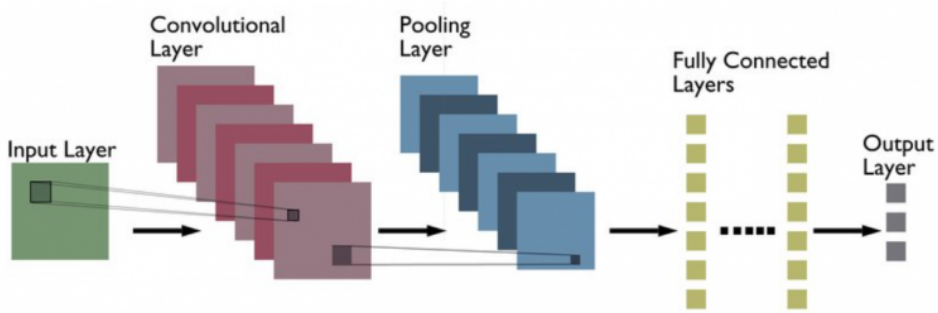
Xử lý hóa đơn, chứng từ: Đối với các cơ sở kinh doanh và một số văn phòng như toà án, văn phòng luật,v.v, số lượng hóa đơn và chứng từ cần được xử lý rất nhiều và phức tạp. Chính vì thế có rất nhiều tổ chức và doanh nghiệp đã ứng dụng OCR để nhập dữ liệu và chuyển đổi các thông tin hợp đồng, chứng từ, hóa đơn thành dạng văn bản kỹ thuật số để thuận tiện hơn cho việc tích hợp, chia sẻ và liên kết dữ liệu. Ngoài ra công nghệ OCR có thể tự động bóc tách dữ liệu theo các trường thông tin cơ bản như từ khoá, ngày, tháng, loại chứng từ, loại hóa đơn, v.v từ đó giúp việc tìm kiếm tài liệu trở nên dễ dàng và nhanh chóng.

Số hóa quy trình quản lý và lưu trữ tài liệu: Ngoài việc lưu trữ các tài liệu của công ty như hóa đơn, hợp đồng, chứng từ, văn bản, v.v. OCR còn góp phần không nhỏ trong việc giúp lưu trữ các văn bản có giá trị lịch sử. Đối với các loại bản thảo, hồi ký, bút ký hay tài liệu lịch sử có giá trị cao cần có những phương pháp cất giữ đặc biệt, chính vì thế việc nhập liệu thủ công tất cả các loại tài liệu trên gần như là không thể. Công nghệ OCR ra đời giúp cho việc lưu trữ và bảo tồn chúng trở nên dễ dàng hơn rất nhiều.

1.2.2. Mô hình Convolutional Neural Networks

Convolutional Neural Networks (CNN) là một loại mô hình học sâu được sử dụng rộng rãi trong xử lý hình ảnh và thị giác máy tính. Nó được thiết kế đặc biệt để hiệu quả trong việc xử lý thông tin không gian và bao gồm các lớp tích chập và lớp gộp để trích xuất các đặc trưng từ hình ảnh.

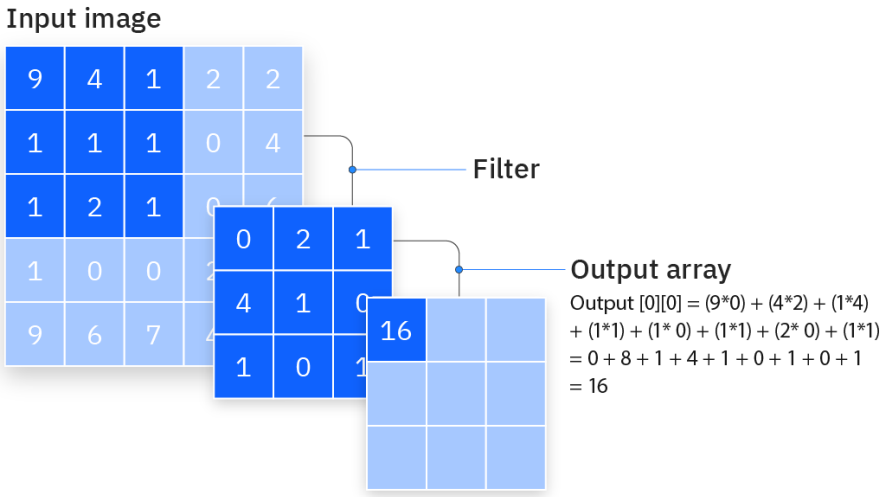
CNN bao gồm tập hợp các lớp cơ bản được thể hiện như hình 1.6 bao gồm lớp convolutional, lớp pooling, lớp fully connected. Một ảnh đầu vào được lan truyền qua lớp convolutional đầu tiên sau đó các giá trị tính toán được lan truyền qua lớp pooling. Hai lớp convolutional và lớp pooling được lặp lại nhiều lần phụ thuộc vào cấu trúc cụ thể của mạng và độ phức tạp của bài toán, sau đó được lan truyền qua lớp fully connected để phân loại.



Hình 1.6: Cấu trúc của mô hình CNN [1]

Lớp Convolutional: Là lớp quan trọng nhất và cũng là lớp đầu tiên của mô hình CNN. Chức năng chính của lớp này là phát hiện các đặc trưng không gian hiệu quả. Lớp convolutional yêu cầu các thành phần chính sau:

- Ma trận đầu vào: Là hình ảnh đầu vào được biểu diễn dưới dạng ma trận.
- Bộ filters (bộ lọc): Các bộ lọc này là những ma trận có kích thước nhỏ được áp dụng lên ma trận đầu vào để phát hiện các đặc trưng.
- Feature map: Là kết quả của việc áp dụng các bộ lọc lên ma trận đầu vào. Mỗi bộ lọc sẽ tạo ra một feature map riêng, biểu diễn các đặc trưng cụ thể được phát hiện trong hình ảnh.



Hình 1.7: Phép Convolution

Các bộ lọc di chuyển qua ma trận đầu vào thông qua phép tích chập để tạo ra các feature map được thể hiện ở hình 1.7. Các feature map này sau đó sẽ được đưa vào các lớp tiếp theo của mô hình CNN để trích xuất và xử lý thông tin từ hình ảnh.

Các tham số của lớp convolutional là:

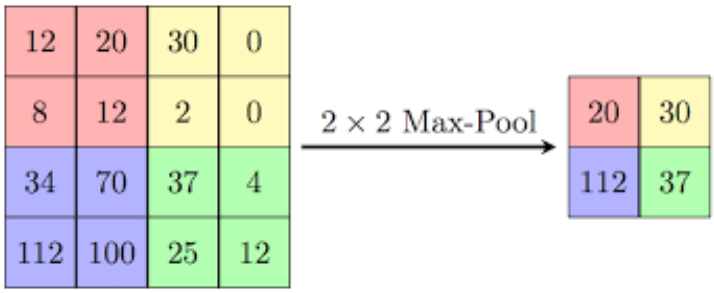
Kích thước bộ filter: Đây là một trong những tham số quan trọng nhất của lớp convolutional. Kích thước này quyết định độ lớn của vùng nhận thức (receptive field) của lớp convolutional, tức là vùng trong đó các điểm ảnh của hình ảnh đầu vào được xem xét để tạo ra một điểm ảnh trong feature map. Kích thước bộ filter thường được biểu thị bằng một số nguyên dương, chẳng hạn như 3x3 hoặc 5x5.

Stride: Stride (bước) là khoảng cách giữa các vị trí mà bộ filter di chuyển qua hình ảnh đầu vào. Nó quyết định tốc độ di chuyển của bộ filter và độ lớn của feature map đầu ra. Nếu Stride bằng 1, bộ filter sẽ di chuyển một điểm ảnh mỗi lần. Nếu Stride lớn hơn 1, bộ filter sẽ di chuyển nhanh hơn và feature map đầu ra sẽ có kích thước nhỏ hơn.

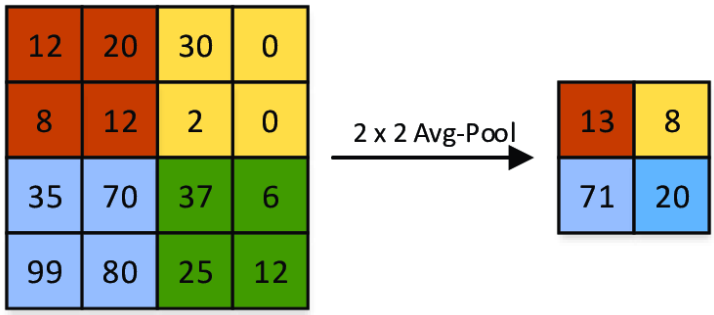
Padding: Padding (đệm) là việc thêm các giá trị 0 xung quanh biên của hình ảnh đầu vào trước khi áp dụng phép tích chập. Padding được sử dụng để duy trì kích thước của feature map đầu ra giống như kích thước của hình ảnh đầu vào. Có hai loại padding: valid (không padding) và same (padding sao cho kích thước của feature map đầu ra giống với kích thước của hình ảnh đầu vào).

Lớp Pooling là một phần không thể thiếu trong cấu trúc của một mô hình CNN. Sau khi áp dụng hàm kích hoạt sẽ sử dụng lớp Pooling để giảm kích thước của dữ liệu.

Có một số loại lớp Pooling phổ biến như max-pooling và average pooling. Chức năng chính của chúng là giảm kích thước của dữ liệu từ tầng trước, giúp giảm độ phức tạp tính toán.



Hình 1.8: Max Pooling [12]



Hình 1.9: Average Pooling [12]

Khi sử dụng một lớp Pooling có kích thước ví dụ 2x2 sau đó di chuyển một khối dữ liệu 2x2 trên các vùng ảnh cùng kích thước trong dữ liệu đầu vào. Sau đó tính giá trị lớn nhất như hình 1.8 hoặc trung bình như hình 1.9 của các giá trị trong khối dữ liệu đó. Lớp này giúp giảm kích thước của dữ liệu mà không làm mất đi thông tin quan trọng.

Lớp Pooling không chỉ giúp giảm kích thước của dữ liệu mà còn giúp tăng tốc quá trình huấn luyện. Nhờ vào lớp Pooling, mô hình CNN trở nên hiệu quả hơn trong việc nhận diện và phân loại hình ảnh.

Lớp fully Connected [5]: Lớp cuối cùng của mô hình CNN trong bài toán phân loại ảnh là lớp fully connected. Lớp này có chức năng chuyển ma trận đặc trưng ở lớp trước thành vector chứa xác suất của các đối tượng cần được dự đoán

1.2.3. Giải thuật Canny

Có hai phương pháp phát hiện biên là:

- Phát hiện trực tiếp: là phương pháp phát hiện biên dựa vào sự thay đổi của mức xám sử dụng các kỹ thuật thay đổi theo hướng.

- Phát hiện gián tiếp: Kỹ thuật phát hiện biên bằng phương pháp Canny là phương pháp được sử dụng phổ biến nhất, có hiệu quả cao và phức tạp so với nhiều phương pháp khác. Nó là một thuật toán nhiều giai đoạn được sử dụng để phát hiện, xác định một loạt các biên.

Thuật toán Canny là một thuật toán phát hiện cạnh trong xử lý ảnh. Nó được phát triển bởi John F. Canny vào năm 1986 và được xem là một trong những phương pháp phát hiện cạnh hiệu quả nhất và được sử dụng rộng rãi trong lĩnh vực thị giác máy tính và xử lý ảnh.

Canny đưa ra ba điểm chính mà một phương pháp phát hiện biên phải xác định được đó là:

- Mức lỗi: Phương pháp phải làm sao chỉ có hiệu quả đối với các điểm biên, phải tìm ra tất cả các biên và không có đường biên nào bị bỏ sót.
- Định vị: Khoảng cách giữa các điểm biên được tìm thấy trong giải thuật và biên trong thực tế phải càng nhỏ càng tốt.
- Hiệu suất: Không được phép chỉ ra nhiều biên trong khi chỉ có một biên tồn tại.

Giải thuật phát hiện biên Canny được trình bày như sau:

Loại bớt nhiễu trong ảnh

Vì tất cả các kết quả phát hiện cạnh đều dễ bị ảnh hưởng bởi nhiễu trong hình ảnh, nên việc lọc bỏ nhiễu là điều cần thiết để ngăn chặn phát hiện sai do nó gây ra. Để làm mịn hình ảnh, một hạt nhân bộ lọc Gaussian được đối chiếu với hình ảnh, chẳng hạn bộ lọc gauss 5x5 với hệ số δ = 1.4:

$$\mathbf{M} = \frac{1}{5}\begin{vmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{vmatrix}$$

Tìm gradient cường độ của hình ảnh

Một cạnh trong hình ảnh có thể hướng theo nhiều hướng khác nhau, do đó, thuật toán Canny sử dụng bốn bộ lọc để phát hiện các cạnh ngang, dọc và chéo trong hình ảnh bị mờ. Các phương pháp phát hiện cạnh (chẳng hạn như Roberts, Prewitt, hoặc Sobel) tính toán giá trị cho các đạo hàm bậc nhất theo hướng ngang (G_x) và hướng thẳng đứng (G_y) [15]:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{1.1}$$

Sau đó tính độ lớn gradient trong ảnh:

$$G = \sqrt{G_x^2 + G_y^2} \tag{1.2}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{1.3}$$

Trong đó, G_x và G_y chính là đạo hàm theo hướng x, y đang xét. Góc θ sẽ được làm tròn theo các hướng thẳng đứng, nằm ngang và theo hướng chéo (0°, 45°, 90° và 135°).

Loại bỏ các giá trị không phải cực đại

Sự triệt tiêu tối thiểu của cường độ gradient, hoặc ngưỡng giới hạn dưới, là một kỹ thuật làm mỏng cạnh. Áp dụng phương pháp triệt tiêu giới hạn dưới để tìm các vị trí có giá trị cường độ thay đổi mạnh nhất. Thuật toán cho mỗi điểm ảnh trong hình ảnh gradient là:

- So sánh cường độ cạnh của điểm ảnh hiện tại với cường độ cạnh của điểm ảnh theo hướng gradient âm và dương.
- Nếu cường độ cạnh của điểm ảnh hiện tại là lớn nhất so với các điểm ảnh khác trong mặt nạ có cùng hướng (ví dụ: điểm ảnh đang trỏ theo hướng y sẽ được so sánh với điểm ảnh ở trên và dưới điểm ảnh theo trục tung), giá trị sẽ được giữ nguyên. Nếu không, giá trị sẽ bị chặn.

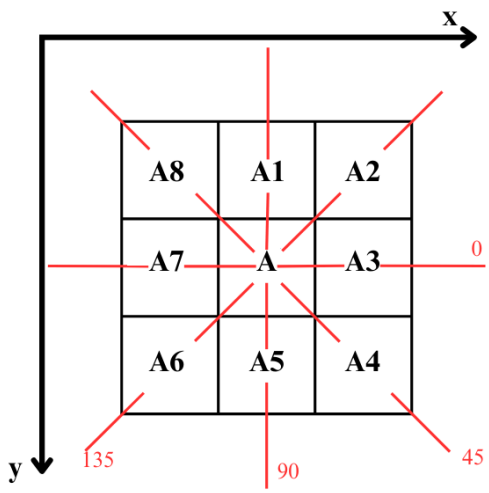
Trong một số triển khai, thuật toán phân loại các hướng gradient liên tục thành một tập hợp nhỏ các hướng rời rạc, và sau đó di chuyển bộ lọc 3x3 qua đầu ra của bước trước (nghĩa là cường độ cạnh và hướng gradient). Tại mỗi điểm ảnh, nó triệt tiêu cường độ cạnh của điểm ảnh trung tâm (bằng cách đặt giá trị của nó thành 0) nếu độ lớn của nó không lớn hơn độ lớn của hai điểm lân cận theo hướng gradient.

Nếu $\theta = 0$, khi đó điểm A sẽ được xem xét là một điểm trên biên độ nếu độ lớn gradient tại A lớn hơn gradient của các điểm tại A3, A7 được thể hiện như hình 1.10.

Nếu $\theta = 45$, khi đó điểm A được xem là một điểm trên biên độ nếu độ lớn gradient tại A lớn hơn độ lớn gradient của các điểm tại A4, A8 được thể hiện như hình 1.10.

Nếu $\theta = 90$, khi đó điểm A sẽ được coi là một điểm nằm trên biên độ nếu độ lớn gradient tại A lớn hơn độ lớn gradient của các điểm tại A1, A5 được thể hiện như hình 1.10.

Nếu $\theta = 135$, khi đó điểm A được xem là một điểm nằm trên biên độ nếu độ lớn gradient tại A lớn hơn độ lớn gradient của các điểm tại A2, A6 được thể hiện như hình 1.10.



Hình 1.10: Góc của θ trong thuật toán Canny

Chọn ra biên của đối tượng trong ảnh

Các điểm được đánh dấu là cạnh mạnh được lựa chọn ngay là điểm cạnh. Những điểm cạnh yếu được chọn là điểm cạnh khi và chỉ khi nó nằm cạnh một điểm cạnh mạnh. Ở đây các điểm cạnh mạnh được coi là cạnh thực, những điểm cạnh yếu thì có thể là cạnh thực hoặc nhiễu. Và chúng chỉ có thể là cạnh thực nếu liền kề với một điểm là cạnh thực.

1.3. Mô hình YOLOv8

YOLOv8 là một thuật toán thị giác máy tính được sử dụng để phát hiện đối tượng. Mô hình này được phát triển dựa trên các phiên bản trước của YOLO (You Only Look Once – Nhận diện vật thể bằng một lần nhìn).Ngoài ra, YOLOv8 còn được thiết kế để phát hiện đối tượng trong thời gian thực và có thể xử lý hình ảnh trong vài mili giây.

1.3.1. Khái quát về hệ thống phát hiện đối tượng

Phát hiện đối tượng (Object Detection)

Phát hiện đối tượng [4] là một công nghệ máy tính liên quan đến thị giác máy tính (Computer Vision) và xử lý hình ảnh (Image Processing) nhằm phát hiện các trường hợp của đối tượng ngữ nghĩa của một lớp nhất định trong hình ảnh và video kỹ thuật số.

Các mô hình phát hiện đối tượng thường được đào tạo để phát hiện sự hiện diện của các đối tượng cụ thể trong hình ảnh, video hoặc hoạt động thời gian thực (real-time).

Phát hiện đối tượng được coi là một trong những lĩnh vực quan trọng nhất trong phát triển của học sâu (Deep Learning) và xử lý hình ảnh.

Như Hình 1.11 trên, có tổng cộng 3 vật thể được mô hình phát hiện và dự đoán hộp giới hạn, sau đó sẽ được phân loại đối tượng bằng các dữ liệu thử nghiệm được cung cấp trước đó.

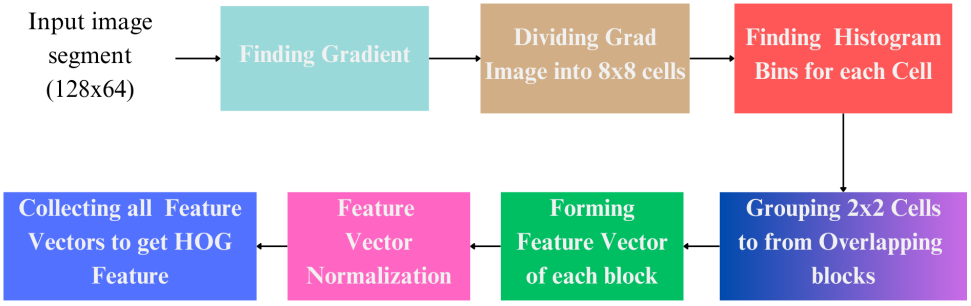
Hình 1.11: Mô hình YOLOv8 phát hiện và dự đoán hộp giới hạn

Độ chính xác của hộp giới hạn được dự đoán và hiệu suất tối ưu sẽ phụ thuộc vào dữ liệu thử nghiệm dùng để huấn luyện.

Các thuật toán phát hiện đối tượng phổ biến

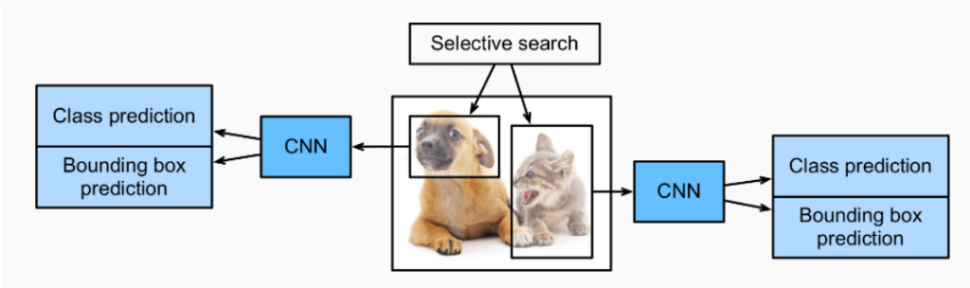
a. Phương pháp mô tả đặc trưng (Histogram of Oriented Gradients – HOG)

HOG [4] dựa trên việc chia ảnh đầu vào thành các ảnh con, tính toán histogram của ảnh để tổng hợp và trích rút ra các vector gọi là vector đặc trưng ứng với từng đối tượng được thể hiện như hình 1.12.

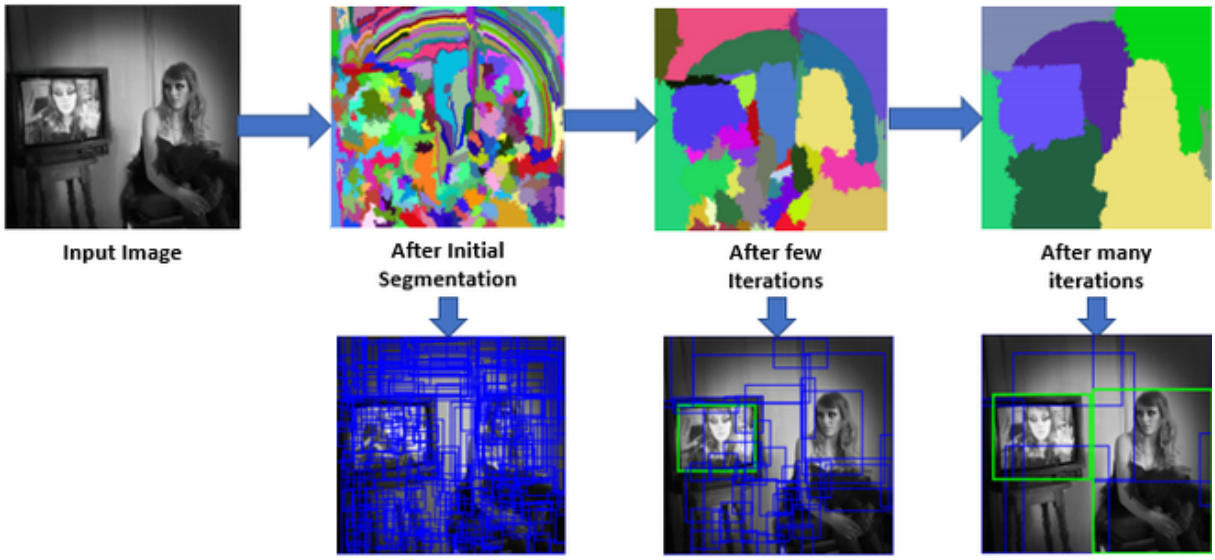


Hình 1.12: Kiến trúc hệ thống HOG để phát hiện đối tượng.

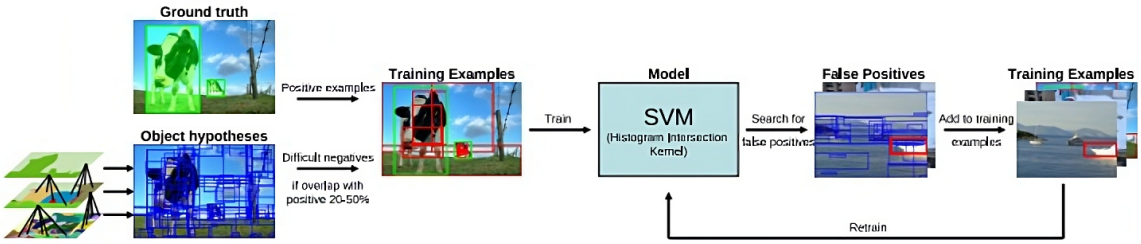
b. Mạng nơ-ron tích chập theo vùng (R-CNN) [4] Là một cải tiến mới trong kỹ thuật phát hiện đối tượng từ HOG và SIFT (Scale-invariant feature transform). Trong các mô hình R-CNN như hình 1.13 thường trích xuất các đặc trưng cần thiết nhất của đối tượng (2000 đặc trưng) bằng cách sử dụng giải thuật chọn lọc (selective search) như hình 1.14. Quá trình lựa chọn các đặc trưng quan trọng nhất có thể được tính toán với sự trợ giúp của thuật toán tìm kiếm chọn lọc.



Hình 1.13: Chu trình phát hiện đối tượng với Mạng nơ-ron tích chập dựa trên khu vực (R-CNN)



Hình 1.14: Giải thuật Selective Search

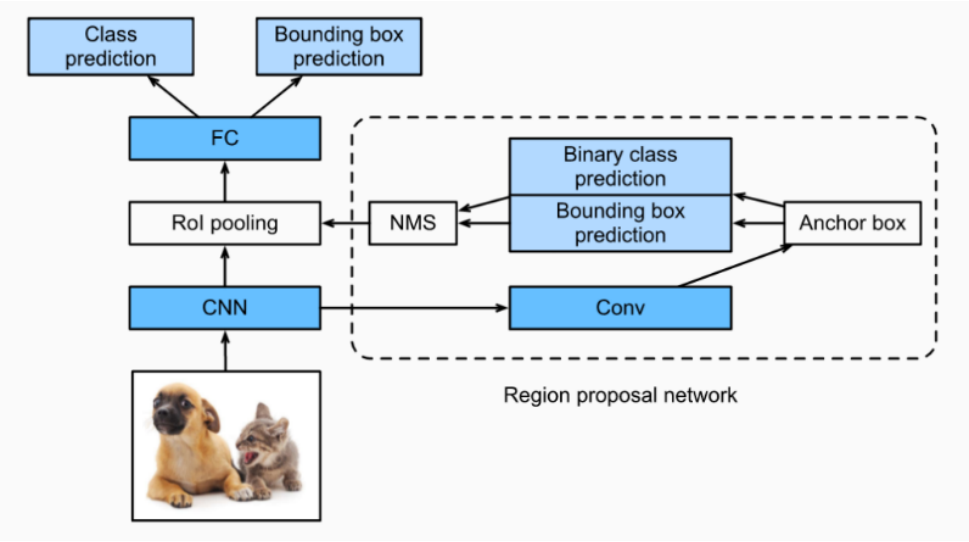


Hình 1.15: Nhận dạng đối tượng bằng Selective Search

c. Faster R-CNN

Mô hình Fast R-CNN và Faster R-CNN [4]đã được phát triển để giải quyết vấn đề tốc độ mô hình và xử lý các nhược điểm của R-CNN.

Trong khi mô hình R-CNN và Fast R-CNN sử dụng thuật toán tìm kiếm chọn lọc để tính toán các đề xuất vùng, thì phương pháp Faster R-CNN sẽ thay thế phương pháp hiện có này bằng một mạng đề xuất các vùng đặc trưng (Region Proposal Network – RPN) Được thể hiện như hình 1.16.



Hình 1.16: Minh họa mô hình Faster R-CNN

1.3.2. Giới thiệu YOLOv8

YOLOv8 là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLOv8 được tạo ra từ việc kết hợp giữa các lớp convolutional (Conv) và lớp fully connected. Trong đó các lớp convolutional sẽ trích xuất ra các feature của ảnh, còn lớp fully-connected (FC) sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh.

1.3.3. YOLOv8 và sự phát triển từ phiên bản trước

YOLOv8 là một mô hình nhận dạng đối tượng dựa trên CNN được phát triển bởi Joseph Redmon và nhóm nghiên cứu của ông tại Đại học Washington.

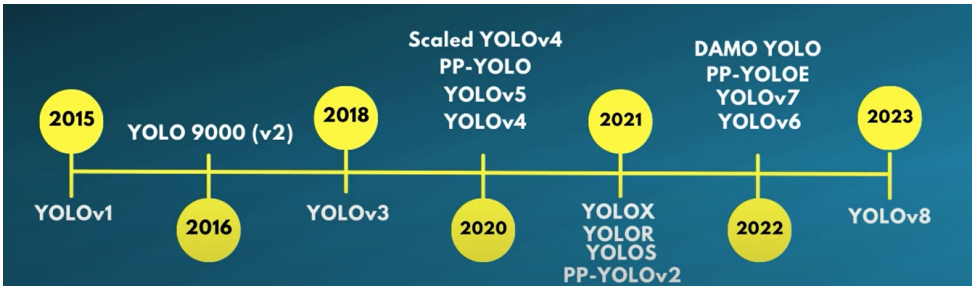
YOLOv8 là phiên bản nâng cấp của YOLOv7, với khả năng nhận diện đối tượng nhanh hơn và chính xác hơn. Điều này được đạt được thông qua một số cải tiến, bao gồm mạng kim tự tháp đặc trưng, các mô đun chú ý không gian và các kỹ thuật tăng cường dữ liệu tiên tiến.

Mô hình YOLOv8 sử dụng một mạng neural kiến trúc darknet-53 để trích xuất đặc trưng của hình ảnh và áp dụng thuật toán nhận dạng đối tượng YOLOv8 trên các đặc trưng đó.

Các đặc điểm chính

- Tăng cường mô hình bằng cách thêm các kênh phân tán để tăng tốc độ tính toán.
- Sử dụng kỹ thuật Attention để cải thiện khả năng nhận dạng đối tượng của mô hình.
- Áp dụng phương pháp đào tạo mới để tăng tốc độ hội tụ.
- Sử dụng kiến trúc mạng nơ-ron mới: Sử dụng kiến trúc YOLOv4 làm cơ sở để tăng hiệu suất và độ chính xác của mô hình.
- Tích hợp cơ chế tự động điều chỉnh tỷ lệ tăng kích thước của hình ảnh đầu vào (AutoScale).
- Hỗ trợ giám sát bằng video (Video Supervision): Mô hình có khả năng phát hiện và giám sát vật thể trong các video và đưa ra dự đoán liên tục trên toàn bộ video.
- Tích hợp công nghệ Ensemble.
- Tính năng điều chỉnh tỷ lệ tự động (AutoAnchor): Cải thiện việc phát hiện đối tượng với nhiều tỷ lệ khác nhau.

Tốc độ phát triển của Yolo được thể hiện như hình 1.17.



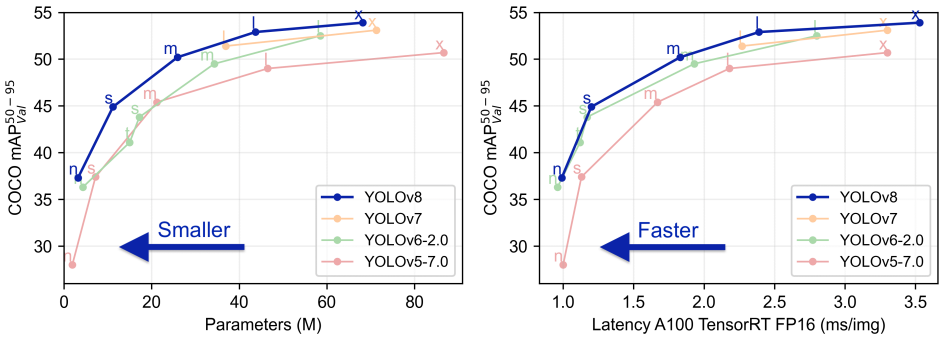
Hình 1.17: Dòng thời gian phát triển của YOLO

1.3.4. Ưu điểm và hạn chế của YOLOv8

Ưu điểm

- **Tốc độ:** YOLOv8 được đánh giá là nhanh chóng và thời gian phản hồi thấp, giúp xử lý các tác vụ nhận diện đối tượng và phân luồng ảnh trong thời gian thực.
- **Độ chính xác:** YOLOv8 được xây dựng trên các tiến bộ về học sâu và thị giác máy tính, đảm bảo độ chính xác cao trong việc nhận diện đối tượng. Hình 1.18 so sánh độ chính xác và hiệu suất của mô hình Yolo.

- **Sự linh hoạt:** YOLOv8 hỗ trợ việc nhận diện đối tượng và phân luồng trên cả GPU và CPU, tận dụng các công nghệ như TensorRT của Nvidia và OpenVino của Intel.



Hình 1.18: So sánh độ chính xác và hiệu suất của các mô hình YOLO

Hạn chế

Để sử dụng YOLOv8 hiệu quả cần phải [4]:

- Có kiến thức chuyên sâu về Machine Learning, Deep Learning và các thuật toán liên quan.
- Cần phải được huấn luyện trên một tập dữ liệu đủ lớn và đa dạng để đạt được hiệu quả cao nhất.
- Yêu cầu các tài nguyên tính toán cao để đạt được tốc độ xử lý nhanh và chính xác.
- Thuật toán YOLOv8 không phải là mã nguồn mở và chỉ có sẵn thông qua các thỏa thuận cấp phép với người tạo ra nó, Joseph Redmon.
- YOLOv8 có thể không hoạt động tốt trong tất cả các môi trường và có thể cần thêm điều chỉnh hoặc tối ưu hóa để đạt được hiệu suất tối ưu.

1.3.5. Các bước huấn luyện và ứng dụng của YOLOv8

1.3.5.1. Chuẩn bị dữ liệu

Để huấn luyện mô hình YOLOv8, cần có tập dữ liệu hình ảnh được gắn nhãn và chú thích đối tượng tương ứng của chúng. Các chú thích bao gồm nhãn lớp đối tượng (class) và tọa độ của hộp giới hạn (bounding box) của nó trong ảnh. Dữ liệu này được sử dụng để đào tạo mô hình phát hiện chính xác các đối tượng trong hình ảnh mới.

Việc tạo ra tập dữ liệu tùy chỉnh có thể là một quá trình khó khăn. Có thể mất hàng chục hoặc thậm chí hàng trăm giờ để thu thập hình ảnh, gắn nhãn và chuyển đổi chúng ở định dạng thích hợp. Nhưng ngày nay đã có nhiều công cụ hỗ trợ việc xử lý dữ liệu như LabelImg, RectLabel hay Roboflow (Hình 2.12).

1.3.5.2. Tạo tập dữ liệu

- Thu thập dữ liệu: Bước đầu tiên trong việc tạo tập dữ liệu là xác định vấn đề và loại dữ liệu cần thiết. Sau khi đã xác định được vấn đề và loại dữ liệu, bước tiếp theo là thu thập dữ liệu thực tế. Các hoạt động này có thể bao gồm chụp ảnh, tải xuống hình ảnh từ internet hoặc thu thập dữ liệu từ các nguồn khác.
- Gán nhãn cho dữ liệu: Dữ liệu sau khi được thu thập cần được gán nhãn. Gán nhãn bao gồm việc thêm chú thích về vị trí và loại của các đối tượng trong ảnh. Một số công cụ hỗ trợ gán nhãn ảnh như: LabelImg, RectLabel hay Roboflow.
- Chuyển đổi dữ liệu:Tùy thuộc vào thư viện CNN, dữ liệu sau khi gán nhãn cần được chuyển sang một định dạng phù hợp với thư viện đó. Ngoài ra ảnh cũng có thể được xử lý thông qua một số thao tác như thay đổi kích thước, chuẩn hóa giá trị hay áp dụng kỹ thuật tăng cường để tăng kích thước tập dữ liệu.

1.3.5.3. Quá trình huấn luyện

Quá trình huấn luyện sử dụng một số tham số trực tiếp quyết định đến hiệu suất và độ chính xác của mô hình như: batch size, learning rate và epochs:

- Batch size: là một trong những tham số xác định số lượng dữ liệu huấn luyện được sử dụng trong một lần lặp lại thuật toán. Nó chỉ định số lượng mẫu sẽ được truyền qua mạng thần kinh (Neural Network) cùng một lúc. Việc lựa chọn batch size phụ thuộc vào kích thước của tập dữ liệu, độ phức tạp của mô hình và tài nguyên máy tính.
- Learning rate: là một tham số xác định kích thước weight (giá trị) mà tại đó các tham số của mô hình được cập nhật trong quá trình đào tạo. Learning rate lớn có nghĩa là các tham số của mô hình được cập nhật mạnh mẽ hơn, điều này có thể dẫn đến việc học nhanh hơn nhưng cũng có thể dẫn đến vượt quá các giá trị tối ưu. Learning rate nhỏ có nghĩa là các tham số của mô hình được cập nhật thận trọng hơn, điều này có thể làm chậm quá trình học nhưng có thể dẫn đến độ ổn định và độ chính xác được cải thiện.
- Epochs: là một lần lặp hoàn chỉnh thông qua toàn bộ tập dữ liệu trong quá trình đào tạo. Trong mỗi Epochs, mô hình được trình bày với toàn bộ tập dữ liệu và nó cập nhật các trọng số của nó dựa trên các lỗi mà nó mắc phải trong khi dự đoán đầu ra chính xác. Mục tiêu của nhiều Epochs là cải thiện độ chính xác của mô hình bằng cách điều chỉnh weight của nó trên toàn bộ tập dữ liệu nhiều lần.

1.3.5.4. Đánh Giá Mô Hình

Sau khi thiết lập tham số thì việc đào tạo hoàn toàn được máy tính xử lý, tuy nhiên ta cần đánh giá mô hình sau khi hoàn thành là một bước quan trọng để kiểm tra hiệu suất của mô hình với những tệp dữ liệu mới. Một số chỉ số được sử dụng để đánh giá hiệu suất của mô hình như:

- Accuracy (độ chính xác): là số liệu đơn giản và phổ biến nhất đo lường số dự đoán chính xác do mô hình đưa ra chia cho tổng số dự đoán. Tuy nhiên, độ chính xác thôi có thể không đủ để đánh giá các mô hình cho dữ liệu mất ổn định.
- Precision (độ chuẩn xác): là phần nhỏ của các dự đoán tích cực thực sự trong số tất cả các dự đoán tích cực do mô hình đưa ra. Số liệu này rất quan trọng trong các tình huống mà kết quả dương tính giả trở thành vấn đề.
- Recall (khả năng thu hồi): là phần dự đoán tích cực thực sự trong số tất cả các điểm dữ liệu tích cực. Số liệu này rất quan trọng trong các tình huống mà âm tính giả trở thành vấn đề.
- F1 Score: là một chỉ số giữa độ chính xác và khả năng thu hồi. Nó cung cấp sự cân bằng giữa hai số liệu và hữu ích khi cả độ chính xác và khả năng thu hồi đều quan trọng.

1.4. Tìm hiểu về các thiết bị nhúng sử dụng trong hệ thống

1.4.1. Giới thiệu chung về máy tính nhúng Raspberry

Raspberry Pi như hình 1.19 [11] là chiếc máy tính kích thước nhỏ được tích hợp nhiều phần cứng đủ khả năng chạy hệ điều hành và cài đặt được nhiều ứng dụng trên nó.Raspberry Pi phù hợp cho những ứng dụng cần khả năng xử lý mạnh mẽ, đa nhiệm hoặc giải trí và đặc biệt cần chi phí thấp.

Hình 1.19: Máy tính nhúng Raspberry Pi

Các model của Raspberry hiện nay: Raspberry Pi Zero,Raspberry Pi 1, Raspberry Pi 2, Raspberry Pi 3, Raspberry Pi 4.

1.4.1.1. Hệ điều hành cho Raspberry

Một số hệ điều hành được sử dụng cho Raspberry Pi:

- Raspbian: Đây là hệ điều hành phổ biến và được ưa chuộng nhất.
  - Sử dụng Raspberry Pi như máy tính thu nhỏ với chức năng văn phòng
  - Nghiên cứu và xây dựng hệ thống điều khiển tự động.
  - Cung cấp các dịch vụ như web, File server, printer server, office, game…
  - Sử dụng Raspberry Pi như máy tính thu nhỏ với chức năng văn phòng
  - Nghiên cứu và xây dựng hệ thống điều khiển tự động.
  - Cung cấp các dịch vụ như web, File server, printer server, office, game…
- Ubuntu Mate: hệ điều hành phù hợp cho nhân viên văn phòng với những tiện ích hỗ trợ. Và hệ điều hành này so với hệ điều hành Raspbian có giao diện đẹp hơn và được phát triển cùng với Ubuntu.
- Snappy Core Ubuntu: là một hệ điều hành thu nhỏ của Ubuntu. Và được sử dụng và chạy các ứng dụng đám mây và là một phần quan trọng trong IoT, giúp cho các thiết bị kết nối với nhau thông qua internet một cách hoàn hảo. Nó có thể chạy được nhiều ứng dụng mà không cần một hệ điều hành Ubuntu đầy đủ.
- Windows 10 IoT Core: Được sử dụng với mục đích phát triển các ứng dụng IoT. Nó được thiết kế để có thể hoạt động với một loạt các ngôn ngữ mã nguồn mở, giúp phát triển và dễ dàng cài đặt trên các thiết bị của riêng mình.

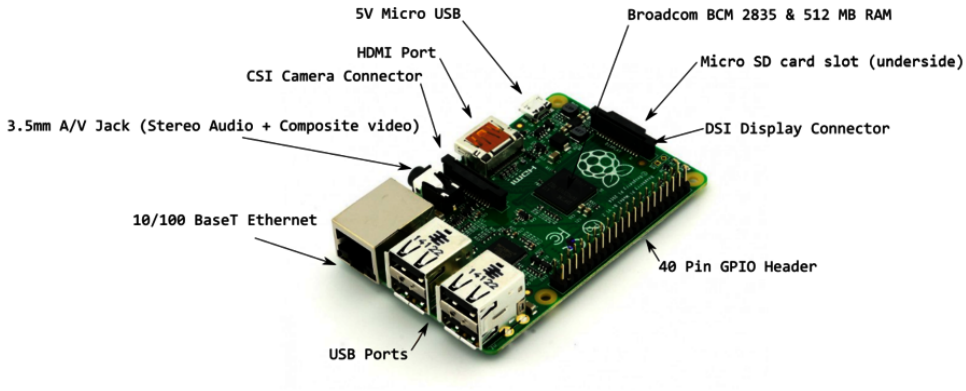
1.4.1.2. Raspberry Pi3 Model B+

Raspberry Pi3 như hình 1.20 nổi bật với chip 4 nhân 64-bit có tốc độ 1.4GHz – nhanh nhất từ trước đến nay! Phiên bản mới còn hỗ trợ Wifi Dual-band 2.4GHz và 5GHz, Bluetooth 4.2/Bluetooth Low Energy, cổng Ethernet tốc độ cao (300Mbps) và Power over Ethernet (PoE) thông qua PoE HAT.

Thông số kỹ thuật chi tiết:

- Vi xử lý: Broadcom BCM2837B0, quad-core A53 (ARMv8) 64-bit SoC @1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Kết nối: 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (Tối đa 300Mbps).
- Cổng USB: 4 x 2.0
- Mở rộng: 40-pin GPIO
- Video và âm thanh: 1 cổng full-sized HDMI, Cổng MIPI DSI Display, cổng MIPI CSI Camera, cổng stereo output và composite video 4 chân.

- Multimedia: H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
- Lưu trữ: MicroSD
- Nguồn điện sử dụng: 5V/2.5A DC cổng micro USB, 5V DC trên chân GPIO, Power over Ethernet (PoE) (yêu cầu thêm PoE HAT).



Hình 1.20: Raspberry Pi 3 Model B+ [3]

1.4.2. Camera USB

USB Camera Cmos 800x600 Cho Máy Tính Nhúng Raspberry Pi được sử dụng gắn với cổng USB của Raspberry Pi để truyền hình ảnh về máy. Thông số kỹ thuật Camera usb hỗ trợ tất cả các phiên bản của Raspberry Pi, máy tính nhúng.

Hình 1.21: Camera USB

Thông số kỹ thuật:

- Camera usb hỗ trợ tất cả các phiên bản của Raspberry Pi, máy tính nhúng.
- Ống kính tiêu cự: F6.0MM
- Focus khoảng: 20MM
- Độ phân giải Video: 640 x 480

- Kích thước: 3.8 x 1.5 x 3cm
- Dây có thể được kéo dài hoặc rút ngắn, dài 65 cm

1.4.3. Động Cơ Servo SG90S

Động cơ servo SG90s có kích thước nhỏ, giá thành thấp là loại động cơ servo phổ biến sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng, phù hợp với mọi người, đặc biệt là học sinh, sinh viên với nhu cầu nghiên cứu, ứng dụng, học tập,v.v Động cơ servo SG90 có các bánh răng được làm bằng nhựa, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

Hình 1.22: Động Cơ Servo SG90S [13]

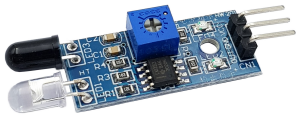
Thông số kỹ thuật SG90S:

- Chất liệu bánh răng: Kim loại
- Kích thước 21.5 x 11.8 x 22.7mm
- Trọng lượng 9g
- Tốc độ không tải 0.12 giây / 60º (4.8V)
- Mô-men xoắn 1.2-2.4 kg.cm (4.8V)
- Nhiệt độ hoạt động -30 → +60 º C
- Điện áp hoạt động 4.8 → 6 V DC

1.4.4. Cảm biến vật cản hồng ngoại

Led phát hồng ngoại (IR LED) luôn luôn phát ra sóng ánh sáng có bước sóng hồng ngoại mắt người không thể nhìn thấy ánh sáng này, vì vậy người ta sử dụng led thu hồng ngoại, led thu hồng ngoại bình thường nó có nội trở rất lớn (vài trăm KΩ), khi led thu được tia hồng ngoại chiếu vào đủ lớn thì nội trở của nó giảm xuống (cỡ vài chục Ω).

Khi gặp vật cản, những chùm tia hồng ngoại gặp vật cản và phản xạ lại led thu làm led thu thay đổi giá trị điện trở. Ở đây ta thấy cầu chia áp ở điện trở R2 và mắt thu hồng ngoại, sự thay đổi điện trở của mắt thu hồng ngoại dẫn đến điện áp đầu vào chân 3 Op-Amp cũng thay đổi.



Hình 1.23: Cảm biến hồng ngoại [9]

Khi khoảng cách càng gần, sự thay đổi càng lớn.

Khi đó, điện áp đầu vào chân 3 Op-Amp được so sánh với giá trị điện áp không đổi trên biến trở R3, nếu điện áp chân 3 Op-Amp lớn hơn điện áp chân 2 Op-Amp thì Op-Amp xuất mức 1 (bằng VCC) . Ngược lại nếu điện áp chân 3 Op-Amp nhỏ hơn điện áp chân 2 Op-Amp thì Op-Amp xuất mức 0 (bằng GND)

Điện trở như R1 (150 Ω), R2 (10K Ω), R4 (1 KΩ) giúp các led hoạt động mà không bị cháy

Biến trở R3 dùng để chỉnh độ nhạy của biến trở

Thông số kỹ thuật.

- Điện áp cung cấp cho mạch: 3.3V-5V.
- Bộ so sánh dùng Op Amp LM358 hoặc LM393.
- Đầu ra kỹ thuật số mức (1 và 0).

1.5. Tìm hiểu về Flask Framework

1.5.1. Giới thiệu chung về Flask Framework

Flask [10] là một framework web nhỏ gọn và linh hoạt, được xây dựng cho ngôn ngữ lập trình Python. Được biết đến với sự đơn giản và sử dụng ít tài nguyên hệ thống, Flask cung cấp một cách tiếp cận dễ hiểu và tự do trong phát triển ứng dụng web. Khác với các framework lớn hơn, Flask không áp đặt cấu trúc hay thư viện cụ thể, cho phép lập trình viên tùy chỉnh ứng dụng của mình theo ý muốn.

Đặc điểm chính:

Flask là một micro-framework nhẹ, không chứa nhiều thành phần giúp giảm độ phức tạp và tăng tính linh hoạt.

Đơn giản và dễ hiểu:

Cú pháp của Flask rất đơn giản và dễ hiểu, làm cho việc học và triển khai ứng dụng trở nên thuận tiện.

Jinja2 Template Engine:

Flask sử dụng Jinja2, một engine template cho Python, giúp tạo ra các trang HTML động một cách dễ dàng.

Werkzeug Library:

Flask sử dụng thư viện Werkzeug để giải quyết các vấn đề liên quan đến HTTP, như định tuyến và xử lý yêu cầu.

Cơ sở dữ liệu:

Flask không buộc phải sử dụng một cơ sở dữ liệu cụ thể, giúp lập trình viên lựa chọn giữa nhiều loại cơ sở dữ liệu. Flask hỗ trợ nhiều cơ sở dữ liệu như SQLite, MySQL, PostgreSQL, giúp linh hoạt trong lựa chọn công nghệ lưu trữ dữ liệu.

Mở rộng dễ dàng:

Flask cung cấp các thành phần mở rộng để mở rộng chức năng, giúp tối ưu hóa và mở rộng ứng dụng theo nhu cầu cụ thể.

Ứng dụng RESTful:

Hỗ trợ phát triển các ứng dụng RESTful thông qua quá trình định tuyến đơn giản.

Cộng đồng lớn và hỗ trợ tốt:

Flask có cộng đồng lớn, cung cấp tài liệu phong phú và sự hỗ trợ đa dạng từ cộng đồng.

Giấy phép BSD:

Flask sử dụng giấy phép BSD, mang lại tính tự do và linh hoạt trong việc sử dụng cho mục đích thương mại.

1.5.2. Tính năng của Flask Framework

Flask cung cấp khả năng tùy biến khi phát triển ứng dụng web, nó cung cấp các công cụ, thư viện và cơ chế cho phép xây dựng một ứng dụng web nhưng nó sẽ không thực thi bất kỳ sự phụ thuộc nào.

- Khả năng tương thích công cụ dựa trên ứng dụng Google.
- Có nhiều tiện ích cho các chức năng khác nhau.
- Tính mô đun hóa và thiết kế gọn nhẹ.
- Có độ linh hoạt cao.
- API độc đáo và mạch lạc.
- Dễ dàng triển khai.

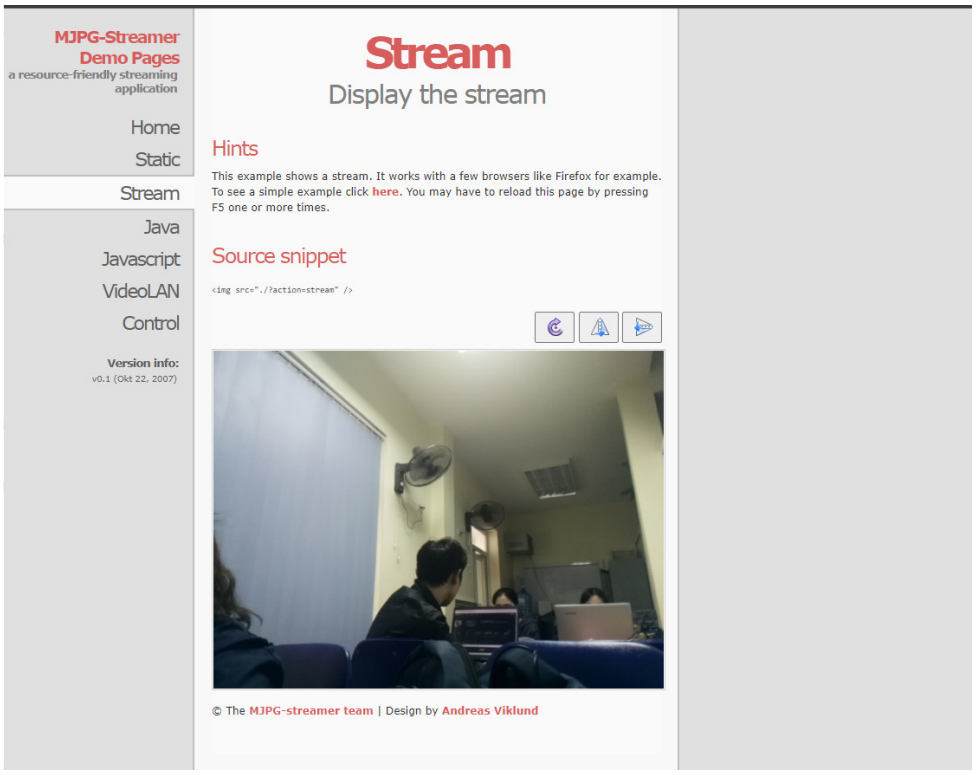
1.5.3. Mjpg-streamer

Mjpg-streamer là một ứng dụng dòng lệnh sao chép các khung hình JPEG từ một hoặc nhiều đầu vào sang nhiều đầu ra. Nó có thể được sử dụng để truyền các tệp JPEG qua mạng dựa trên IP từ máy ảnh đến nhiều loại trình duyệt khác nhau như Chrome, Firefox, Cambozola, VLC, mplayer và các phần mềm khác có khả năng nhận luồng MJPG. Hình 1.24 là hình ảnh được phát lên trình duyệt [6].

Các ưu điểm của MJPEG bao gồm độ đơn giản và dễ hiểu, cũng như khả năng xử lý các khung hình độc lập mà không cần phải giải nén toàn bộ video. Tuy nhiên, định dạng này có thể tốn nhiều băng thông hơn so với các định dạng video khác như H.264 hoặc H.265, vì mỗi khung hình đều được lưu trữ riêng lẻ dưới dạng hình ảnh nén.

MJPEG thường được sử dụng trong các ứng dụng giám sát an ninh và các hệ thống giám sát video khác. Nó cũng thường được sử dụng trong các ứng dụng nơi có yêu cầu độ chính xác cao với mỗi khung hình cụ thể.

Một "MJPEG stream" đơn giản là một chuỗi các hình ảnh JPEG được truyền liên tục qua mạng để tạo thành một luồng video MJPEG. Như vậy có thể được sử dụng để xem trực tiếp hoặc ghi lại video từ một nguồn hình ảnh, chẳng hạn như một camera IP hoặc webcam thông qua mạng. Để xem hoặc xử lý MJPEG stream, có thể sử dụng các phần mềm hoặc thư viện hỗ trợ định dạng này.



Hình 1.24: Video stream lên trên trình duyệt

Chương 2
THIẾT KẾ XÂY DỰNG HỆ THỐNG VÀO RA TẠI HỌC VIỆN

2.1. Phân tích chức năng

2.1.1. Chức năng của hệ thống

Hệ thống quản lý ra vào tại Học viện Kỹ thuật mật mã được xây dựng gồm các chức năng sau:

- Chức năng đăng nhập
- Chức năng thêm người dùng
- Chức năng xem lịch sử ra vào
- Chức năng sửa thông tin người dùng
- Chức năng xóa người dùng
- Chức năng thống kê
- Chức năng tải dữ liệu thống kê
- Chức năng nhận dạng biển số xe

2.1.1.1. **Chức năng đăng nhập**

Để đăng nhập vào trong hệ thống điều khiển trên ứng dụng dành cho người dùng sẽ thực hiện theo các bước như bảng 2.1

Bảng 2.1: Bảng phân tích chức năng đăng nhập

Tên chức năng	Đăng nhập cho admin của hệ thống
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Thuộc tài khoản admin
Mô tả	1. Người dùng đăng nhập tài khoản bằng user-name và password 2. Người dùng nhấn đăng nhập 3. Hệ thống kiểm tra thông tin trên cơ sở dữ liệu 4. Hệ thống thông báo trạng thái đăng nhập tới người dùng
Kết quả	Đăng nhập thành công vào hệ thống

2.1.1.2. **Chức năng thêm người dùng**

Để thêm người dùng vào trong hệ thống admin cần thực hiện như bảng 2.2

Bảng 2.2: Bảng phân tích chức năng thêm người dùng

Tên chức năng	Thêm người dùng
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Đăng nhập thành công vào hệ thống
Mô tả	1. Admin đăng nhập vào hệ thống 2. Nhập thông tin các trường họ và tên, biển số xe và bộ phận vào trang thêm người dùng 3. Hệ thống kiểm tra thông tin nhập vào 4. Hệ thống thông báo trạng thái 5. Admin nhấn nút thêm
Kết quả	Thêm thành công thông tin vào hệ thống

2.1.1.3. **Chứng năng xem lịch sử ra vào**

Hệ thống có thể xem giờ ra vào của các phương tiện ra vào. Quá trình thực hiện như bảng 2.3

Bảng 2.3: Bảng phân tích chức năng xem lịch sử ra vào

Tên chức năng	Quản lý ra vào
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Đăng nhập thành công vào hệ thống
Mô tả	1. Admin đăng nhập vào hệ thống 2. Nhấn vào trang quản lý xe 3. Hệ thống xuất trạng thái ra vào trên cơ sở dữ liệu
Kết quả	Người dùng xem thành công

2.1.1.4. **Chức năng sửa thông tin người dùng**

Chức năng sửa thông tin người dùng được thực hiện như bảng 2.4

Bảng 2.4: Bảng phân tích chức năng sửa thông tin người dùng

Tên chức năng	Sửa thông tin người dùng
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Đăng nhập thành công vào hệ thống
Mô tả	1. Nhấn vào hàng chứa thông tin cần sửa 2. Hệ thống truy cập database đưa thông tin người dùng ra bảng để sửa thông tin 3. Sửa các trường cần sửa thông tin 4. Hệ thống cập nhật lại database và hiển thị thông tin lên bảng
Kết quả	Sửa thông tin người dùng thành công

2.1.1.5. **Chức năng xóa người dùng**

Chức năng xóa người dùng được thực hiện như bảng 2.5

Bảng 2.5: Bảng phân tích chức năng xóa người dùng

Tên chức năng	Xóa người dùng
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Đăng nhập thành công vào hệ thống
Mô tả	1. Admin đăng nhập vào trang quản lý người dùng 2. Nhấn nút xóa 3. Hệ thống xóa và cập nhật lên cơ sở dữ liệu 4. Hiển thị lại danh sách người dùng
Kết quả	Xóa thành công người dùng

2.1.1.6. **Chức năng thống kê**

Chức năng thống kê theo ngày, tháng của xe ra vào được được thực hiện như bảng 2.6

Bảng 2.6: Bảng phân tích chức năng thống kê theo ngày, tháng

Tên chức năng	Thống kê theo ngày, tháng
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Đăng nhập thành công vào hệ thống
Mô tả	1. Admin đăng nhập vào trang thống kê 2. Chọn ngày, tháng muốn thống kê và dữ liệu được gửi lên hệ thống 3. Hệ thống nhận dữ liệu (ngày, tháng, năm) từ web và truy cập cơ sở dữ liệu trả về dữ liệu về trang web 4. Hiển thị lại danh sách đã thống kê theo ngày, tháng hoặc năm
Kết quả	Hiển thị thành công danh sách đã thống kê

2.1.1.7. **Chức năng tải dữ liệu thống kê**

Chức năng tải dữ liệu thống kê dưới dạng CSV về trên hệ thống được thể hiện dưới dạng bảng reftab:csv

Bảng 2.7: Bảng phân tích chức năng tải dữ liệu thống kê

Tên chức năng	Tải dữ liệu thống kê
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Đăng nhập thành công vào hệ thống
Mô tả	1. Admin đăng nhập vào trang thống kê, người dùng, lịch sử ra vào 2. Chọn nút tải xuống và bảng muốn tải về để xem dữ liệu 3. Web gửi yêu cầu lên server, Server gửi dữ liệu yêu cầu về trang web 4. Tải thành công và lưu vào hệ thống trên máy tính người dùng
Kết quả	Tải thành công về máy

2.1.1.8. **Chức năng nhận dạng biển số xe**

Chức năng nhận dạng biển số xe được thực hiện như bảng 2.8

Bảng 2.8: Bảng phân tích chức năng nhận diện biển số xe

Tên chức năng	Nhận diện biển số xe
Đối tượng sử dụng	Admin hệ thống
Điều kiện đầu vào	Đăng nhập thành công vào hệ thống
Mô tả	1. Camera pi chụp ảnh biển số xe 2. Ảnh được truyền từ raspberry pi về server để xử lý 3. Hệ thống xác định vị trí biển số xe trong ảnh 4. Hệ thống cắt ảnh chỉ bao gồm biển số xe 5. Đọc và đưa ra biển số xe đã vào
Kết quả	Nhận diện thành công biển số xe

2.1.2. Yêu cầu phi chức năng

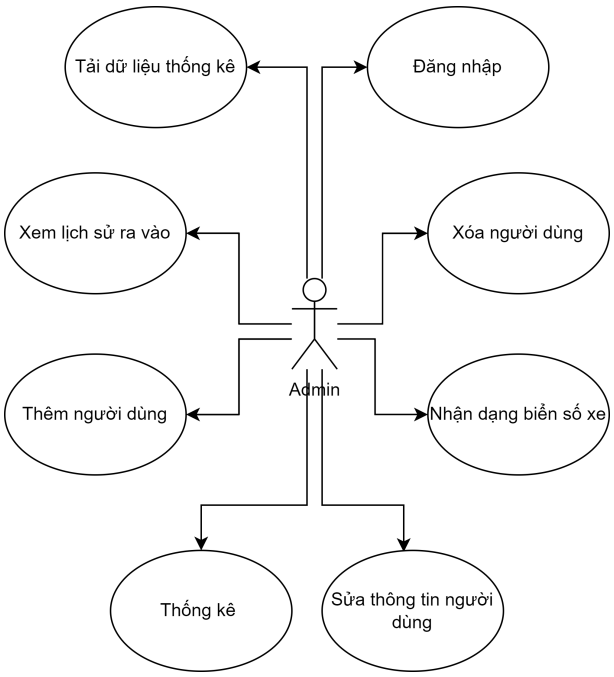
Hệ thống được xây dựng với yêu cầu phi chức năng như sau:

- Hệ thống có khả năng nhận diện chính xác biển số xe trong nhiều điều kiện ánh sáng khác nhau.
- Hệ thống hoạt động một cách nhanh chóng và đáp ứng trong thời gian thực, đặc biệt là trong các tình huống giao thông nhanh.
- Hệ thống đảm bảo ổn định và tin cậy trong mọi điều kiện môi trường và ánh sáng.
- Giao diện người dùng nên được thiết kế sao cho người quản trị và người dùng có thể sử dụng một cách thuận tiện và hiệu quả.

2.1.3. Biểu đồ

2.1.3.1. Biểu đồ Usecase

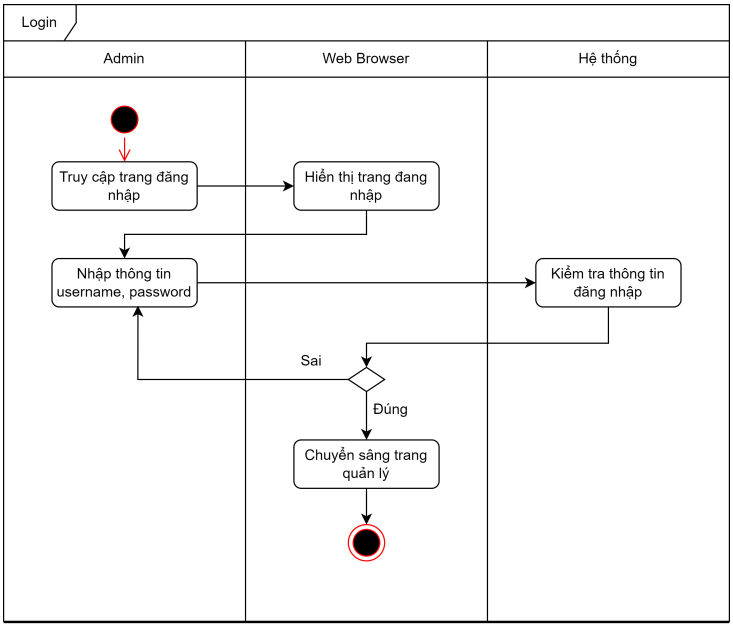
Hệ thống có các chức năng sau: đăng nhập, nhận dạng biển số xe, thêm người dùng, xóa người dùng, sửa thông tin người dùng, thống kê và xem lịch sử ra vào. Sau đây là biểu đồ usecase của hệ thống được thể hiện như hình 2.1.



Hình 2.1: Biểu đồ Usecase tổng quát

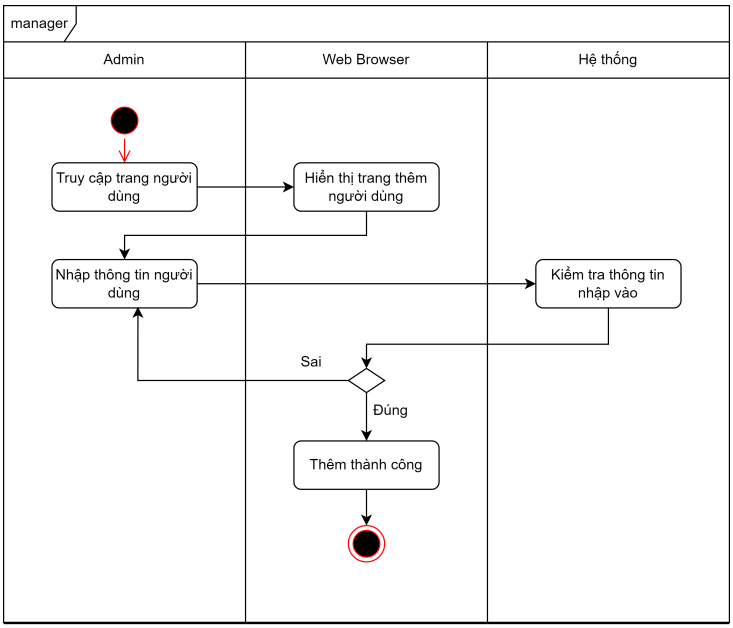
2.1.3.2. Biểu đồ hoạt động

Hình 2.2 là biểu đồ hoạt động của chức năng đăng nhập. Người dùng nhập username và password hệ thống sẽ kiểm tra tài khoản, nếu đúng sẽ di chuyển vào màn hình chính, nếu sai sẽ xuất ra thông báo và yêu cầu người dùng nhập lại.



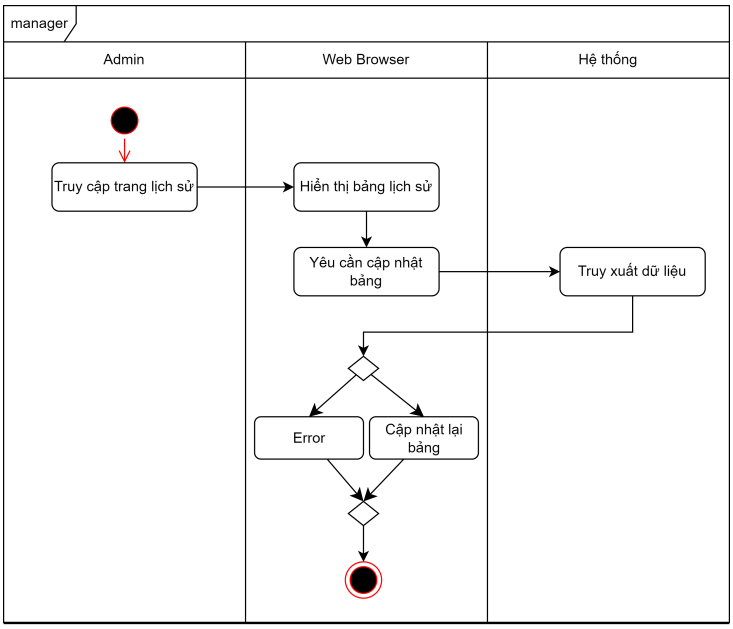
Hình 2.2: Biểu đồ hoạt động chức năng đăng nhập

Hình 2.3 là biểu đồ hoạt động thêm người dùng: Admin sẽ đang nhập vào trang thêm người dùng và nhập thông tin của người dùng sau đó nhấn thêm người dùng. Hệ thống sẽ thêm thành công người dùng vào cơ sở dữ liệu và hiển thị lên trên trang web.



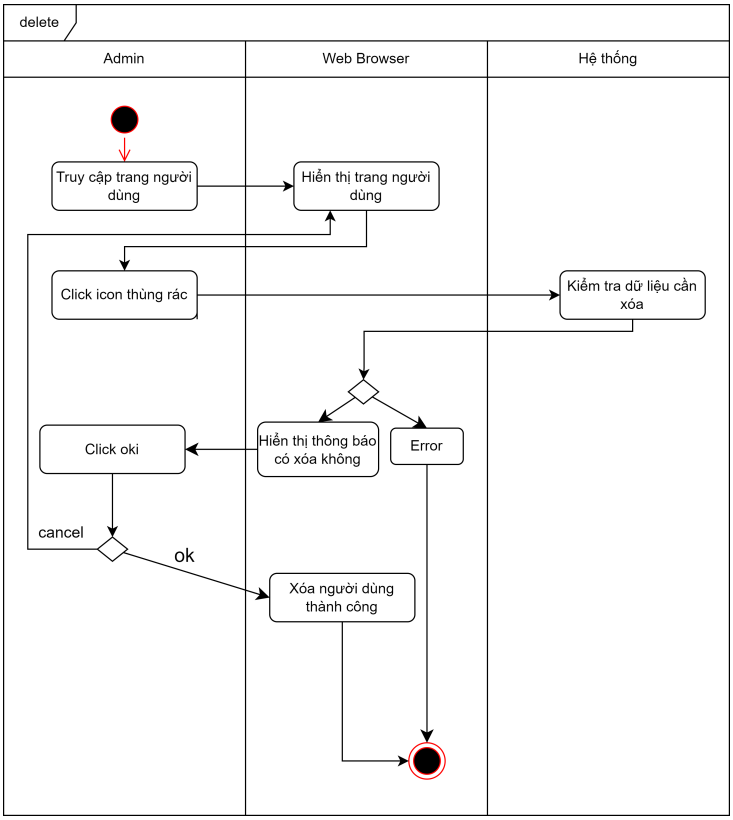
Hình 2.3: Biểu đồ hoạt động thêm người dùng

Hình 2.4 là biểu đồ hoạt động của chức năng xem lịch sử ra vào hệ thống và lấy ra danh sách thông tin ra vào từ cơ sở dữ liệu. Nếu lấy thành công, hệ thống sẽ hiển thị danh sách lên trên trang web, nếu không thành công hệ thống sẽ báo lỗi.



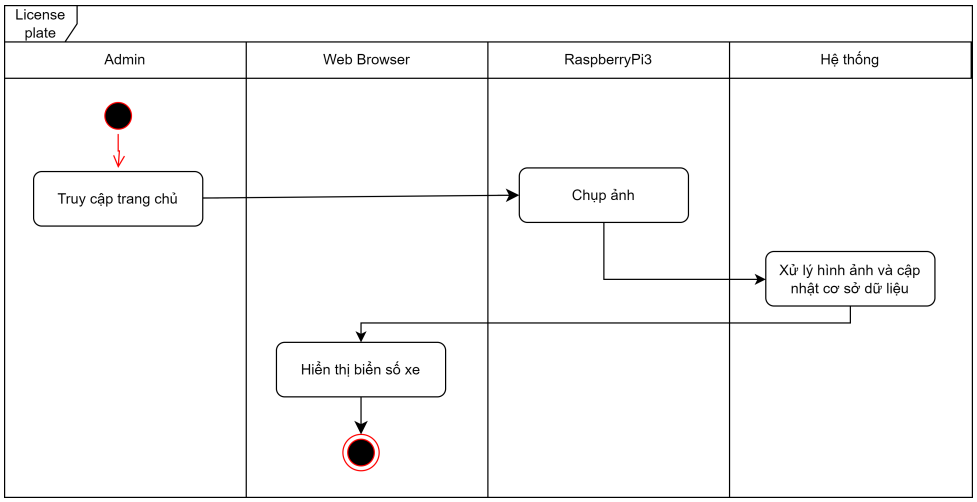
Hình 2.4: Biểu đồ hoạt động xem lịch sử ra vào

Hình 2.5 là biểu đồ hoạt động chức năng xóa người dùng. Admin đăng nhập vào màn quản lý người dùng và nhấn nút xóa. Nếu xóa thành công hệ thống cập nhật lại danh sách hiển thị lên màn hình, không thành công hệ thống sẽ báo lỗi.



Hình 2.5: Biểu đồ hoạt động chức năng xóa

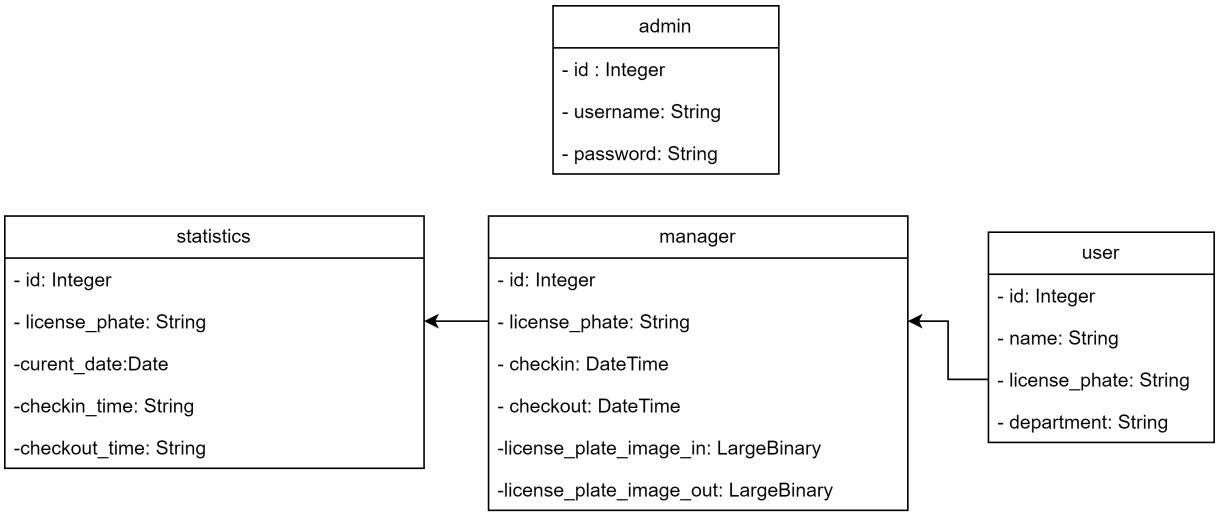
Trong hình 2.6, hệ thống sẽ lấy hình ảnh từ Raspberry Pi và gửi lên hệ thống sau đó hệ thống sẽ xác định vị trí biển số xe và tiến hành xử lý để lấy ra thông tin biển số xe mà chủ xe đang giữ.



Hình 2.6: Biểu đồ hoạt động chức nhận diện biển số xe

2.1.3.3. Biểu đồ cơ sở dữ liệu

Hệ thống được chia làm 4 bảng trên cơ sở dữ liệu: bảng admin, bảng manager, bảng user và bảng sattistics. Sự liên kết được thể hiện như hình 2.7.



Hình 2.7: Biểu đồ cơ sở dữ liệu

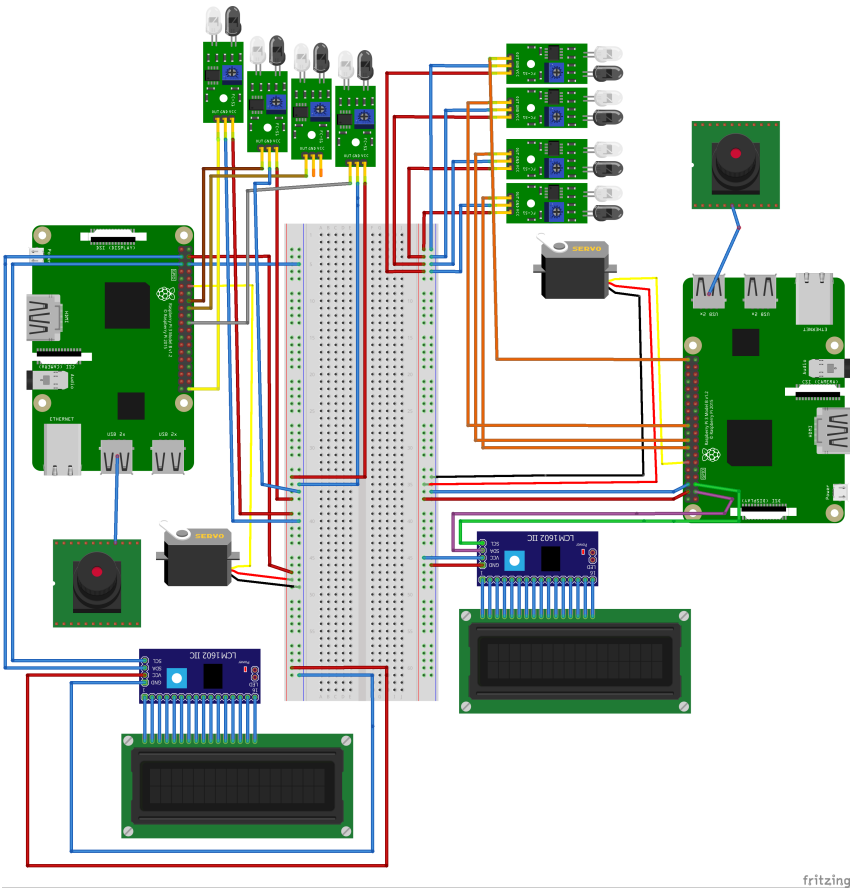
2.2. Thiết kế hệ thống

2.2.1. Phần cứng

Hệ thống bao gồm một rasbery pi3 và kết nối với các thiết bị camera, màn hình LCD I2C và các cảm biến hồng ngoại được nối các chân như hình 2.8.

Mỗi Raspberry Pi 3 được kết nối với một màn hình LCD 16x2, 4 cảm biến vật cản hồng ngoại, một động cơ servo và một camera USB. Sơ đồ được nối như sau:

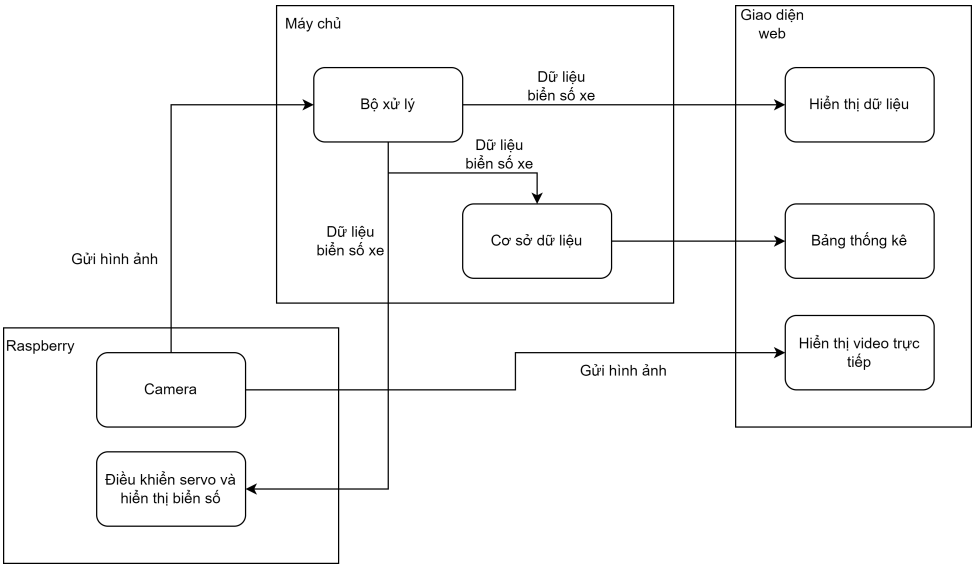
- Servo nối với chân: GPIO18
- LCD được nối với các chân: GPIO2(SDA1), GPIO3(SCL1)
- Cảm biến vật cản hồng ngoại được nối với các chân: GPIO23, GPIO24, GPIO25, GPIO21.



Hình 2.8: Mô hình đi chân của các thiết bị phần cứng

2.2.2. *Phần mềm*

Kiến trúc của toàn bộ hệ thống được thể hiện qua hình 2.9.



Hình 2.9: Kiến trúc phần mềm hệ thống

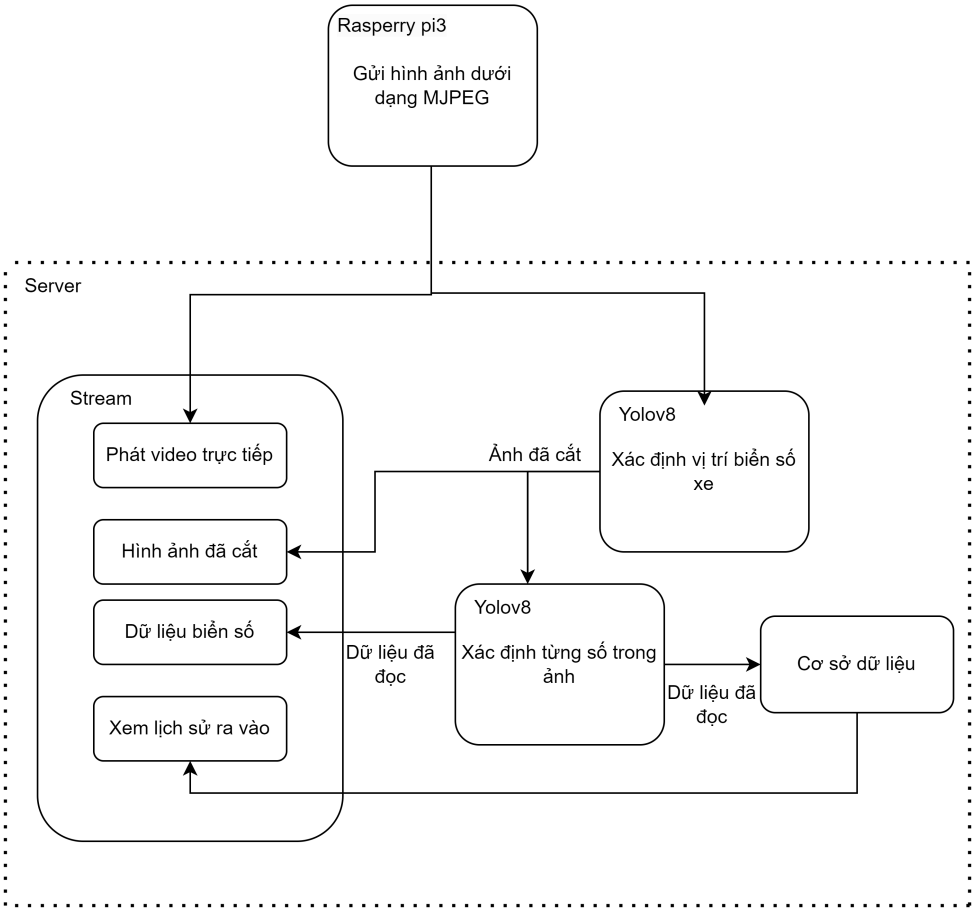
Hoạt động xử lý hình ảnh để đưa ra dữ liệu biển số xe tại PC được hoạt động theo hình 2.10:

Bước 1: Hệ thống lấy hình ảnh từ camera pi để phát video trực tiếp và đưa vào xử lý

Bước 2: Hệ thống sử dụng YOLOv8 để xác định vị trí biển số xe trong ảnh

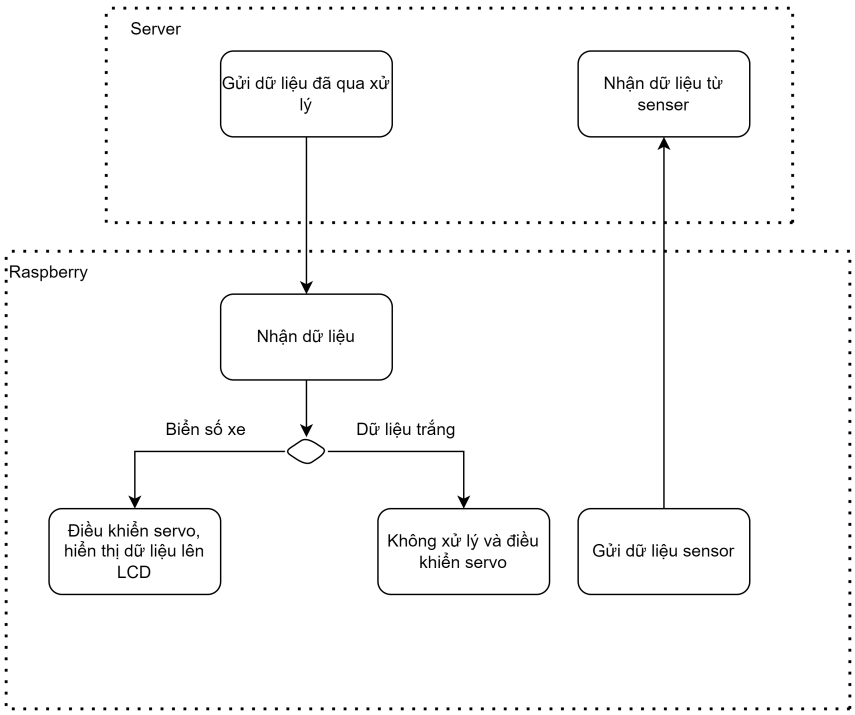
Bước 3: Hệ thống xử lý cắt ảnh chỉ bao gồm vùng chứa biển số xe

Bước 4: Hệ thống sử dụng YOLOv8 để xác định văn bản trong ảnh và trả về một chuỗi ký tự để hiển thị lên web và lưu lại vào cơ sở dữ liệu



Hình 2.10: Hoạt động xử lý hình ảnh tại bộ sử lý trung tâm (PC)

Tại Raspberyy Pi dữ liệu được gửi sang server như hình 2.11.



Hình 2.11: Hoạt động xử lý tại Raspberyy Pi

2.3. Phương pháp nhận diện biển số xe

2.3.1. Xác định vị trí trong ảnh

2.3.1.1. Chuẩn bị dataset

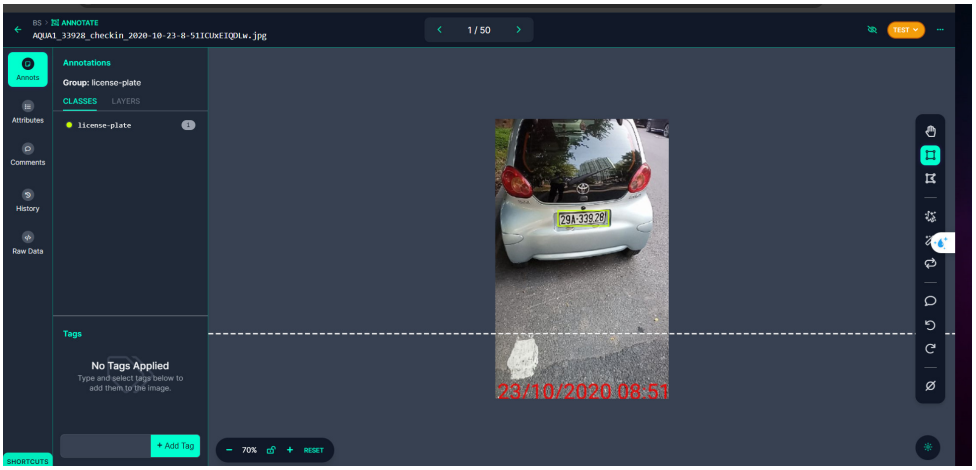
Bước 1: Lấy dữ liệu:

- Tải tệp dữ liệu từ các đường dẫn sau:
  - Dữ liệu xác định vị trí biển số xe trong ảnh: https://github.com/thigiac-maytinh/DataThiGiacMayTinh/blob/main/GreenParking.zip
  - Dữ liệu xác định các số trong biển số xe: https://universe.roboflow.com/franz-bpzvh/license-ocr-qqq6v
- Tệp dữ liệu xác định vị trí biển số xe chứa hơn 800 ảnh với biển số xe dài và ngắn.
- Tệp dữ liệu xác định chữ số và chữ cái trong biển bao gồm 9987 ảnh trong đó có các số từ 0-9 và chữ từ A-Z.

Bước 2: Gán nhãn trên Roboflow:

- Sử dụng Roboflow để gán nhãn cho mỗi ảnh trong tệp dữ liệu như hình 2.12 và hình 2.13
- Quá trình gán nhãn bao gồm đánh dấu bounding box (hộp giới hạn) xung quanh biển số xe, các số hoặc chữ cái để được gán nhãn tương ứng.
- Khi quá trình gán nhãn hoàn tất, hệ thống sẽ tự động chia dữ liệu thành ba thư mục:

– 70% dữ liệu dùng để huấn luyện mô hình.
– 20% dữ liệu dùng để kiểm tra mô hình.
– 10% dữ liệu dùng để xác thực mô hình.



Hình 2.12: Gán nhãn cho vị trí biển số xe trên roboflow

Hình 2.13: Gán nhãn cho số trong biển số xe trên roboflow

2.3.1.2. Huấn luyện dữ liệu

Bước 1: Cài đặt môi trường và thư viện:

- Cài đặt Python: Đảm bảo rằng đã cài đặt Python trên máy tính của mình.
- Cài đặt Ultralytics: Ultralytics là một framework hỗ trợ huấn luyện nhanh chóng và dễ dàng cho nhiều mô hình học máy, bao gồm cả YOLO.

Bước 2: Tiến hành huấn luyện dữ liệu:

Sử dụng framework YOLOv8 và Ultralytics để huấn luyện mô hình. Huấn luyện bằng lệnh sau:

yolo detect train data=data.yaml model=yolov8n.yaml epochs=100 imgsz=640 [**Glenn**]

Trong đó:

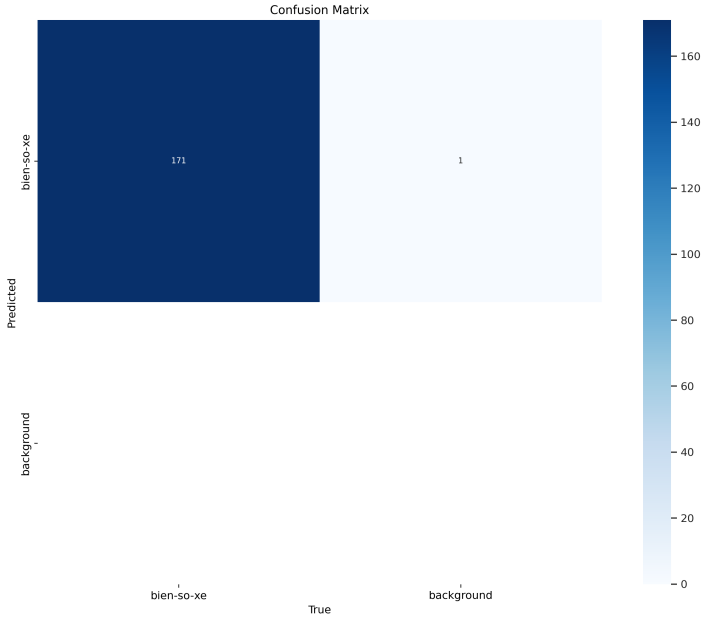
- yolo detect train: Là lệnh chính để thực hiện quá trình dự đoán (detect) sử dụng mô hình YOLO.
- data=data.yaml: Đường dẫn đến tệp dữ liệu YAML (data.yaml) chứa thông tin về tập dữ liệu và cấu hình huấn luyện. Trong tệp này bao gồm đường dẫn đến thư mục chứa hình ảnh, số lượng lớp, tên lớp, và các thông số khác.
- model=yolov8n.yaml: Đường dẫn đến file cấu hình của mô hình YOLOv8n (yolov8n.yaml). Trong file này, bao gồm thông số cấu hình của mô hình, bao gồm số lượng lớp, kích thước của grid, và các thông số khác.
- epochs=100: Số lượng epochs (vòng lặp) được sử dụng trong quá trình dự đoán. Trong trường hợp này, đặt là 100.
- imgsz=640: Kích thước ảnh đầu vào được sử dụng trong quá trình dự đoán. Trong trường hợp này, ảnh sẽ được thay đổi kích thước về kích thước 640x640 pixels.

```
!yolo train model=yolov8n.pt data=data.yaml epochs=30 imgsz=640

albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))
val: Scanning /content/a/valid/labels... 170 images, 0 backgrounds, 0 corrupt: 100% 170/170 [00:00<00:00, 2001.10it/s]
val: New cache created: /content/a/valid/labels.cache
Plotting labels to runs/detect/train2/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.002, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 0 dataloader workers
Logging results to runs/detect/train2
Starting training for 30 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       1/30         0G     0.9846      1.738       1.03          4        640: 100% 117/117 [27:04<00:00, 13.88s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:52<00:00,  8.68s/it]
                   all        170        170      0.971       0.98      0.989      0.718

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       2/30         0G      1.022       1.12      1.042          7        640: 100% 117/117 [26:59<00:00, 13.84s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:51<00:00,  8.58s/it]
                   all        170        170      0.789      0.859      0.898      0.625

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       3/30         0G      1.008     0.9056      1.054          4        640: 100% 117/117 [26:37<00:00, 13.65s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:54<00:00,  9.04s/it]
                   all        170        170      0.946      0.965      0.987      0.741

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       4/30         0G      0.983     0.7725      1.047          5        640: 100% 117/117 [26:37<00:00, 13.65s/it]
```

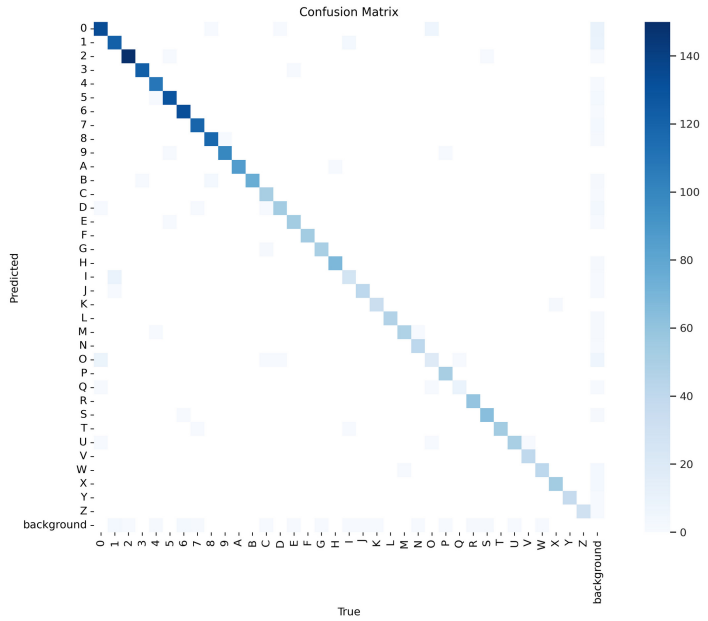
Hình 2.14: Huấn luyện dữ liệu bằng lệnh trên cmd

Sau quá trình huấn luyện, hệ thống đã tạo ra một loạt các biểu đồ để thể hiện kết quả đạt được [7]. Dưới đây là mô tả chi tiết về mỗi biểu đồ:

Confusion Matrix (Ma trận nhầm lẫn) được biểu diễn ở hình 2.15 và hình 2.16: Biểu đồ này minh họa số lượng dự đoán đúng và sai lệch của hệ thống trên từng nhóm lớp.



Hình 2.15: Biểu đồ confusion matrix cho nhận dạng vị trí biển số xe



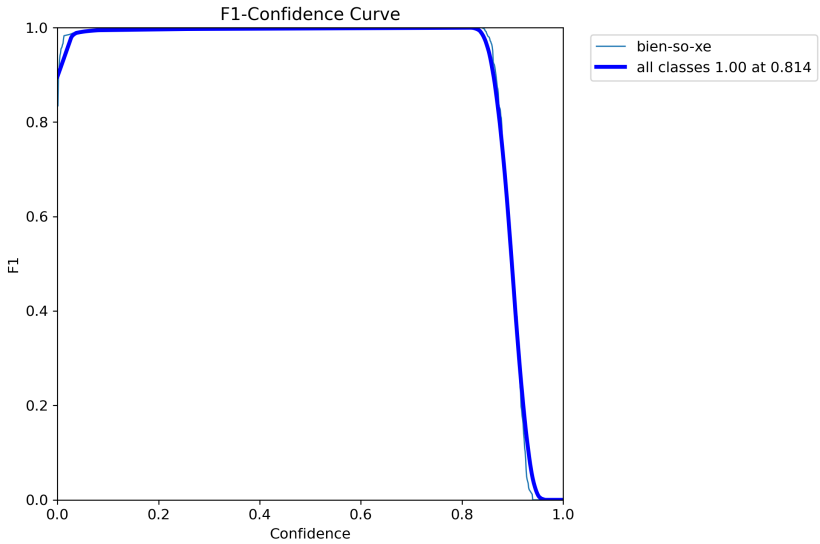
Hình 2.16: Biểu đồ confusion matrix cho nhận dạng ký tự trong biển số xe

Trên hình 2.15 có các số liệu như sau: có tổng cộng 171 ảnh được dự đoán đúng là biển số xe, trong khi có 1 ảnh được dự đoán là biển số xe nhưng thực tế lại là sai. Dựa trên các số liệu trên, ta có thể kết luận rằng tỉ lệ xác định vị trí biển số xe trong ảnh đạt độ chính xác 99.4%. Tỉ lệ này, được gọi là độ chính xác của việc dự đoán, cho thấy rằng mô hình có khả năng xác định biển số xe với mức độ tin cậy rất cao.

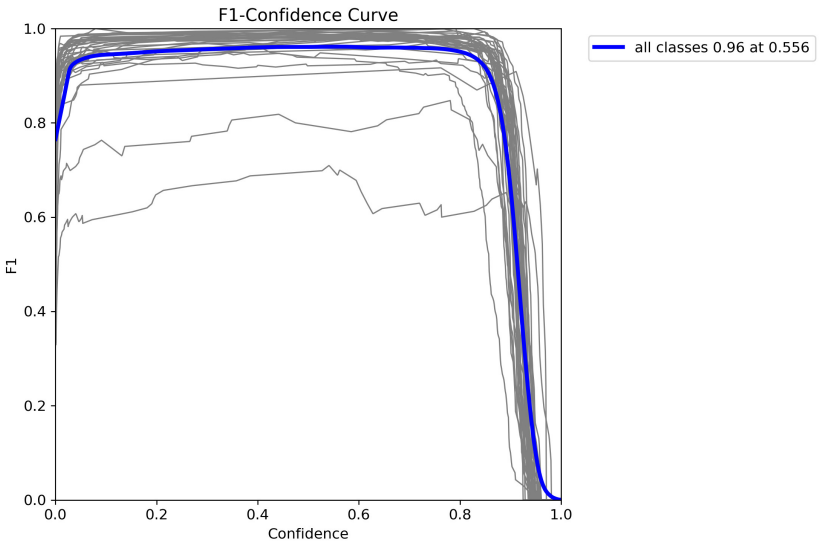
Hình 2.16 minh họa ma trận nhầm lẫn của mô hình nhận dạng số và chữ cái trong biển số xe. Dựa trên màu sắc của các ô trong ma trận nhầm lẫn, ta có thể nhận thấy rằng mô hình này có độ chính xác khá cao. Các ô chính dọc theo đường chéo của ma trận (đại diện cho các dự đoán đúng) có màu đậm, cho thấy tỷ lệ dự đoán đúng cao. Trong khi đó, các ô ngoài đường chéo (đại diện cho các dự đoán sai) có màu nhạt hơn nhiều, chứng tỏ số lượng dự đoán sai là rất ít. Qua hình trên khẳng định rằng mô hình có khả năng nhận dạng các số trong biển số xe với độ chính xác tốt.

F1_Curve (đường cong F1) là một biểu đồ biểu diễn mối quan hệ giữa độ chính xác (precision) và độ thu hồi (recall) của một mô hình học máy ở các ngưỡng khác nhau.

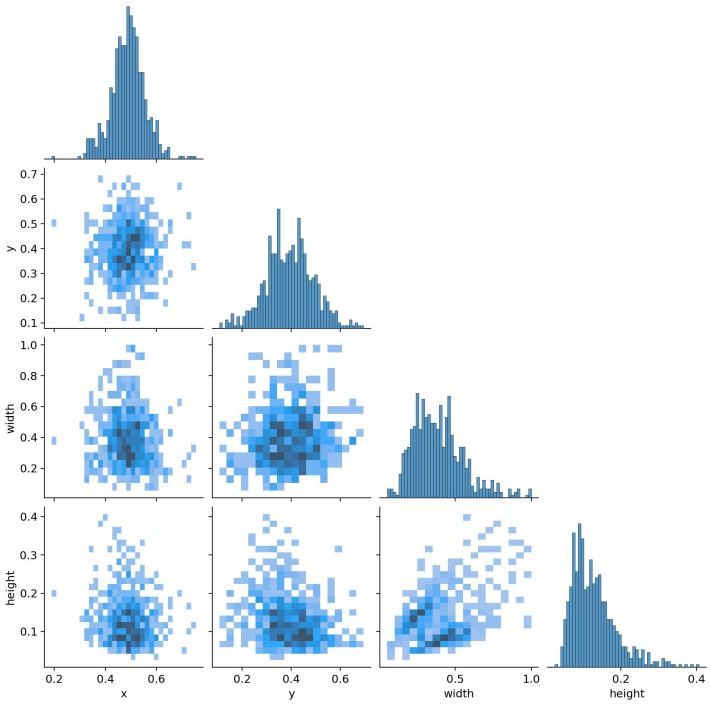
Tại hình 2.17 mô hình nhận diện vị trí biển số xe trong ảnh đã đạt được F1-Score hoàn hảo là 1.00 tại ngưỡng 0.814, trong khi tại hình 2.18 mô hình đạt được F1-Score là 0.96 tại ngưỡng 0.556. Điều này là minh chứng rõ ràng cho hiệu suất xuất sắc của mô hình, với độ chính xác và độ thu hồi gần như tuyệt đối cho tất cả các lớp. Ở cả hai ngưỡng này, mô hình đã thực hiện các dự đoán với độ chính xác cao và không gặp phải bất kỳ lỗi nào trong việc phân loại, tức là không có dự đoán sai sót nào xảy ra.



Hình 2.17: Biểu đồ F1-Confidence Curve cho nhận diện vị trí biển số xe

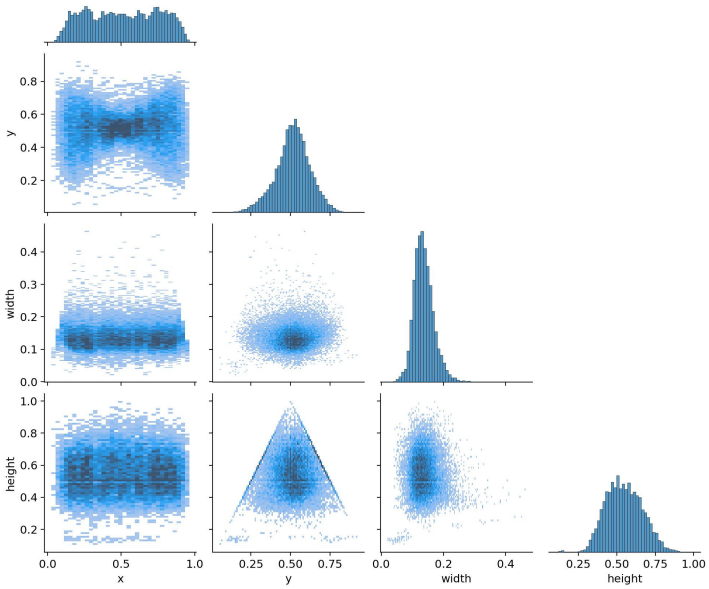


Hình 2.18: Biểu đồ F1-Confidence Curve cho nhận dạng số trong biển số xe



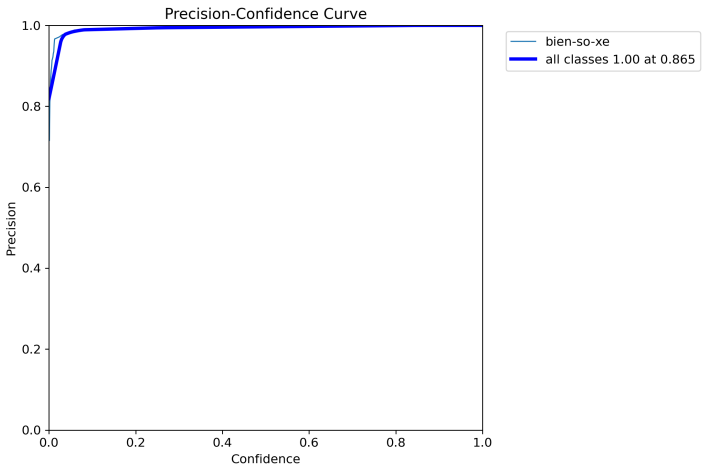
Hình 2.19: Biểu đồ Labels Correlogram cho nhận diện vị trí biển số xe

Labels Correlogram được biểu thị ở hình 2.19 và hình 2.20: Biểu đồ này thể hiện sự tương quan giữa các nhãn dự đoán và nhãn thực tế, giúp đánh giá sự liên quan giữa các lớp.

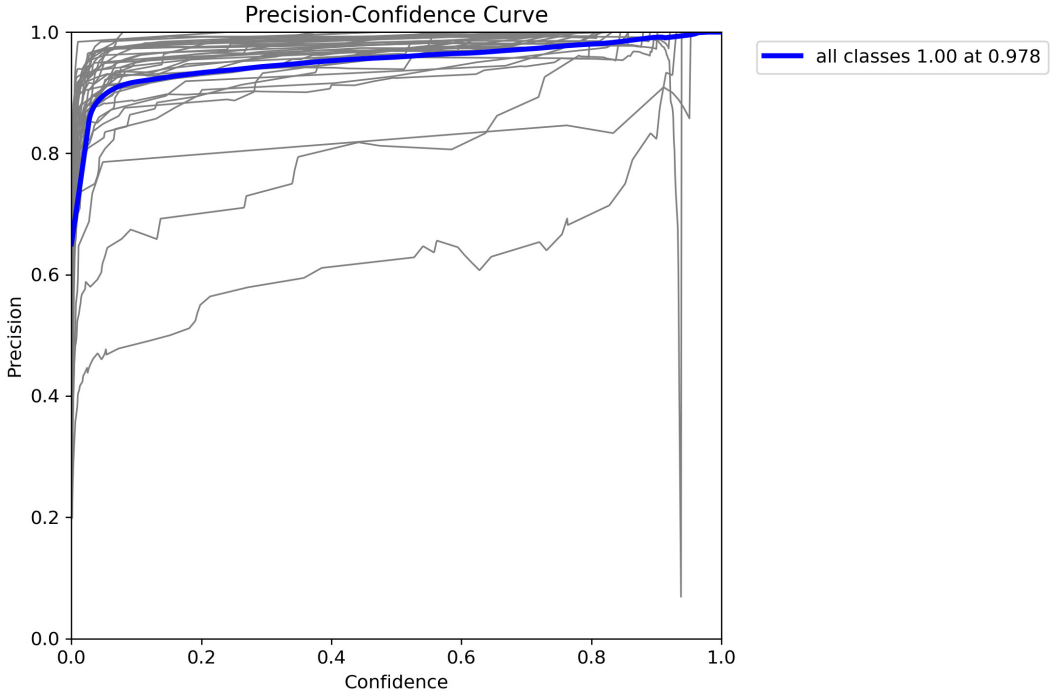


Hình 2.20: Biểu đồ Labels Correlogram cho nhận diện số trong biển số xe

Biểu đồ P Curve ở hình 2.21 và hình 2.22 thể hiện sự biến động giữa độ chính xác và độ phủ trên các ngưỡng xác định khác nhau. Đánh giá cho việc nhận diện vị trí biển số xe đã đạt được độ chính xác và độ phủ tất cả các lớp là 1.00 tại ngưỡng 0.865. Đối với việc nhận diện số trong biển số xe, mô hình cũng đạt được độ chính xác và độ phủ tất cả các lớp là 1.00 tại ngưỡng 0.978. Điều này cho thấy mô hình hoạt động với hiệu suất tuyệt vời, đảm bảo độ chính xác và độ phủ cao cho cả việc nhận diện vị trí biển số và nhận diện các số trong biển số.

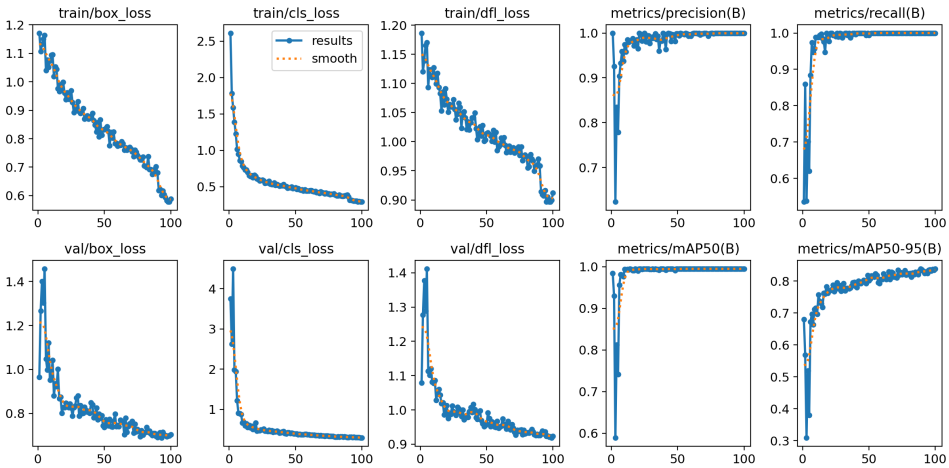


Hình 2.21: Biểu đồ P Curve cho nhận diện vị trí biển số xe

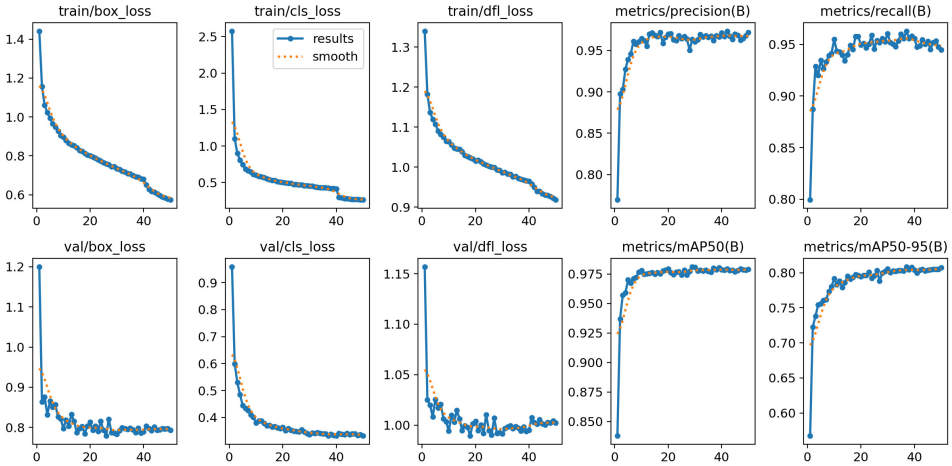


Hình 2.22: Biểu đồ P Curve cho nhận diện số trong biển số xe

Results được thể hiện tại hình 2.23 và hình 2.24: Biểu đồ này tổng hợp kết quả chính của mô hình, bao gồm độ chính xác, độ phủ, và các thước đo hiệu suất khác. Các biểu đồ trên giúp phân tích chi tiết về hiệu suất và độ chính xác của mô hình, từ đó cung cấp thông tin quan trọng để đánh giá và cải thiện hệ thống.



Hình 2.23: Biểu đồ Results cho nhận diện vị trí biển số xe



Hình 2.24: Biểu đồ Results cho nhận diện số trong biển số xe

2.3.2. Xử lý hình ảnh

Sau khi xác định được vị trí của biển số xe thông qua quá trình phát hiện, tiếp theo thực hiện các bước xử lý như cắt ảnh theo bounding box đã xác định để lấy vùng chứa biển số xe. Trong quá trình này, có thể thực hiện các thao tác như cắt ảnh, xoay ảnh, hoặc các xử lý hình ảnh khác để tối ưu hóa vùng chứa biển số xe:

Cắt ảnh theo Bounding Box:

- Sử dụng tọa độ của bounding box (x_{min}, y_{min}, x_{max}, y_{max}) để xác định vùng chứa biển số xe trên ảnh gốc.
- Cắt ảnh theo vùng này để thu được ảnh con chứa biển số.

Xoay ảnh:

- Nếu biển số xe có thể nghiêng hoặc xoay, có thể sử dụng các phương pháp xử lý hình ảnh để xoay ảnh và làm cho biển số nằm ngang và đúng chiều.
- Một số kỹ thuật thông dụng bao gồm xoay dựa trên góc nghiêng của biển số hoặc sử dụng kỹ thuật xoay tự động.

2.3.3. Sử dụng YOLOv8 để xác định biển số xe

Sau khi có được hình ảnh chỉ chứa biển số xe,chúng ta lại đưa hình ảnh vào trong mô hình YOLOv8 đã huấn luyện(Các bước huấn luyện dữ liệu giống phần 2.3.1) để lấy ra vị trí biển số xe như hình 2.25 các số có trong biển số xe như hình 2.26. Phần code ở xác định và lấy ra biển số xe nằm ở phần phụ lục

Hình 2.25: Ảnh xác định vị trí biển số xe

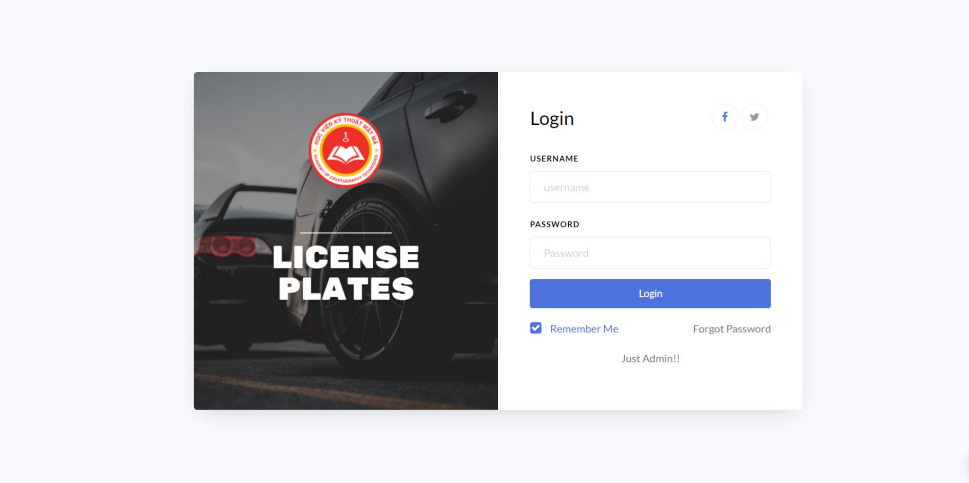
Hình 2.26: Biểu số xe sau khi đã xác định số

Chương 3
THỰC NGHIỆM

3.1. Các chức năng

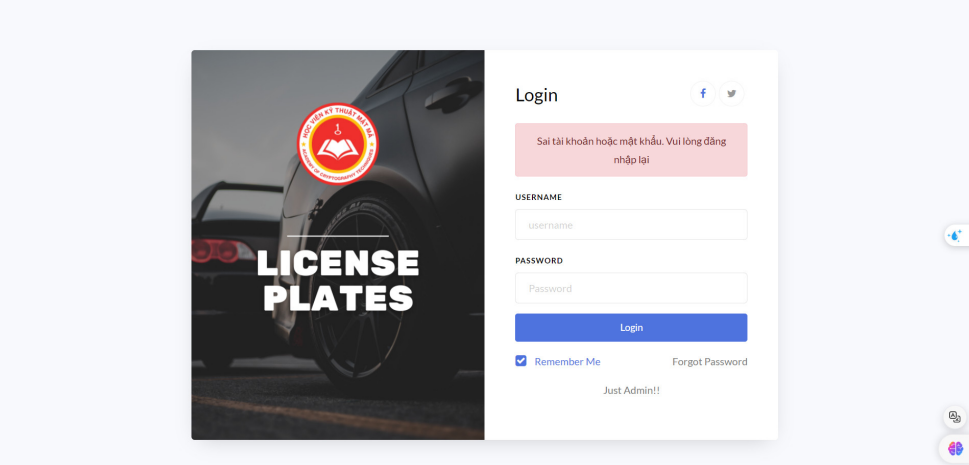
3.1.1. Chức năng đăng nhập

Trong hình 3.1 minh họa giao diện đăng nhập, nơi người dùng được yêu cầu nhập thông tin tên đăng nhập và mật khẩu để tiếp tục.



Hình 3.1: Giao diện đăng nhập của hệ thống

Khi người dùng nhập thông tin và thực hiện đăng nhập, hệ thống sẽ kiểm tra thông tin đầu vào và xác định xem chúng có hợp lệ hay không. Trong trường hợp thông tin đăng nhập không chính xác, hệ thống sẽ trả về một thông báo lỗi và yêu cầu người dùng nhập lại. Hình 3.2 mô tả giao diện hiển thị khi hệ thống phát hiện thông tin đăng nhập không chính xác.

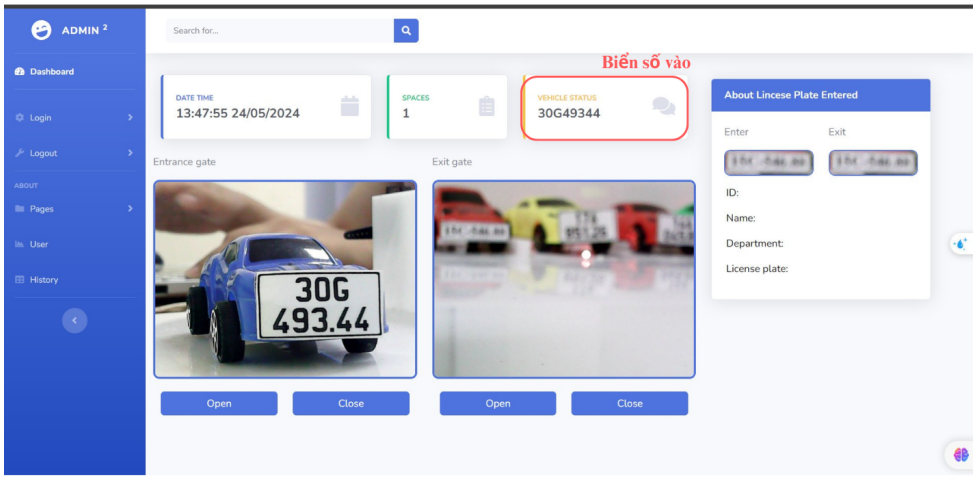


Hình 3.2: Giao diện đăng nhập sai

3.1.2. *Chức năng nhận diện biển số xe và phát video lên hệ thống*

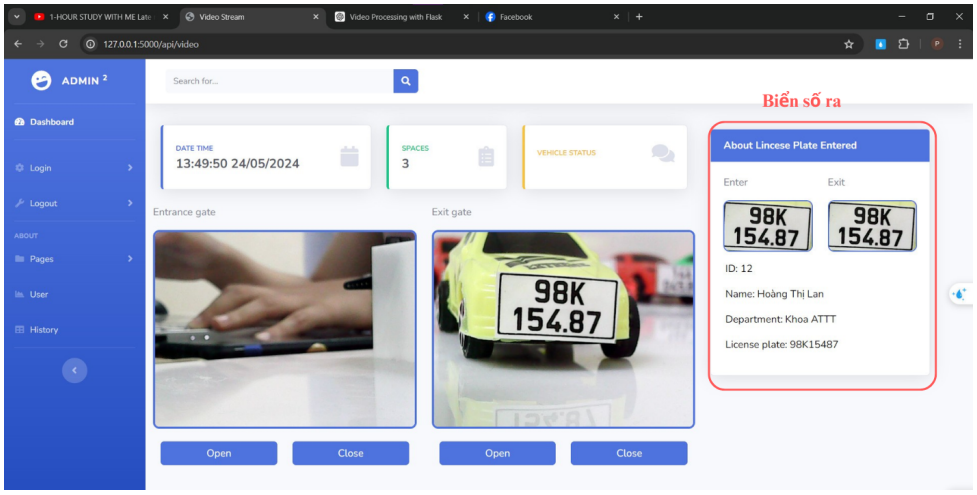
Video đã được tải lên và phát trực tiếp trên trang chủ, hiển thị hình ảnh từ camera chiều ra và camera chiều vào của hệ thống quản lý thông qua internet.

Đối với trường hợp xe đi vào, hình ảnh từ camera chiều vào được truyền về máy chủ để xử lý. Sau đó, hệ thống sẽ hiển thị biển số xe lên giao diện và tự động mở cổng sau một khoảng thời gian là 5 giây, trước khi tự động đóng lại. Hình 3.3 minh họa giao diện khi xe đi vào.



Hình 3.3: Giao diện xe đi vào

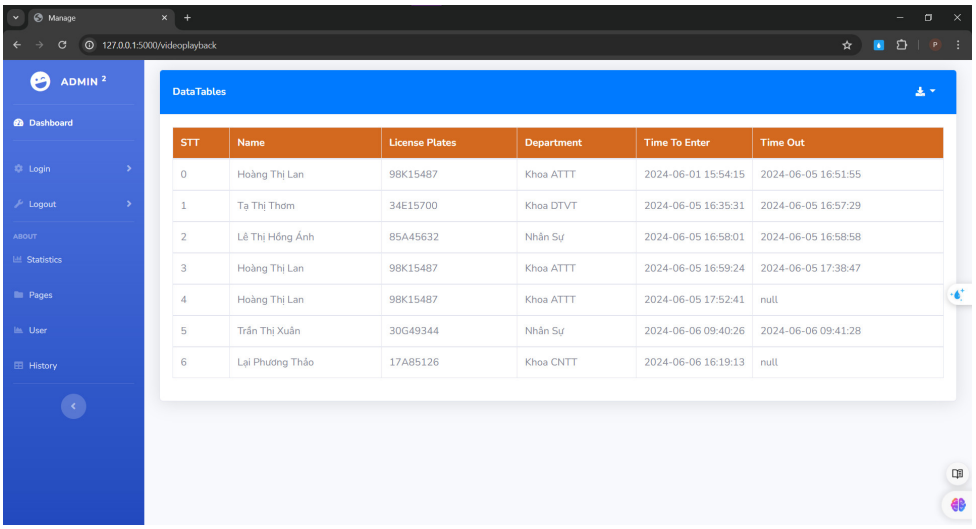
Đối với trường hợp xe ra, hệ thống sẽ truy xuất cơ sở dữ liệu để lấy hình ảnh và thông tin người dùng liên quan. Sau đó, thông tin này sẽ được hiển thị lên giao diện, như được thể hiện trong hình 3.4.



Hình 3.4: Giao diện xe đi ra

3.1.3. Chức năng xem lịch sử xe ra vào

Giao diện để xem trạng thái vào ra của xe tại học viện được trình bày như hình 3.5. Hệ thống cho phép người dùng xem thông tin chi tiết về việc xe ra vào, bao gồm thời gian cụ thể của việc ra vào, cũng như giờ ra và giờ vào của từng xe.



STT	Name	License Plates	Department	Time To Enter	Time Out
0	Hoàng Thị Lan	98K15487	Khoa ATTT	2024-06-01 15:54:15	2024-06-05 16:51:55
1	Tạ Thị Thơm	34E15700	Khoa DTVT	2024-06-05 16:35:31	2024-06-05 16:57:29
2	Lê Thị Hồng Ánh	85A45632	Nhân Sự	2024-06-05 16:58:01	2024-06-05 16:58:58
3	Hoàng Thị Lan	98K15487	Khoa ATTT	2024-06-05 16:59:24	2024-06-05 17:38:47
4	Hoàng Thị Lan	98K15487	Khoa ATTT	2024-06-05 17:52:41	null
5	Trần Thị Xuân	30G49344	Nhân Sự	2024-06-06 09:40:26	2024-06-06 09:41:28
6	Lại Phương Thảo	17A85126	Khoa CNTT	2024-06-06 16:19:13	null

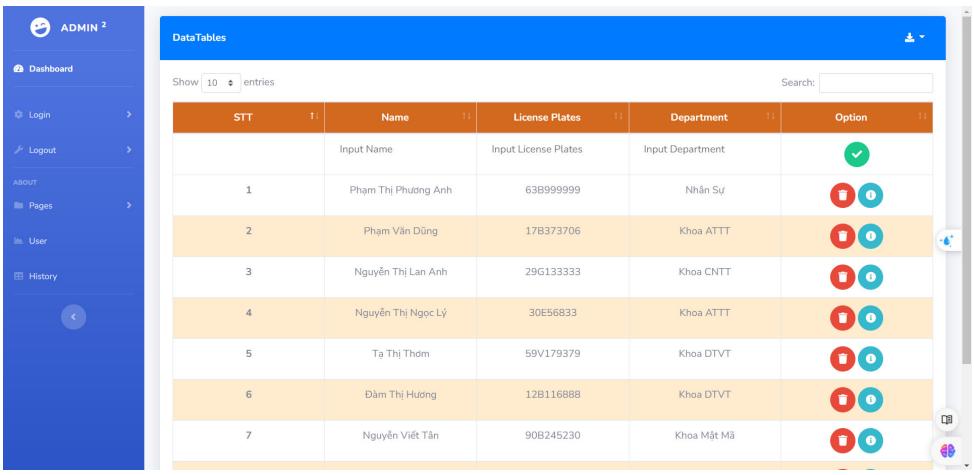
Hình 3.5: Giao diện xem trạng thái ra vào

Trong giao diện này, mỗi dòng tương ứng với một lần xe ra vào. Các cột hiển thị thông tin như sau:

- STT: Định danh duy nhất của lần xe ra vào.
- Biển số xe: Biển số xe của phương tiện.
- Thời gian vào: Thời điểm xe vào học viện.
- Thời gian ra: Thời điểm xe ra khỏi học viện, nếu xe vẫn đang ở trong học viện thì giá trị sẽ là null.

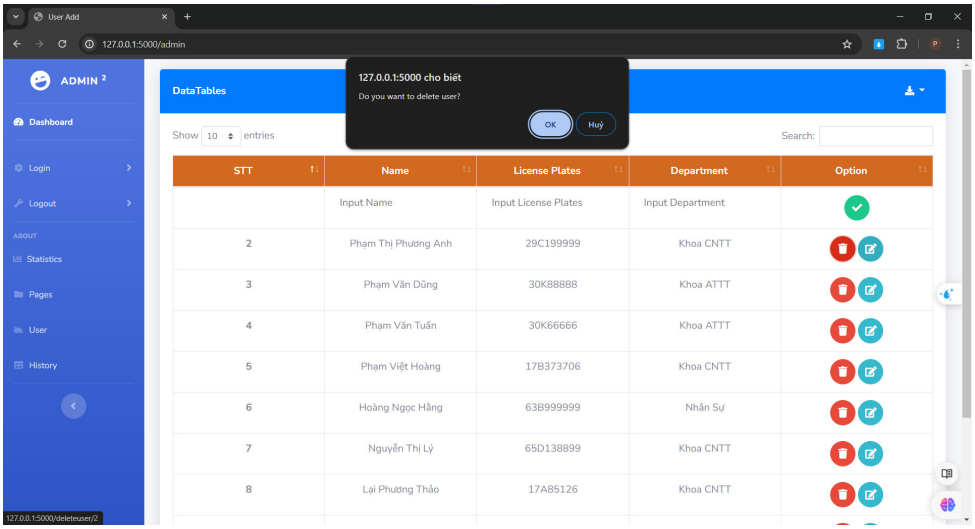
3.1.4. Chức năng quản lý người dùng

Giao diện thêm người dùng như hình 3.6, cho phép người quản trị nhập thông tin người dùng mới vào hệ thống. Trong giao diện này, có chức năng tìm kiếm thông tin tại các trường họ tên, biển số xe và phòng ban khi người dùng nhập dữ liệu vào ô tìm kiếm.



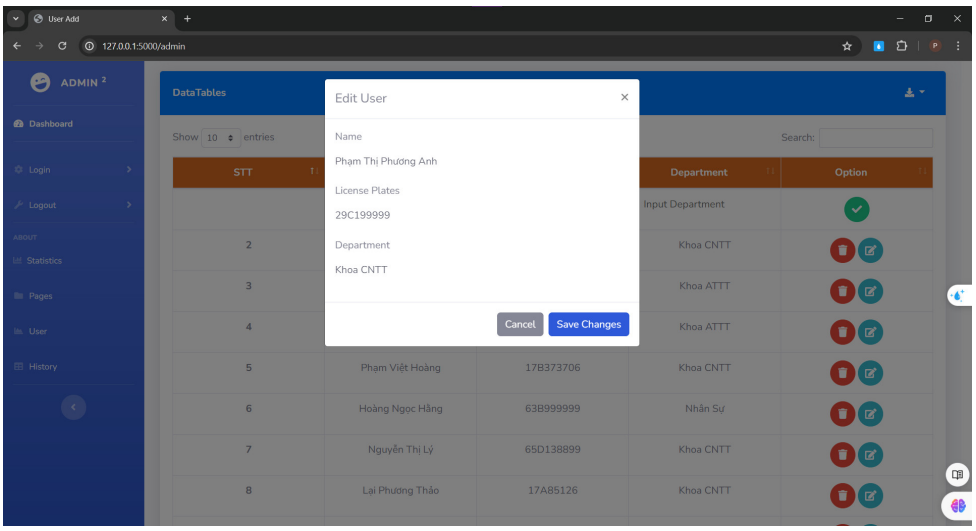
Hình 3.6: Giao diện thêm người dùng

Để xóa thông tin của một người dùng, người quản trị có thể nhấn vào biểu tượng thùng rác trên giao diện người dùng, sau đó hệ thống sẽ hiển thị một thông báo xác nhận để người dùng xác nhận việc xóa, như hình 3.7.



Hình 3.7: Giao diện xóa người dùng

Khi muốn chỉnh sửa thông tin của một người dùng, người quản trị có thể nhấn vào biểu tượng chỉnh sửa trên màn hình người dùng và sửa thông tin họ và tên, biển số, và phòng ban như hình 3.8.

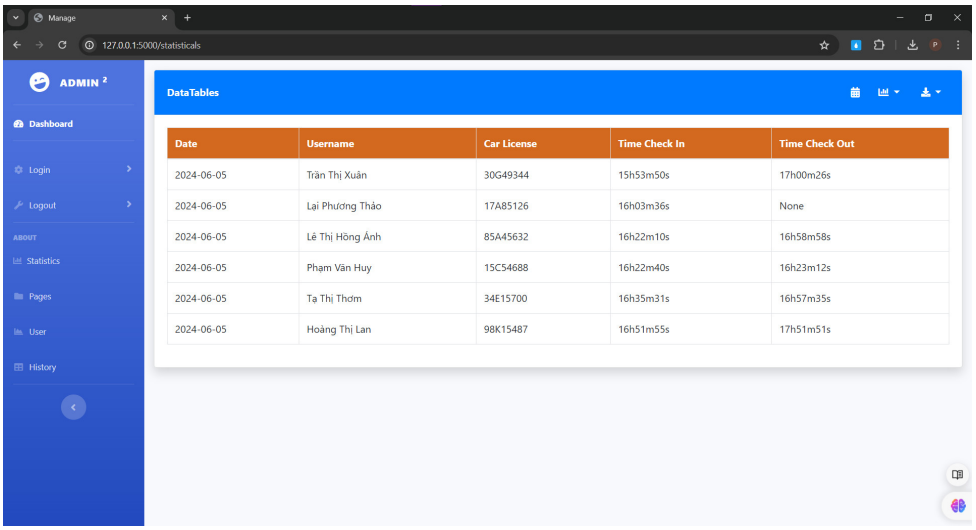


Hình 3.8: Giao diện chỉnh sửa người dùng

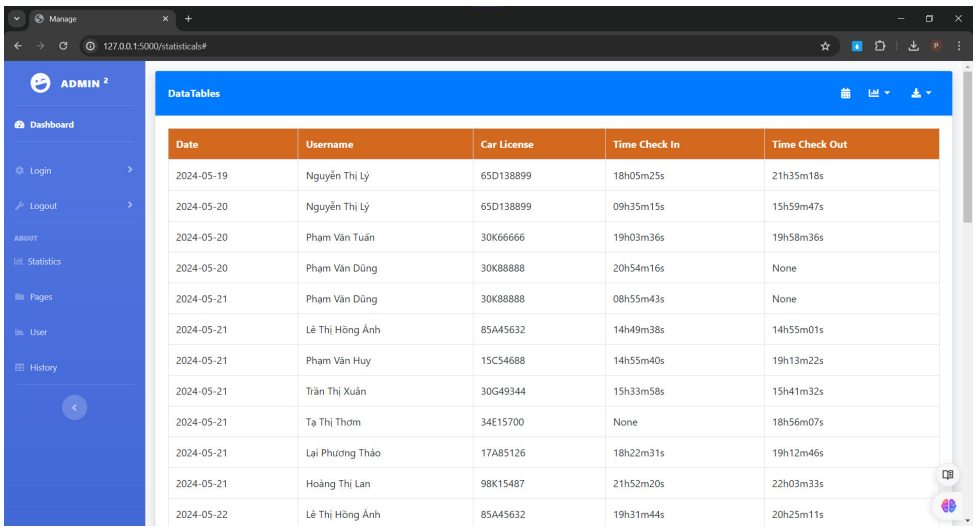
Các chức năng này giúp người quản trị dễ dàng quản lý và điều chỉnh thông tin người dùng trong hệ thống một cách thuận tiện và hiệu quả.

3.1.5. Chức năng thống kê

Hệ thống có các chức năng thống kê theo ngày như hình 3.9, thống kê theo tháng như hình 3.10, thống kê ngày làm việc của người dùng trong tháng như hình 3.11.

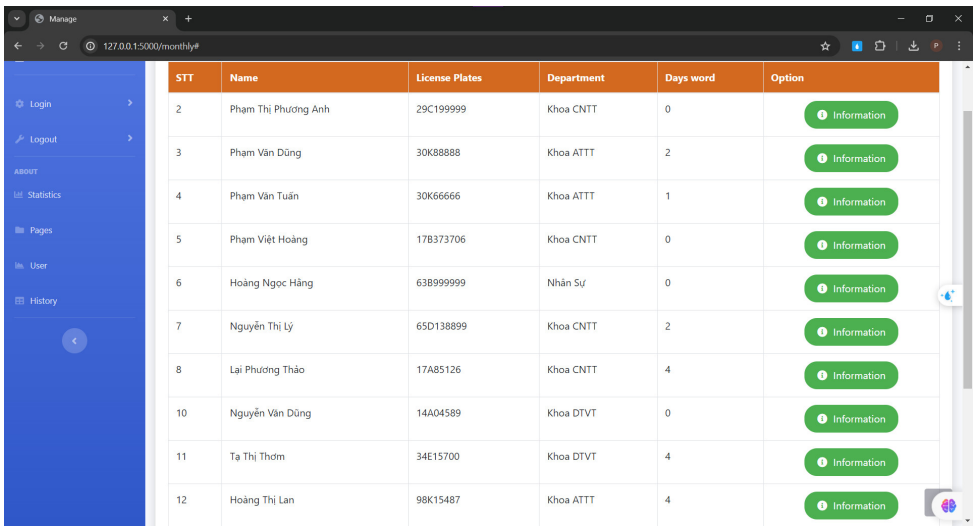


Hình 3.9: Giao diện thống kê theo ngày



Date	Username	Car License	Time Check In	Time Check Out
2024-05-19	Nguyễn Thị Lý	65D138899	18h05m25s	21h35m18s
2024-05-20	Nguyễn Thị Lý	65D138899	09h35m15s	15h59m47s
2024-05-20	Phạm Văn Tuấn	30K66666	19h03m36s	19h58m36s
2024-05-20	Phạm Văn Dũng	30K88888	20h54m16s	None
2024-05-21	Phạm Văn Dũng	30K88888	08h55m43s	None
2024-05-21	Lê Thị Hồng Ánh	85A45632	14h49m38s	14h55m01s
2024-05-21	Phạm Văn Huy	15C54688	14h55m40s	19h13m22s
2024-05-21	Trần Thị Xuân	30G49344	15h33m58s	15h41m32s
2024-05-21	Tạ Thị Thơm	34E15700	None	18h56m07s
2024-05-21	Lại Phương Thảo	17A85126	18h22m31s	19h12m46s
2024-05-21	Hoàng Thị Lan	98K15487	21h52m20s	22h03m33s
2024-05-22	Lê Thị Hồng Ánh	85A45632	19h31m44s	20h25m11s

Hình 3.10: Giao diện thống kê theo tháng

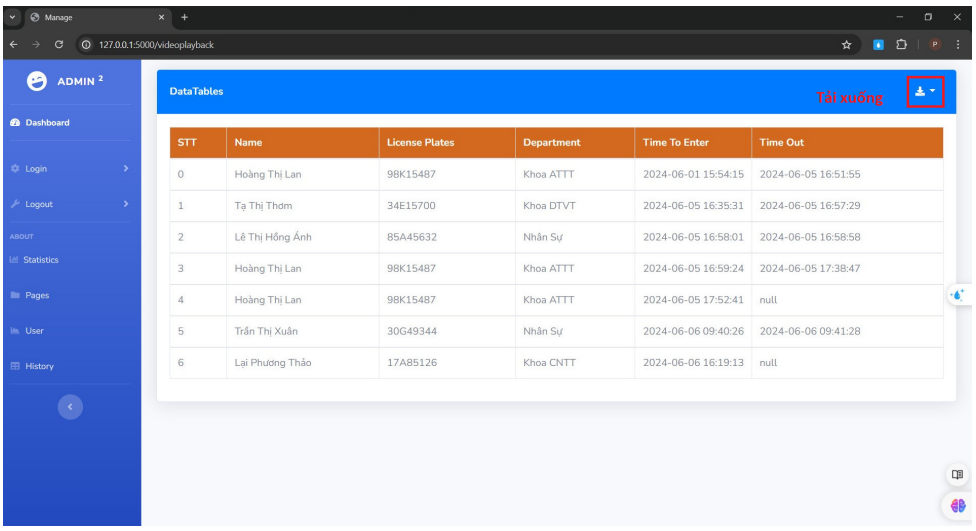


STT	Name	License Plates	Department	Days word	Option
2	Phạm Thị Phương Anh	29C199999	Khoa CNTT	0	Information
3	Phạm Văn Dũng	30K88888	Khoa ATTT	2	Information
4	Phạm Văn Tuấn	30K66666	Khoa ATTT	1	Information
5	Phạm Việt Hoàng	17B373706	Khoa CNTT	0	Information
6	Hoàng Ngọc Hằng	63B999999	Nhân Sự	0	Information
7	Nguyễn Thị Lý	65D138899	Khoa CNTT	2	Information
8	Lại Phương Thảo	17A85126	Khoa CNTT	4	Information
10	Nguyễn Văn Dũng	14A04589	Khoa DTVT	0	Information
11	Tạ Thị Thơm	34E15700	Khoa DTVT	4	Information
12	Hoàng Thị Lan	98K15487	Khoa ATTT	4	Information

Hình 3.11: Giao diện thống kê số ngày làm việc theo tháng

3.1.6. Chức năng tải dữ liệu dưới dạng CSV

Hệ thống có thể tải dữ liệu dưới dạng CSV theo các bảng người dùng như hình 3.14, lịch sử ra vào như hình 3.16, thống kê theo ngày như hình 3.15, thống kê theo tháng như hình 3.17, và thông số ngày làm việc của nhân viên như hình 3.13.



Hình 3.12: Giao diện tải dữ liệu dưới dạng CSV

STT	Name	Lincese Plate	Department	Day Work	Column6
2	Phạm Thị Phương Anh	29C199999	Khoa CNTT	0	Information
3	Phạm Văn Dũng	30K88888	Khoa ATTT	2	Information
4	Phạm Văn Tuấn	30K66666	Khoa ATTT	1	Information
5	Phạm Việt Hoàng	17B373706	Khoa CNTT	0	Information
6	Hoàng Ngọc Hằng	63B999999	Nhân Sự	0	Information
7	Nguyễn Thị Lý	65D138899	Khoa CNTT	2	Information
8	Lại Phương Thảo	17A85126	Khoa CNTT	4	Information
10	Nguyễn Văn Dũng	14A04589	Khoa DTVT	0	Information
11	Tạ Thị Thơm	34E15700	Khoa DTVT	4	Information
12	Hoàng Thị Lan	98K15487	Khoa ATTT	4	Information
13	Đặng Văn Quân	17A14519	Khoa CNTT	3	Information
14	Trần Thị Xuân	30G49344	Nhân Sự	6	Information
15	Phạm Văn Huy	15C54688	Khoa CNTT	4	Information
16	Lê Thị Hồng Ánh	85A45632	Nhân Sự	10	Information

Hình 3.13: Mở dữ liệu đã tải về ở dạng CSV bảng thống kê số ngày làm việc trong tháng

	A	B	C	D
1	ID	Name	License Plates	Department
2	2	Phạm Thị Phương Anh	29C199999	Khoa CNTT
3	3	Phạm Văn Dũng	30K88888	Khoa ATTT
4	4	Phạm Văn Tuấn	30K66666	Khoa ATTT
5	5	Phạm Việt Hoàng	17B373706	Khoa CNTT
6	6	Hoàng Ngọc Hằng	63B999999	Nhân Sự
7	7	Nguyễn Thị Lý	65D138899	Khoa CNTT
8	8	Lại Phương Thảo	17A85126	Khoa CNTT
9	10	Nguyễn Văn Dũng	14A04589	Khoa DTVT
10	11	Tạ Thị Thơm	34E15700	Khoa DTVT
11	12	Hoàng Thị Lan	98K15487	Khoa ATTT
12	13	Đặng Văn Quân	17A14519	Khoa CNTT
13	14	Trần Thị Xuân	30G49344	Nhân Sự
14	15	Phạm Văn Huy	15C54688	Khoa CNTT

Hình 3.14: Mở dữ liệu đã tải về ở dạng CSV bảng người dùng

	A	B	C	D	E
1	Column1	Column2	Column3	Column4	Column5
2	6/5/2024	Trần Thị Xuân	30G49344	15h53m50s	17h00m26s
3	6/5/2024	Lại Phương Thảo	17A85126	16h03m36s	None
4	6/5/2024	Lê Thị Hồng Ánh	85A45632	16h22m10s	16h58m58s
5	6/5/2024	Phạm Văn Huy	15C54688	16h22m40s	16h23m12s
6	6/5/2024	Tạ Thị Thơm	34E15700	16h35m31s	16h57m35s
7	6/5/2024	Hoàng Thị Lan	98K15487	16h51m55s	17h51m51s

Hình 3.15: Mở dữ liệu đã tải về ở dạng CSV bảng thống kê theo ngày

	A	B	C	D	E	F
1	ID	License Plates	Checkin	Checkout		
2	1	63B999999	5/17/2024 14:34	5/17/2024 15:35		
3	2	63B999999	5/17/2024 15:35	5/17/2024 15:36		
4	3	63B999999	5/17/2024 18:49	5/17/2024 18:50		
5	4	63B999999	5/17/2024 18:50	5/18/2024 9:23		
6	5	29C199999	5/18/2024 9:26	5/18/2024 9:27		
7	6	29C199999	5/18/2024 9:27	5/18/2024 9:27		
8	7	29C199999	5/18/2024 13:00	5/18/2024 13:00		
9	8	29C199999	5/18/2024 13:01	5/18/2024 13:01		
10	9	29C199999	5/18/2024 13:01	5/18/2024 13:01		
11	10	29C199999	5/18/2024 13:01	5/18/2024 13:02		
12	11	65X7855	5/18/2024 13:04	5/18/2024 13:04		
13	12	65D138899	5/18/2024 13:05	5/18/2024 13:06		
14	13	65D138899	5/18/2024 13:06	5/18/2024 13:06		
15	14	65D138899	5/18/2024 13:11	5/18/2024 13:13		
16	15	65D138899	5/18/2024 13:15	5/19/2024 18:05		
17	16	65D138899	5/19/2024 18:06	5/19/2024 18:26		
18	17	65D138899	5/19/2024 18:27	5/19/2024 18:32		
19	18	65D138899	5/19/2024 18:35	5/19/2024 18:41		
20	19	65D138899	5/19/2024 19:02	5/19/2024 19:26		
21	20	65D138899	5/19/2024 19:27	5/19/2024 21:35		
22	21	65D138899	5/20/2024 9:35	5/20/2024 9:36		
23	22	65D138899	5/20/2024 12:50	5/20/2024 12:57		
24	23	65D138899	5/20/2024 12:59	5/20/2024 13:02		
25	24	65D138899	5/20/2024 13:03	5/20/2024 13:04		
26	25	65D138899	5/20/2024 13:07	5/20/2024 13:09		
27	26	65D138899	5/20/2024 13:09	5/20/2024 13:13		
28	27	65D138899	5/20/2024 13:14	5/20/2024 15:50		
29	28	65D138899	5/20/2024 15:52	5/20/2024 15:53		
30	29	65D138899	5/20/2024 15:55	5/20/2024 15:55		
31	30	65D138899	5/20/2024 15:56	5/20/2024 15:57		
32	31	65D138899	5/20/2024 15:58	5/20/2024 15:58		

Hình 3.16: Mở dữ liệu đã tải về ở dạng CSV bảng lịch sử ra vào

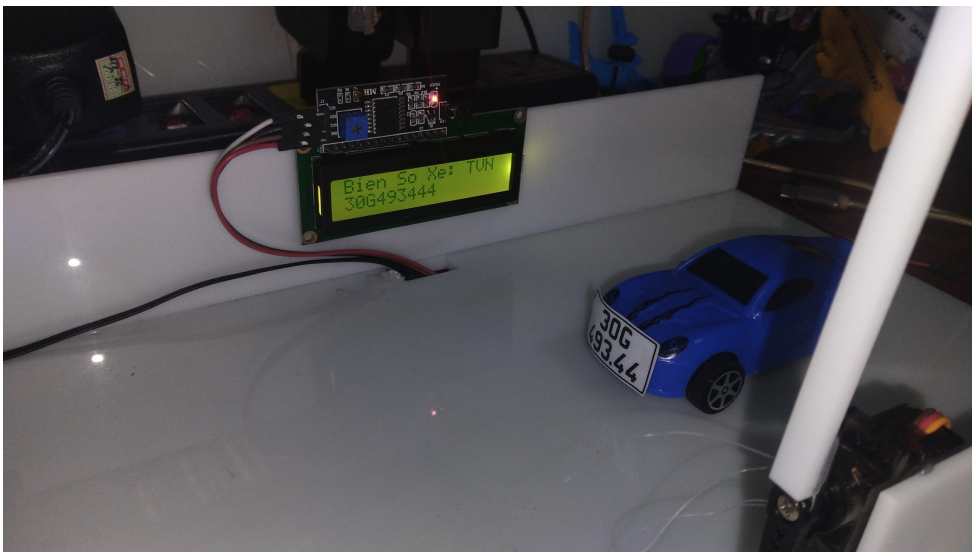
	A	B	C	D	E	F
1	Column1	Column2	Column3	Column4	Column5	
2	6/1/2024	Tạ Thị Thơm	34E15700	15h51m56s	None	
3	6/1/2024	Hoàng Thị Lan	98K15487	15h54m15s	None	
4	6/2/2024	Lại Phương Thảo	17A85126	14h36m53s	14h39m05s	
5	6/3/2024	Tạ Thị Thơm	34E15700	18h44m38s	None	
6	6/3/2024	Trần Thị Xuân	30G49344	19h10m12s	None	
7	6/5/2024	Trần Thị Xuân	30G49344	15h53m50s	17h00m26s	
8	6/5/2024	Lại Phương Thảo	17A85126	16h03m36s	None	
9	6/5/2024	Lê Thị Hồng Ánh	85A45632	16h22m10s	16h58m58s	
10	6/5/2024	Phạm Văn Huy	15C54688	16h22m40s	16h23m12s	
11	6/5/2024	Tạ Thị Thơm	34E15700	16h35m31s	16h57m35s	
12	6/5/2024	Hoàng Thị Lan	98K15487	16h51m55s	17h51m51s	
13	6/6/2024	Trần Thị Xuân	30G49344	09h40m26s	09h41m28s	
14	6/6/2024	Lại Phương Thảo	17A85126	16h19m13s	None	
15						

Hình 3.17: Mở dữ liệu đã tải về ở dạng CSV bảng thống kê theo tháng

3.2. Hệ thống phần cứng

Dưới đây là phần demo mô hình của hệ thống bao gồm các chức năng mở, đóng barie (Hình 3.18), hiển thị biển số xe lên màn hình LCD (Hình 3.19)và các cảm biến vật cản để phát hiện những vị trí đỗ xe còn trống (Hình 3.20).

Hình 3.18: Mô hình sản phẩm



Hình 3.19: Hình ảnh hiển thị biển số xe lên LCD

Hình 3.20: Hình ảnh cảm biến vật cản

3.3. Đánh giá

Với hệ thống đã xây dựng, em đã lập các kịch bản kiểm thử cho hoạt động của hệ thống phần cứng và hệ thống xử lý nhận diện biển số xe như sau (Bảng 3.1):

Hệ thống phần mềm xử lý biển số xe:

- Kiểm thử với 100 biển số xe sạch, điều kiện ánh sáng tốt, thẳng:
  - Mục tiêu: Đánh giá độ chính xác của hệ thống trong điều kiện lý tưởng.
  - Kết quả mong đợi: Hệ thống nhận diện chính xác tất cả các biển số xe.
- Kiểm thử với 100 biển số nghiêng, bị mất một phần:

  - Mục tiêu: Đánh giá khả năng của hệ thống trong việc nhận diện các biển số xe không hoàn hảo.
  - Kết quả mong đợi: Hệ thống nhận diện chính xác một tỷ lệ cao các biển số xe bị nghiêng hoặc mất một phần.
- Kiểm thử với 100 biển số bị che hoặc trong điều kiện ánh sáng yếu:
  - Mục tiêu: Đánh giá độ bền vững và tính chính xác của hệ thống trong điều kiện không lý tưởng.
  - Kết quả mong đợi: Hệ thống vẫn nhận diện chính xác một tỷ lệ nhất định các biển số xe bị che hoặc trong điều kiện ánh sáng yếu.

Hệ thống quản lý, kiểm tốc độ và độ hiển thị chính xác trên LCD:

- Kiểm thử với 100 lần xe vào đối với chức năng hiển thị biển số:
  - Mục tiêu: Đảm bảo hệ thống hiển thị chính xác biển số xe khi xe vào.
  - Kết quả mong đợi: LCD hiển thị chính xác tất cả 100 lần biển số xe vào.
- Kiểm thử với 100 lần xe ra đối với chức năng hiển thị biển số:
  - Mục tiêu: Đảm bảo hệ thống hiển thị chính xác biển số xe khi xe ra.
  - Kết quả mong đợi: LCD hiển thị chính xác tất cả 100 lần biển số xe ra.
- Kiểm thử với 100 lần xe vào với chức năng mở, đóng barie:
  - Mục tiêu: Đánh giá độ ổn định và hiệu quả của hệ thống mở, đóng barie khi xe vào.
  - Kết quả mong đợi: Barie mở và đóng chính xác tất cả 100 lần khi xe vào.
- Kiểm thử với 100 lần xe ra đối với chức năng mở, đóng barie:
  - Mục tiêu: Đánh giá độ ổn định và hiệu quả của hệ thống mở, đóng barie khi xe ra.
  - Kết quả mong đợi: Barie mở và đóng chính xác tất cả 100 lần khi xe ra.

Bảng 3.1: Kết quả đánh giá chức năng với hệ thống phần cứng

STT	Kịch Bản kiểm thử	Số Lượng kiểm thử	Số lượng Kiểm thử Đúng	Số lượng kiểm thử sai	Tỉ lệ
1	Kiểm thử chức năng nhận diện biển số xe (ánh sáng đầy đủ, biển số sạch không nghiêng lệch)	100	91	9	91%
2	Biển số bị nghiêng lệch và bị cắt một phần	100	87	13	87%
3	Biển số xe trong điều kiện tối, bẩn	100	69	31	69%
4	Hệ thống hiển thị biển số xe vào lên LCD	100	97	3	97%
5	Hệ thống hiển thị biển số xe ra lên LCD	100	95	5	95%
6	Chức năng servo hoạt động cổng vào	100	69	31	69%
7	Chức năng servo hoạt động cổng ra	100	63	37	63%

Thông qua bảng 3.1, có thể nhận thấy hệ thống nhận diện biển số xe hoạt động với độ chính xác cao khi điều kiện ánh sáng đầy đủ. Kết quả này cho thấy khả năng xử lý hình ảnh và nhận diện biển số của hệ thống là đáng tin cậy trong môi trường có ánh sáng tốt, đảm bảo hiệu quả trong các ứng dụng thực tế như giám sát giao thông và quản lý phương tiện.

Tuy nhiên, chức năng đóng mở servo trong hệ thống vẫn chưa đạt được sự ổn định mong muốn. Các thử nghiệm cho thấy servo đôi khi hoạt động không chính xác hoặc không phản hồi kịp thời, dẫn đến sự gián đoạn trong quá trình vận hành tổng thể. Điều này cho thấy cần có sự cải tiến thêm về mặt kỹ thuật và phần cứng để đảm bảo servo hoạt động đồng bộ và hiệu quả hơn với các chức năng khác của hệ thống.

Việc cải thiện độ ổn định của servo là rất cần thiết để nâng cao tính ứng dụng và độ tin cậy của hệ thống trong các điều kiện thực tế đa dạng. Các giải pháp có thể bao gồm việc tối ưu hóa thuật toán điều khiển, nâng cấp phần cứng hoặc cải tiến cơ cấu cơ khí của servo.

KẾT LUẬN

1. Kết quả đạt được

Sau một khoảng thời gian thực hiện đề tài đến nay, dưới sự hướng dẫn tận tình của ThS Lê Đức Thuận và ThS Mạc Văn Hải, em đã hoàn thành đồ án tốt nghiệp với đề tài: ”Xây dựng hệ thống kiểm soát ra vào tại Học viện Kỹ thuật mật mã”.

Thông qua 3 chương đã trình bày, thể hiện đồ án đã đạt được một số kết quả như sau:

- Mô hình hóa cụ thể một hệ thống gửi xe bao gồm: mô hình thực nghiệm, phần mềm quản lý trên PC.

- Đối với phần mềm quản lý: hệ thống có sự tương tác qua lại tương đối với các thiết bị phần cứng và giao diện quản lý:

- Camera chụp được ảnh
- Động cơ servo hoạt động có thể tự động mở barrier cho xe vào/ra.
- Thực hiện quá trình nhận diện biển số, lưu ảnh.
- Sử dụng cảm biến hồng ngoại để đếm số lượng chỗ để xe còn trống.

- Trong quá trình nghiên cứu và hoàn thiện đồ án cá nhân em đã nghiên cứu và áp dụng được nhiều kiến thức như:

- Nắm được kiến thức về mô hình học sâu CNN và YOLOv8 và cách hoạt động của chúng.
- Nắm được kiến thức cơ bản về các thuật toán xử lý ảnh để có thể áp dụng vào những bài toán cụ thể khác như: nhận diện ảnh, nhận diện chữ viết,..
- Nguyên lý hoạt động và cách điều khiển động cơ servo.
- Ứng dụng được nhiều công cụ hỗ trợ lập trình về xử lý hình ảnh, SQL, server trên Python.
- Thành thạo LaTeX để viết báo cáo một cách thành thạo.

2. Hạn chế

Bên cạnh đó, do thời gian cũng như kiến thức còn hạn hẹp nên cũng không tránh khỏi một số điều cần cải thiện như:

- Phần mềm quản lý chưa có tính bảo mật cao.
- Hệ thống xử lý nhận diện biển số còn chậm do truyền hình ảnh thông qua internet.
- Nhận diện biển số còn bị sai.

3. Phương hướng phát triển

Qua các điểm cần cải thiện nêu ra, đề tài có thể được phát triển hướng rộng hơn và hoàn thiện hơn như sau:

- Tích hợp, thay thế nhiều cảm biến để tăng tính ổn định và thông minh cho hệ thống.
- Thiết lập thêm tương tác phần mềm quản lý trên PC và ứng dụng di động.
- Lập trình thêm tính năng cho người dùng xem thông tin và quản lý thông tin cá nhân.
- Cải thiện tốc độ đường truyền cho hệ thống bằng cách xây dựng server trên raspberry.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[3]. Công ty TNHH MLAB (n.d.), *Raspberry Pi là gì ? Giới thiệu các ứng dụng của Raspberry Pi 3*, URL: https://raspberrypi.vn/san-pham/raspberry-pi-3-model-b.

[4]. Phạm Diệu Nguyên Đỗ Gia Huy (2023), *Mô Hình Nhận Dạng Vật Thể YOLOv8*, URL: https://hackmd.io/@58ZC49ZfS86wYX--LRGGOg/Viewperm.

[8]. MegaPacking (2021), *Trường Đại Học Công Nghệ Đồng Nai và nhu cầu kiểm soát xe ra vào*, URL: https://megaparking.vn/du-an/thi-cong-he-thong-kiem-soat-xe-ra-vao-truong-dai-hoc-cong-nghe-dong-nai/.

[9]. Nshop (n.d.), *Cảm biến vật cản hồng ngoại hoạt động như thế nào, có những loại cảm biến vật cản hồng ngoại nào ?*, URL: https://nshopvn.com/blog/cam-bien-vat-can-hong-ngoai-hoat-dong-nhu-the-nao-che-tao-bo-do-toc-do-dong-co-su-dung-cam-bien-vat-can-hong-ngoai/.

[11]. raspberrypi (n.d.), *Raspberry Pi 3 Model B+*, URL: https://raspberrypi.vn/san-pham/raspberry-pi-3-model-b.

[13]. TINO (n.d.), *Động Cơ Servo SG90S*, URL: https://www.dientutino.com/details.php?proid=189.

[14]. Trung tâm NCATGT - Học viện CSND (2017), "Tìm hiểu về công nghệ nhận dạng kí tự quang học (OCR - Optical Character Recognition) ứng dụng trong các phần mềm phát hiện và xử lý vi phạm luật giao thông thông qua hệ thống camera giám sát", URL: https://congan.travinh.gov.vn/ch26/351-Tim-hieu-ve-cong-nghe-nhan-dang-ki-tu-quang-hoc-OCR---Optical-Character-Recognition-ung-dung-trong-cac-phan-mem-phat-hien-va-xu-ly-vi-pham-luat-giao-thong-thong-qua-he-thong-camera-giam-sat.mhtml#:~:text=Optical%20Character%20Recognition%20(OCR)%20l%C3%A0,ch%E1%BB%89nh%20s%E1%BB%ADa%20trong%20m%C3%A1y%20t%C3%ADnh..

[15]. VBD (2022), *Canny – Phát hiện cạnh trong OpenCV: Hướng dẫn chi tiết từng bước*, URL: https://vinbigdata.com/kham-pha/canny-phat-hien-canh-trong-opencv-huong-dan-chi-tiet-tung-buoc.html.

Tiếng Anh

[1]. Ajitesh Kumar (2023), *Different Types of CNN Architectures*, URL: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/.

[2]. Amazon (2023), *What is OCR (Optical Character Recognition)?*, URL: https://aws.amazon.com/what-is/ocr/.

[5]. Ibm (2019), *What are convolutional neural networks?*, URL: https://www.ibm.com/topics/convolutional-neural-networks.

[6]. Jacksonliam (2020), *mjpg-streamer*, https://github.com/jacksonliam/mjpg-streamer.

[7]. Ph.D Jacob Murel (4/4/2024), *Train a YOLOv8 object detection model in Python*, URL: https://developer.ibm.com/tutorials/awb-train-yolo-object-detection-model-in-python/.

[10]. Pallets (2010), *Use Flask Framework*, https://flask.palletsprojects.com/en/3.0.x/installation/, Created using Sphinx 7.2.6.

[12]. Phani Ratan (2023), *What is the Convolutional Neural Network Architecture?*, URL: https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/.

[16]. zodinet (2019), *OCR and applications of OCR in life and technology*, URL: https://zodinet.com/en/ocr-and-applications-of-ocr-in-life-and-technology/.

PHỤ LỤC

1. Tạo cơ sở dữ liệu

Cơ sở dữ liệu của hệ thống gồm các bảng user, admin, manager, statistics với các thuộc tính và kiểu dữ liệu như code dưới đây:

```
class User(db.Model, UserMixin):
    __tablename__ = "user"
    id = db.Column(db.Integer, primary_key=True, autoincrement=
       True)
    name = db.Column(db.String(50), nullable=False)
    license_phate = db.Column(db.String(50), nullable=False)
    department = db.Column(db.String(50))

    def __init__(self, name, license_phate, department):
        self.name = name
        self.license_phate = license_phate
        self.department = department

class Admin(db.Model):
    __tablename__ = "admin"
    id = db.Column(db.Integer, primary_key = True, autoincrement=
       True)
    username = db.Column(db.String(50), nullable=False)
    password = db.Column(db.String(50), nullable=False)

    def __init__(self, username, password):
        self.username = username
        self.password = password

    def is_active(self):
        return True

    def get_id(self):
        return str(self.id)

class Manager(db.Model):
    __tablename__="manager"
    id = db.Column(db.Integer, primary_key = True, autoincrement=
       True)
    license_phate = db.Column(db.String(50), nullable=False)
    checkin=db.Column(db.DateTime, nullable=True)
    checkout=db.Column(db.DateTime, nullable=True)
    license_plate_image_in = db.Column(LargeBinary)
    license_plate_image_out = db.Column(LargeBinary)

```

```python
    def __init__(self, license_phate, checkin=None, checkout=None, license_plate_image_in=None, license_plate_image_out=None):
        self.license_phate = license_phate
        self.checkin = checkin
        self.checkout = checkout
        self.license_plate_image_in = license_plate_image_in
        self.license_plate_image_out = license_plate_image_out

class Statistics(db.Model):
    __tablename__ = "statistics"
    id = db.Column(db.Integer, primary_key=True, autoincrement=True)
    license_plate = db.Column(db.String(50), nullable=False)
    current_date = db.Column(db.Date, nullable=False)
    checkin_time = db.Column(db.String(8), nullable=True)
    checkout_time = db.Column(db.String(8), nullable=True)

    def __init__(self, license_plate, current_date, checkin=None, checkout=None):
        self.license_plate = license_plate
        self.current_date = current_date
        self.checkin_time = checkin
        self.checkout_time = checkout

with app.app_context():
    db.create_all()
```

2. Hàm xử lý ảnh để lấy ra biển số xe

Hàm xử lý biển số xe truyền vào một ảnh và xuất ra biển số xe sau khi xử lý.

```python
def process_image(model, image, source):
    mytext = ""
    results = model(image)
    boxes = results[0].boxes
    image_model = YOLO('best1.pt')

    if boxes.shape[0] > 0:
        x_min = int(boxes.xyxy[0][0])
        y_min = int(boxes.xyxy[0][1])
        x_max = int(boxes.xyxy[0][2])
        y_max = int(boxes.xyxy[0][3])
        cropped_image = image[y_min:y_max, x_min:x_max]
        results  = image_model.predict(cropped_image)
        if source == 'pi':
            cv2.imwrite("static/cropped_image_out.jpg", cropped_image)
        elif source == 'ras':
```

```
            cv2.imwrite("static/cropped_image_in.jpg",
               cropped_image)

        gray_image = cv2.cvtColor(cropped_image, cv2.
           COLOR_BGR2GRAY)
        image_results = image_model(cropped_image)
        print(image_results)
        boxes = results[0].boxes.xyxy.tolist()
        classes = results[0].boxes.cls.tolist()
        names = results[0].names
        confidences = results[0].boxes.conf.tolist()
        a = []
        for _ in range(15):
            hang = [0] * 3
            a.append(hang)

        row = 0
        for box, cls, conf in zip(boxes, classes, confidences):
            x1, y1, x2, y2 = box
            confidence = conf
            detected_class = cls
            name = names[int(cls)]
            a[row][0] = x1
            a[row][1] = y1
            a[row][2] = detected_class
            row += 1
        for i in range(row):
            print(f"{a[i][0]} x {a[i][1]} x {a[i][2]}")
        print(f"row: {row}")
        line1 = [15]*row
        line2 = [15]*row
        visit = [15]*row
        for i in range(row):
            line1[i] = 10000
            line2[i] = 10000
            visit[i] = 0

        check = 0
        for i in range(row):
            if(a[i][1]<30):
                if(visit[i] == 0):
                    check = i
                    for j in range(row):
                        if(visit[j] == 0):
                            visit[j] = 1
                            print(f"{a[check][1]} x {a[j][1]}")
                            if(a[j][1] - a[check][1] <= 20):
                                line1.append(j)
                            else:
```

```
                                line2.append(j)


        print(line1)
        print(line2)
        for i in range(len(line1)-1):
            for j in range(i + 1, len(line1)):
                if(line1[i] != 10000):
                    if a[line1[i]][0] > a[line1[j]][0]:
                        tem = line1[i]
                        line1[i] = line1[j]
                        line1[j] = tem

        for i in range(len(line2)-1):
            for j in range(i + 1, len(line2)):
                if(line2[i] != 10000):
                    if a[line2[i]][0] > a[line2[j]][0]:
                        tem = line2[i]
                        line2[i] = line2[j]
                        line2[j] = tem


        for i in range(len(line1)):
            if(line1[i] != 10000):
                mytext += names[a[line1[i]][2]]
        for i in range(len(line2)):
            if(line2[i] != 10000):
                mytext += names[a[line2[i]][2]]
        print(mytext)
        print(line1)
        print(line2)
    print(mytext)
    return mytext
```

3. Hàm gửi text sang Raspberry Pi

Sau khi đã xử lý biển số xe hệ thống sẽ gửi văn bản biển số sang raspberry pi để xử lý các phần tiếp theo

```
def send_text_to_raspi(filtered_string):
    if filtered_string:
        raspi_ip = 'ras'
        raspi_url = f'http://{raspi_ip}:5001/receive_text'
        data = {'filtered_string': filtered_string}
        response = requests.post(raspi_url, json=data)
        if response.status_code == 200:
            print('TC')
        else:
            print('Error')
```

4. Hàm xử lý cập nhật biển số xe lên cơ sở dữ liệu

Sau khi nhận biển số xe đã xử lý hệ thống sẽ kiểm tra xem là hình ảnh đầu vào hay đầu ra để cập nhật dữ liệu theo yêu cầu chức năng của hệ thống đã đặt ra.

```
def process_db_operations(filtered_string, in_or_out):
    user = User.query.filter_by(license_phate=filtered_string).
       first()
    if user:
        tz_VN = pytz.timezone('Asia/Ho_Chi_Minh')
        datetime_VN = datetime.now(tz_VN)
        current_date = datetime_VN.date()
        current_time = datetime_VN.strftime('%Hh%Mm%Ss')

        if in_or_out == 'in':
            result = Manager.query.filter(
                Manager.license_phate == filtered_string,
                (Manager.checkin.is_(None) | Manager.checkout.is_
                   (None))
            ).first()
            if result is None:
                with open("static/cropped_image_in.jpg", 'rb') as
                    f:
                    image_data = f.read()
                new_record = Manager(
                    license_phate=filtered_string,
                    checkin=datetime_VN,
                    license_plate_image_in=image_data
                )
                db.session.add(new_record)

            stat_record = Statistics.query.filter_by(
                license_plate=filtered_string,
                current_date=current_date
            ).first()
            if stat_record is None:
                new_stat_record = Statistics(
                    license_plate=filtered_string,
                    current_date=current_date,
                    checkin=current_time
                )
                db.session.add(new_stat_record)

        elif in_or_out == 'out':

            result = Manager.query.filter(
                Manager.license_phate == filtered_string,
                (Manager.checkout.is_(None))
            ).first()
            if result is not None:
```

```python

                with open("static/cropped_image_out.jpg", 'rb') as f:
                    image_data = f.read()
                result.checkout = datetime_VN
                result.license_plate_image_out = image_data

            stat_record = Statistics.query.filter_by(
                license_plate=filtered_string,
                current_date=current_date
            ).first()
            if stat_record is None:
                new_stat_record = Statistics(
                    license_plate=filtered_string,
                    current_date=current_date,
                    checkin=current_time
                )
                db.session.add(new_stat_record)
            else:
                stat_record.checkout_time = current_time

            db.session.commit()
            return user.id, user.name, user.department

        db.session.commit()
    return "", "", ""
```

5. Hàm xử lý và tải bảng theo yêu cầu của người dùng

Khi người dùng cần xem dữ liệu thống kê theo ngày, thống kê giờ ra và của xe và bảng người dùng hệ thống sẽ xử lý và đưa dữ liệu về web.

```python
def download_csv(table):
    try:
        logging.info(f'Request to download CSV for table: {table}')

        csv_data = StringIO()
        csv_writer = csv.writer(csv_data, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)

        if table == 'manage':
            data = Manager.query.all()
            csv_writer.writerow(['ID', 'License Plates', 'Checkin', 'Checkout'])
            for item in data:
                csv_writer.writerow([item.id, item.license_phate, item.checkin, item.checkout])

        elif table == 'user':
```

```
            data = User.query.all()
            csv_writer.writerow(['ID', 'Name', 'License Plates', 'Department'])
            for item in data:
                csv_writer.writerow([item.id, item.name, item.license_phate, item.department])
        elif table == 'in_and_out':
            data = db.session.query(
                Manager.id,
                User.name,
                User.department,
                Manager.license_phate,
                Manager.checkin,
                Manager.checkout,
            ).join(Manager, User.license_phate == Manager.license_phate).all()
            csv_writer.writerow(['ID', 'Name' , 'Department','License Plates', 'Checkin', 'Checkout'])
            for item in data:
                csv_writer.writerow([item.id,item.name , item.department, item.license_phate, item.checkin, item.checkout])

        response = make_response(csv_data.getvalue())
        response.headers['Content-Disposition'] = f'attachment; filename={table}.csv'
        response.headers['Content-Type'] = 'text/csv; charset=utf-8-sig'

        return response

    except Exception as e:

        logging.error(f'An error occurred: {str(e)}')
        return 'An error occurred while processing the request.', 500
```

6. Hàm xử lý và nhận tín hiệu sensor từ raspberry pi

Hệ thống nhận tín hiệu từ raspberry pi gửi sang và hiển thị số vị trí còn trống lên trên web.

```
sensor_data = {
    "ras": {"sensor1ras": None, "sensor2ras": None,"sensor3ras": None,"sensor4ras": None},
    "pi": {"sensor1ras": None, "sensor2ras": None,"sensor3ras": None,"sensor4ras": None},
}
def update_sensor():
    data = request.json
```

```
    raspberry_id = data['raspberry_id']
    sensor_id = data['sensor_id']
    value = data['value']

    sensor_data[raspberry_id][sensor_id] = value

    return jsonify({"status": "success", "data": sensor_data})
def get_sensor_data():
    total_obstacles = get_total_obstacles()
    return jsonify(total_obstacles)
def get_total_obstacles():
    total_obstacles = 0
    for pi_id in sensor_data:
        for sensor_id in sensor_data[pi_id]:
            if sensor_data[pi_id][sensor_id] is None or
               sensor_data[pi_id][sensor_id] == 1:
                total_obstacles += 1
    return total_obstacles
```

7. Hàm xử lý điều khiển server trên raspberry pi

Khi nhận được yêu cầu đóng hoặc mở barie từ server raspberry pi sẽ xử lý theo yêu cầu đóng hoặc mở.

```
def set_angle_open(angle):
    duty_cycle = (angle / 18) + 2
    GPIO.output(servo_pin_open, True)
    pwm_open.ChangeDutyCycle(duty_cycle)
    time.sleep(1)
    GPIO.output(servo_pin_open, False)
    pwm_open.ChangeDutyCycle(0)

def set_angle_close(angle):
    duty_cycle = (angle / 18) + 2
    GPIO.output(servo_pin_close, True)
    pwm_close.ChangeDutyCycle(duty_cycle)
    time.sleep(1)
    GPIO.output(servo_pin_close, False)
    pwm_close.ChangeDutyCycle(0)

def return_to_initial_position():
    set_angle_open(0)
    set_angle_close(0)

@app_raspi.route('/open', methods=['POST'])
def open_servo():
    set_angle_open(90)
    time.sleep(5)
    set_angle_open(0)
    return {'status': 'success'}
```

```

@app_raspi.route('/close', methods=['POST'])
def close_servo():
    set_angle_close(90)
    time.sleep(5)
    set_angle_close(0)
    return {'status': 'success'}
```

8. Hàm xử lý và hiển thị dữ liệu lên LCD

Hệ thống liên tục nhận dữ liệu và cập nhật lại lên trên màn hình LCD.

```
rows = 4
lcd = liquidcrystal_i2c.LiquidCrystal_I2C(0x26, 1, numlines=rows)
lcd.printline(0, 'Xin Chao  ACTVN')
@app_raspi.route('/receive_text', methods=['POST'])
def receive_filtered_string():
    data = request.json
    filtered_string = data.get('filtered_string')
    if filtered_string:
        print("Filtered string received:", filtered_string)
        lcd.printline(0, '                    ')
        lcd.printline(1, '                    ')
        lcd.printline(0, 'Bien So Xe: ')

        lcd.printline(1, filtered_string)
        return {'status': 'success'}
    else:
        return {'status': 'error', 'message': 'Không có ữd ệliu
           filtered_string'}
```

9. Hàm nhận dữ liệu từ senser và gửi lên server

Hệ thống liên tục cập nhật giá trị của cảm biến vật cản và trả về giá trị cho server quản lý

```
while True:
        sensor1_state = GPIO.input(SENSOR_1_PIN)
        sensor2_state = GPIO.input(SENSOR_2_PIN)
        sensor3_state = GPIO.input(SENSOR_3_PIN)
        sensor4_state = GPIO.input(SENSOR_4_PIN)
        # ửGi ữd ệliu ảcm ếbin 1
        data1 = {
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor1ras",
            "value": sensor1_state
        }
        requests.post(server_url, json=data1)

        # ửGi ữd ệliu ảcm ếbin 2
        data2 = {
```

```
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor2ras",
            "value": sensor2_state
        }
        requests.post(server_url, json=data2)
        data3 = {
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor3ras",
            "value": sensor3_state
        }
        requests.post(server_url, json=data3)
        data4 = {
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor4ras",
            "value": sensor4_state
        }
        requests.post(server_url, json=data4)

        time.sleep(1)
```

10. Gửi dữ liệu cảm biến từ raspberry lên server

Trong hệ thống tại raspberry dữ liệu sẽ được gửi từ các chân cảm biến vật cản hồng ngoại lên server như sau:

```
import RPi.GPIO as GPIO
import time
import requests

# ấCu hình GPIO cho Raspberry Pi
SENSOR_1_PIN = 23
SENSOR_2_PIN = 24
SENSOR_3_PIN = 20
SENSOR_4_PIN = 21
GPIO.setmode(GPIO.BCM)
GPIO.setup(SENSOR_1_PIN, GPIO.IN)
GPIO.setup(SENSOR_2_PIN, GPIO.IN)
GPIO.setup(SENSOR_3_PIN, GPIO.IN)
GPIO.setup(SENSOR_4_PIN, GPIO.IN)
server_url = "http://desktop:5000/update_sensor"  # Thay
    SERVER_IP ằbng địa ỉch IP ủca server trung tâm

raspberry_id = "pi"  # ID ủca Raspberry Pi này

try:
    while True:
        sensor1_state = GPIO.input(SENSOR_1_PIN)
        sensor2_state = GPIO.input(SENSOR_2_PIN)
        sensor3_state = GPIO.input(SENSOR_3_PIN)
        sensor4_state = GPIO.input(SENSOR_4_PIN)
```

```
        data1 = {
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor1ras",
            "value": sensor1_state
        }
        requests.post(server_url, json=data1)

        data2 = {
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor2ras",
            "value": sensor2_state
        }
        requests.post(server_url, json=data2)
        data3 = {
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor3ras",
            "value": sensor3_state
        }
        requests.post(server_url, json=data3)
        data4 = {
            "raspberry_id": raspberry_id,
            "sensor_id": "sensor4ras",
            "value": sensor4_state
        }
        requests.post(server_url, json=data4)

        time.sleep(1)
except KeyboardInterrupt:
    print("Measurement stopped by user")
    GPIO.cleanup()
```

11. Điều khiển servo và hiển thị dữ liệu lên màn hình LCD

Tại raspberry hệ thống điều khiển raspberry sẽ điều khiển servo và hiển thị biển số xe màn hình LCD như dưới đây:

```
from flask import Flask, request
import RPi.GPIO as GPIO
import time
import liquidcrystal_i2c

app_raspi = Flask(__name__)
servo_pin_open = 24
servo_pin_close = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(servo_pin_open, GPIO.OUT)
GPIO.setup(servo_pin_close, GPIO.OUT)
pwm_open = GPIO.PWM(servo_pin_open, 50)
pwm_open.start(0)
pwm_close = GPIO.PWM(servo_pin_close, 50)
```

```
pwm_close.start(0)
rows = 4
lcd = liquidcrystal_i2c.LiquidCrystal_I2C(0x26, 1, numlines=rows)
lcd.printline(0, 'Xin Chao  ACTVN')

def set_angle_open(angle):
    duty_cycle = (angle / 18) + 2
    GPIO.output(servo_pin_open, True)
    pwm_open.ChangeDutyCycle(duty_cycle)
    time.sleep(1)
    GPIO.output(servo_pin_open, False)
    pwm_open.ChangeDutyCycle(0)

def set_angle_close(angle):
    duty_cycle = (angle / 18) + 2
    GPIO.output(servo_pin_close, True)
    pwm_close.ChangeDutyCycle(duty_cycle)
    time.sleep(1)
    GPIO.output(servo_pin_close, False)
    pwm_close.ChangeDutyCycle(0)

def return_to_initial_position():
    set_angle_open(0)
    set_angle_close(0)

@app_raspi.route('/open', methods=['POST'])
def open_servo():
    set_angle_open(90)
    time.sleep(5)
    set_angle_open(0)
    return {'status': 'success'}

@app_raspi.route('/close', methods=['POST'])
def close_servo():
    set_angle_close(90)
    time.sleep(5)
    set_angle_close(0)
    return {'status': 'success'}

@app_raspi.route('/receive_text', methods=['POST'])
def receive_filtered_string():
    data = request.json
    filtered_string = data.get('filtered_string')

    if filtered_string:
        print("Filtered string received:", filtered_string)
        lcd.printline(0, '                    ')
        lcd.printline(1, '                    ')
        lcd.printline(0, 'Bien So Xe: ')
```

```
        set_angle_open(90)
        time.sleep(5)
        set_angle_open(0)
        lcd.printline(1, filtered_string)
        return {'status': 'success'}
    else:
        return {'status': 'error', 'message': 'Không có ữd ệliu filtered_string'}

if __name__ == '__main__':
    try:
        app_raspi.run(host='0.0.0.0', port=5001, debug=True)
    finally:
        pwm_open.stop()
        pwm_close.stop()
        GPIO.cleanup()
```