

RESEARCH ON DESIGN OF A MODEL OF SELF-DRIVING VEHICLE RECOGNIZING TRAFFIC SIGNS

Hoang Van Thuc

TNU - University of Information and Communication Technology

ARTICLE INFO	ABSTRACT
Received: 08/01/2025	In the era of Industry 4.0, the development of image processing technology, artificial intelligence and automation systems has opened a new era for the automotive industry, especially in the field of autonomous vehicles. Autonomous vehicles are not only a means of transportation, but also a symbol of technological progress, promising to bring many benefits such as reducing traffic accidents, saving energy and optimizing urban traffic. However, for autonomous vehicles to operate safely and effectively, they must be able to recognize and respond to surrounding environmental factors, in which traffic signs play a leading role. Based on this idea, the article presents research results on autonomous vehicle systems that apply image processing technology and artificial intelligence to improve the accuracy, recognition speed and response ability of autonomous vehicles. Experiments show that autonomous vehicles ensure the proposed quality criteria thanks to computer vision. Computer Vision has demonstrated outstanding performance in image recognition with the Cascade Adaboost algorithm in real time with a high accuracy rate. The results demonstrate the effectiveness of the research direction on automatic navigation of autonomous vehicles in different moving environments.
Revised: 28/3/2025	
Published: 03/4/2025	
KEYWORDS	
Autonomous vehicles	
Computer vision	
Image processing	
Navigation	
Autonomous transportation	
Mobile robots	

NGHIÊN CỨU THIẾT KẾ MÔ HÌNH XE TỰ HÀNH NHẬN DIỆN BIỂN BÁO GIAO THÔNG

Hoàng Văn Thục

Trường Đại học Công nghệ thông tin và Truyền thông - ĐH Thái Nguyên

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 08/01/2025	Trong thời đại công nghệ 4.0, sự phát triển của các công nghệ xử lý ảnh, trí tuệ nhân tạo và các hệ thống tự động hóa đã mở ra một kỷ nguyên mới cho ngành công nghiệp ô tô, đặc biệt là trong lĩnh vực xe tự hành. Xe tự hành không chỉ là phương tiện di chuyển, mà còn là biểu tượng của sự tiến bộ công nghệ, hứa hẹn mang lại nhiều lợi ích như giảm thiểu tai nạn giao thông, tiết kiệm năng lượng và tối ưu hóa giao thông đô thị. Tuy nhiên, để xe tự hành có thể hoạt động một cách an toàn và hiệu quả, chúng phải có khả năng nhận diện và phản ứng với các yếu tố môi trường xung quanh, trong đó biển báo giao thông đóng vai trò quan trọng hàng đầu. Xuất phát từ ý tưởng này, bài báo trình bày kết quả nghiên cứu về hệ thống xe tự hành ứng dụng các công nghệ xử lý ảnh, trí tuệ nhân tạo để nâng cao độ chính xác, tốc độ nhận diện, khả năng phản ứng của xe tự hành. Thử nghiệm cho thấy xe tự hành đảm bảo các chỉ tiêu chất lượng đề ra nhờ thị giác máy tính Computer Vision đã chứng minh hiệu quả vượt trội trong việc nhận diện hình ảnh với thuật toán Cascade Adaboost ở thời gian thực với tỷ lệ chính xác cao. Kết quả thể hiện sự hiệu quả của hướng nghiên cứu về sự điều hướng tự động của xe tự hành trong các môi trường di chuyển khác nhau.
Ngày hoàn thiện: 28/3/2025	
Ngày đăng: 03/4/2025	
TỪ KHÓA	
Xe tự hành	
Thị giác máy tính	
Xử lý ảnh	
Điều hướng	
Xe tự hành giao thông	
Robot di động	

DOI: <https://doi.org/10.34238/tnu-jst.11837>

Email: hvthuc@ictu.edu.vn

<http://jst.tnu.edu.vn>

19

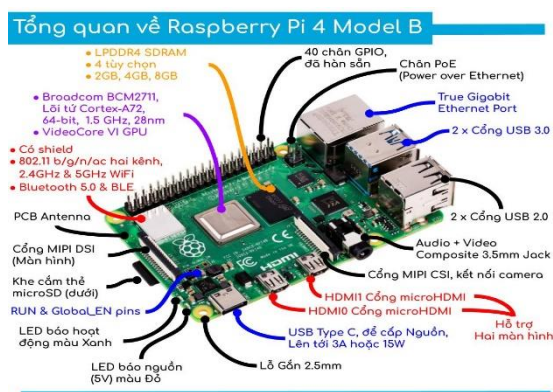
Email: jst@tnu.edu.vn

1. Giới thiệu

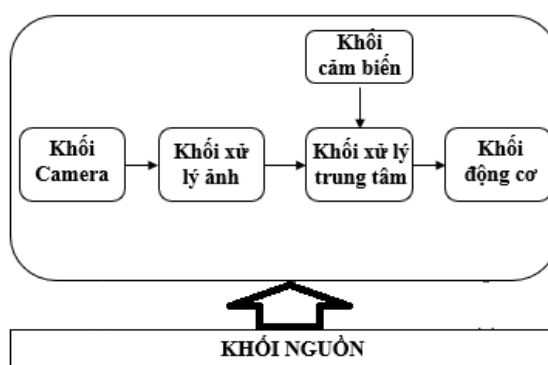
Trong các năm gần đây, các công nghệ mới dần thay đổi các công nghệ cũ. Trong đó phải kể đến các công nghệ xử lý ảnh, trí tuệ nhân tạo để nhận diện biển báo giao thông đã đạt được nhiều tiến bộ đáng kể, tuy nhiên vẫn còn nhiều vấn đề cần được giải quyết [1], [2]. Các hệ thống hiện tại đôi khi gặp khó khăn khi phải nhận diện biển báo bị mờ, biến dạng, hoặc bị che khuất một phần. Hơn nữa, tốc độ và độ chính xác của việc nhận diện trong thời gian thực cũng là những yếu tố quan trọng ảnh hưởng đến hiệu quả và độ an toàn của hệ thống xe tự hành. Các nghiên cứu về xe tự hành nhận dạng biển báo tập trung vào việc phát triển các hệ thống sử dụng các phương pháp học máy và trí tuệ nhân tạo để giúp xe tự động nhận diện và phản ứng với các biển báo giao thông [3], [4]. Các kỹ thuật phổ biến bao gồm học sâu (deep learning), mạng nơ-ron tích chập (CNN), và các phương pháp xử lý ảnh để nhận diện biển báo trong các điều kiện môi trường khác nhau. Các nghiên cứu này cũng hướng tới việc cải thiện độ chính xác và hiệu suất trong việc nhận diện biển báo, đặc biệt là trong các tình huống thay đổi như ánh sáng yếu hay thời tiết xấu. Một số nghiên cứu còn tích hợp các cảm biến như LIDAR và radar để hỗ trợ nhận dạng, nâng cao khả năng phản ứng an toàn của xe. Chúng đóng vai trò quan trọng trong việc đảm bảo an toàn và hiệu quả cho các phương tiện tự hành trong giao thông thực tế [5]-[7].

Bài báo này tập trung nghiên cứu thiết kế mô hình xe tự hành nhận diện biển báo giao thông. Với mô hình nhỏ gọn, giá thành rẻ và hợp lý, độ chính xác tin cậy, giao diện trực quan dễ quan sát và có thể ứng dụng rộng rãi trong thực tế. Cụ thể, hệ thống sẽ được xây dựng dựa trên các công nghệ xử lý ảnh, trí tuệ nhân tạo và ứng dụng thuật toán Adaboost với đặc trưng Haar-like để xây dựng các bộ phân lớp, nhận diện và phân loại cho từng loại biển báo trong thời gian thực. Đây là một bước tiến quan trọng hướng đến việc hoàn thiện hệ thống xe tự hành an toàn và hiệu quả hơn, đóng góp vào sự phát triển của giao thông thông minh trong tương lai [8], [9]. Kết quả thực nghiệm và khảo sát hệ thống trên nhiều môi trường di chuyển khác nhau tương đối phù hợp. Sai số về độ trễ có thể từ nhiều nguyên nhân khác nhau. Nếu được điều chỉnh tốt, ý tưởng này kết hợp với mô hình tự động hóa trong thực tế với quy mô lớn sẽ trở thành một hệ thống lớn đáp ứng nhu cầu điều khiển, quản lý điều khiển các thiết bị một cách hiện đại sẽ nâng cao đời sống tiện ích trong giao thông vận tải và công nghiệp. Vì vậy, hệ thống không chỉ góp phần vào việc phát triển công nghệ xe tự hành mà còn đóng góp vào sự an toàn và hiệu quả, lợi ích lớn của hệ thống giao thông tương lai ở các nước đang phát triển như Việt Nam [10], [11].

2. Phương pháp nghiên cứu và sơ đồ khối thiết kế hệ thống



Hình 1. Raspberry pi 4B



Hình 2. Sơ đồ khối thiết kế hệ thống

Raspberry pi 4B được mô tả trên Hình 1 là một sản phẩm mới nhất trong họ máy tính nhúng Raspberry Pi, nó được ra đời vào cuối tháng 6/2019. Cấu hình của Raspberry Pi 4B nổi bật với vi mạch Broadcom BCM2711, Cortex-A72 (ARM v8) 4 nhân 64 bit với tốc độ xử lý là 1,5GHz, đây là dòng vi mạch có tốc độ xử lý nhanh nhất từ trước đến nay của họ máy tính nhúng này. Với

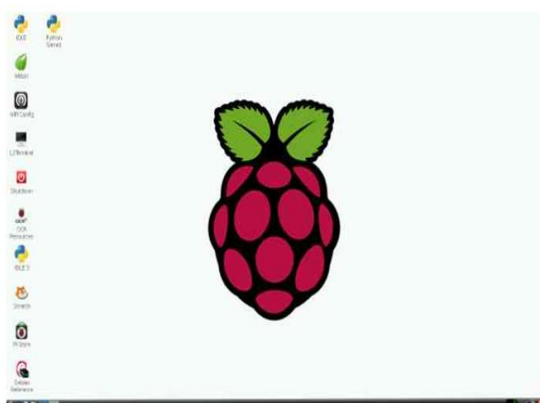
sự gắn kết của Raspberry pi 4B với các hệ thống điều khiển, cảm biến, thiết bị trong thực tế với mô hình lớn sẽ trở thành một hệ thống lớn đáp ứng nhu cầu điều khiển, quản lý tất cả các thiết bị một cách tối ưu nhất, nâng cao đời sống với tiện ích hiện đại trong công nghiệp tự động hóa. Hệ thống sẽ thu thập và xử lý dữ liệu với Raspberry pi 4B [1], [2].

2.1. Thiết kế hệ thống

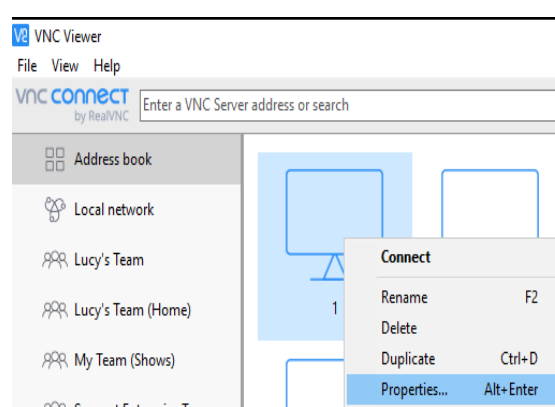
Với mục đích xây dựng một mô hình hệ thống theo cấu trúc được thể hiện qua sơ đồ khối ở Hình 2 bao gồm 6 khối chính: Khối nguồn, khối camera, khối xử lý trung tâm, khối động cơ, khối cảm biến. Mỗi khối này đảm nhận một chức năng cụ thể để đảm bảo hoạt động hiệu quả của toàn bộ hệ thống. Trong đó, khối xử lý trung tâm sử dụng Module Raspberry pi 4B đóng vai trò quan trọng trong việc truyền nhận tín hiệu giữa các khối, điều khiển và quản lý dữ liệu đảm bảo sự hoạt động đồng bộ của giữa phần tử hệ thống [3].

2.2. Hệ điều hành và phần mềm của hệ thống

Hệ điều hành Raspbian được mô tả trên Hình 3 là một hệ điều hành dựa trên Debian được tối ưu hóa cho phần cứng của Raspberry Pi, Raspbian đi kèm với tất cả các chương trình và tiện ích cơ bản bạn mong đợi từ một hệ điều hành có mục đích chung. Được hỗ trợ chính thức bởi nền tảng Raspberry, hệ điều hành này nổi tiếng với hiệu năng nhanh và hơn 35.000 gói.



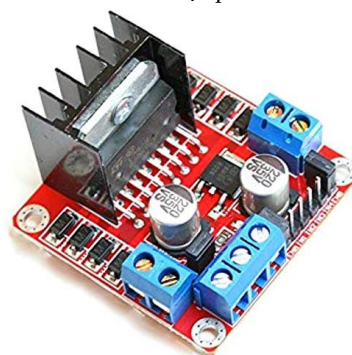
Hình 3. Hệ điều hành Raspbian



Hình 4. Giao diện phần mềm VNC



Hình 5. Module Arduino Uno R3



Hình 6. Module L298N

Giao diện phần mềm VNC (Virtual Network Computing) trong Hình 4 là hệ thống phần mềm và giao thức cho phép điều khiển và quản lý máy tính từ xa qua mạng. VNC gồm hai thành phần: máy chủ (server) chạy phần mềm VNC server và máy khách (client) chạy phần mềm VNC client. Dữ liệu màn hình cùng các sự kiện nhập liệu (bàn phím, chuột) được mã hóa và truyền qua mạng. VNC thường được sử dụng cho quản lý, hỗ trợ từ xa, mô phỏng và kiểm thử.

Module Arduino Uno R3

Module Arduino Uno R3 được mô tả trên Hình 5 sử dụng vi điều khiển Atmega328 để vận hành. Nó sở hữu 14 chân I/O (bao gồm 6 chân có khả năng xuất tín hiệu PWM), 6 đầu ra Analog, một cổng USB, một jack nguồn, ICSP và một nút Reset. Bo mạch này tích hợp đầy đủ các thành phần cần thiết để hỗ trợ vi điều khiển, giúp kết nối với máy tính qua cáp USB hoặc sử dụng nguồn điện từ bộ chuyển đổi AC sang DC hay pin để hoạt động [4], [5].

Module L298N

Module L298N là module điều khiển động cơ trong các xe DC và động cơ bước. Module có một IC điều khiển động cơ L298 và một bộ điều chỉnh điện áp 5V 78M05. Module L298N có thể điều khiển tối đa 4 động cơ DC hoặc 2 động cơ DC với khả năng điều khiển hướng và tốc độ. Hình ảnh thật của Module L298N được mô tả trên Hình 6.

Raspberry Pi Camera Module



Hình 7. Module Raspberry Pi Camera



Hình 8. Mạch hạ áp LM2596

Module Raspberry Pi Camera, như được trình bày trong Hình 7, tích hợp một camera 5 megapixel với khả năng nhạy sáng cao, cho phép chụp ảnh sắc nét trong nhiều điều kiện ánh sáng, cả trong nhà lẫn ngoài trời. Một điểm nổi bật của camera này là khả năng chụp ảnh độ phân giải cao trong khi quay video. Khi sử dụng Raspberry Pi Camera, người dùng không cần phải sử dụng cổng USB cho camera, vì nó được kết nối trực tiếp vào socket CSI. Điều này giúp giảm tải cho băng thông của chip xử lý USB trên mạch Raspberry. Ngoài ra, chiều dài cáp nối camera được tính toán hợp lý để đảm bảo tốc độ truyền dữ liệu hình ảnh từ module đến RPi mà không bị gián đoạn.

Mạch hạ áp

Trên Hình 8 là hình ảnh mạch hạ áp LM2596 với kích thước nhỏ gọn có khả năng giảm áp từ 30VDC xuống 1,5VDC mà vẫn đạt hiệu suất cao (92%), thích hợp cho các ứng dụng chia nguồn, hạ áp, cấp cho các thiết bị như camera, robot, xe tự hành.



Hình 9. Nguồn Adapter



Hình 10. Khung xe [3]

Nguồn Adapter

Nguồn Adapter trên Hình 9 là một thiết bị điện tử dùng để chuyển đổi dòng điện có điện áp cao (220V) xuống dòng điện có điện áp thấp hơn (24V, 19V, 12V, 5V). Adapter thường có tính năng bảo vệ quá dòng, quá áp và ngắn mạch, giúp đảm bảo an toàn cho thiết bị và người sử dụng. Một số loại adapter còn có khả năng điều chỉnh điện áp, đáp ứng nhu cầu sử dụng đa dạng.

Khung xe

Khung gầm ô tô thông minh trên Hình 10 được trang bị bộ dụng cụ với bộ mã hóa tốc độ dành cho Arduino 51 M26. Bộ khung có bánh xe với đường kính khoảng 6 cm, giúp xe di chuyển linh hoạt và mượt mà. Tấm đáy của khung gầm có độ dày khoảng 3 mm, đảm bảo độ chắc chắn và ổn định khi hoạt động. Kích thước tổng thể của khung là chiều dài khoảng 25 cm và chiều rộng khoảng 14,8 cm, phù hợp để lắp đặt các linh kiện và mạch điện tử phục vụ cho các ứng dụng điều khiển và robot tự hành. Xe được trang bị động cơ DC mạnh mẽ, cho phép điều khiển chính xác hướng di chuyển và tốc độ, tạo điều kiện cho các nhiệm vụ tự động hóa hoặc thí nghiệm nghiên cứu về robot tự hành.



Hình 11. Cảm biến siêu âm HC - SR04

Cảm biến siêu âm HC – SR04

Cảm biến siêu âm được thể hiện trong Hình 11, là một công cụ phổ biến để đo khoảng cách. Cảm biến HC-SR04 hoạt động bằng cách sử dụng sóng siêu âm và có khả năng đo khoảng cách từ 2 đến 300 cm, với độ chính xác chủ yếu phụ thuộc vào cách lập trình. Các chân kết nối của cảm biến gồm VCC (5V), trig (chân điều khiển phát sóng), echo (chân nhận tín hiệu phản hồi) và GND (chân nối đất).

Màn hình 7 inch

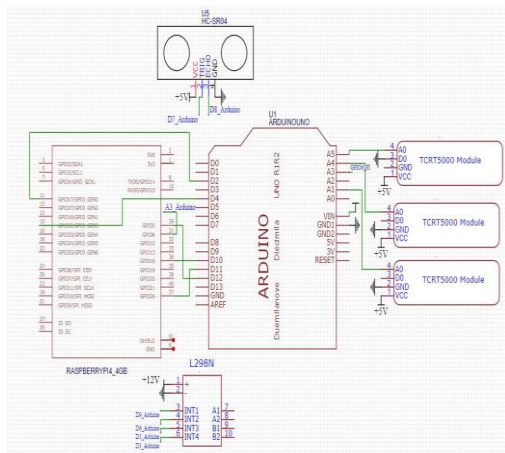
Màn hình 7 inch được mô tả trên Hình 12 dùng để kết nối với Raspberry Pi 4B giúp hiển thị giao diện và thông tin trực tiếp từ Raspberry Pi. Màn hình này thường sử dụng kết nối GPIO hoặc HDMI, cho phép dễ dàng tích hợp vào các dự án DIY hoặc hệ thống nhúng. Ngoài ra, màn hình còn có thể hỗ trợ cảm ứng, giúp tăng tính tương tác cho các dự án. Sự kết hợp này rất phổ biến trong các dự án robot, xe tự hành, màn hình điều khiển hoặc các ứng dụng Internet of Things (IoT) [6], [7].



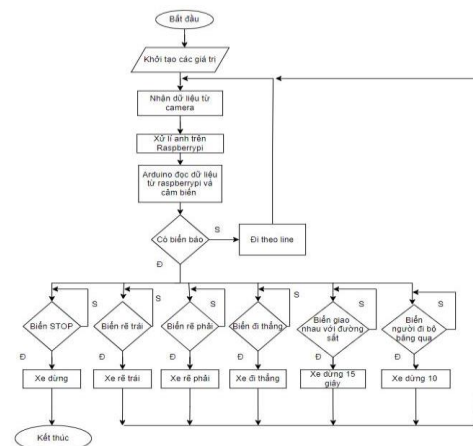
Hình 12. Màn hình 7 inch

2.3. Sơ đồ nguyên lý và lưu đồ thuật toán của hệ thống

Từ những phân tích và thiết kế trên, tác giả đã hoàn thiện sơ đồ nguyên lý toàn mạch được mô tả dưới Hình 13. Hệ thống được xây dựng hoàn chỉnh dựa trên nền tảng cấu trúc và chức năng của máy tính nhúng Raspberry pi 4B. Hệ thống hoạt động bằng cách sử dụng camera và Raspberry Pi để nhận diện biển báo giao thông và gửi lệnh điều khiển đến Arduino. Trong trường hợp không có biển báo, 3 cảm biến hồng ngoại giúp dò line, và cảm biến khoảng cách đảm bảo xe dừng lại khi phát hiện vật cản gần. Arduino điều chỉnh động cơ thông qua L298 để xe di chuyển đúng hướng hoặc dừng lại khi cần [8].



Hình 13. Sơ đồ nguyên lý toàn mạch của hệ thống



Hình 14. Lưu đồ thuật toán của hệ thống

Lưu đồ thuật toán của hệ thống được thể hiện trên Hình 14: Bắt đầu chương trình khởi tạo các giá trị, cấu hình các biến cần thiết và các thiết lập ban đầu: Khởi tạo kết nối giữa Raspberry Pi và Arduino qua GPIO. Khởi tạo camera để bắt đầu thu thập hình ảnh. Khởi tạo các cảm biến hồng ngoại trên xe. Raspberry Pi thu thập hình ảnh các biển báo từ camera theo thời gian thực. Raspberry Pi thực hiện xử lý ảnh để nhận diện các biển báo giao thông trong hình ảnh. Sử dụng thuật toán Cascade Adaboost để xác định, nhận diện các loại biển báo (Stop, rẽ trái, rẽ phải, giảm tốc độ). Khi Raspberry Pi gửi thông tin về biển báo giao thông (nếu có) đến Arduino qua cổng GPIO, Arduino sẽ đồng thời đọc dữ liệu từ các cảm biến hồng ngoại và kiểm tra xem có biển báo nào được nhận hay không, ví dụ như biển báo rẽ trái, rẽ phải, giao nhau với đường sắt, người đi bộ, hay đi thẳng. Nếu có biển báo, Arduino sẽ điều khiển xe thực hiện hành động tương ứng. Nếu không có biển báo, Arduino sẽ điều khiển xe di chuyển theo line dựa trên dữ liệu từ các cảm biến hồng ngoại. Khi gặp biển báo STOP, xe sẽ dừng lại và chương trình sẽ kết thúc.

2.4. Phân loại biển báo ứng dụng Cascade Adaboost

Quá trình nhận dạng biển báo giao thông bao gồm hai bước cơ bản: phát hiện và nhận dạng biển báo. Đầu tiên, bài toán phát hiện tập trung vào việc xác định vị trí của biển báo trong ảnh hoặc video. Sau đó, bài toán nhận dạng sẽ phân loại các biển báo theo từng loại cụ thể. Trong nghiên cứu này, nhóm tác giả đề xuất sử dụng thuật toán Cascaded Adaboost kết hợp với đặc trưng Haar-like [9] để xây dựng các bộ phân lớp hiệu quả cho từng loại biển báo. Thuật toán Adaboost với cấu trúc cascaded giúp cải thiện độ chính xác và tốc độ nhận dạng, trong khi đặc trưng Haar-like hỗ trợ nhận diện các đặc điểm quan trọng của biển báo trong các điều kiện môi trường khác nhau [10].

Trong nghiên cứu này, nhóm tác giả thực hiện việc nhận dạng bốn loại biển báo giao thông được thể hiện ở Hình 15 bao gồm: biển hạn chế tốc độ, biển rẽ trái, biển rẽ phải và biển dừng. Mỗi loại biển báo sẽ được xây dựng một bộ phân lớp riêng biệt, giúp hệ thống có khả năng nhận diện chính xác từng loại biển trong môi trường thực tế. Quá trình này yêu cầu việc thiết kế và huấn luyện các bộ phân lớp sao cho có thể phân biệt rõ ràng các đặc điểm và hình dáng đặc trưng của mỗi biển báo, từ đó nâng cao hiệu quả nhận dạng và ứng dụng trong các hệ thống giao thông thông minh.

Dữ liệu để huấn luyện bao gồm 3050 ảnh nền không chứa biển báo, còn lại mỗi loại sử dụng 1230 ảnh. Các thông số được chọn cho bộ huấn luyện Cascade Adaboost như sau: false positive rate, detection rate và cascade false positive có giá trị lần lượt là 0,5; 0,5 và 0,005. Các bộ phân lớp sau đó được kiểm thử với 220 ảnh trong bộ test chuẩn của cuộc thi [11] và kết quả được thể hiện trên Bảng 1.



Hình 15. Biển báo cần nhận diện

Bảng 1. Kết quả huấn luyện phân lớp biển báo bằng thuật toán Cascade Adaboost

	Biển báo rẽ trái	Biển báo rẽ phải	Biển dừng	Biển báo hạn chế 50 km/h
Số lượng được kiểm tra	55	55	55	55
Số lượng được phát hiện	60	59	66	65
Số lượng nhận dạng đúng	52	51	49	50
Độ chính xác	94,5%	92,7%	89,1%	90,1%

2.5. Xây dựng ma trận nhầm lẫn từ bảng kết quả huấn luyện phân lớp biển báo bằng thuật toán Cascade Adaboost

Để xây dựng ma trận nhầm lẫn cho mỗi loại biển báo, chúng ta cần xác định số lượng dự đoán đúng (True Positive - TP) và dự đoán sai (False Positive - FP) đối với mỗi loại biển báo. Ma trận nhầm lẫn sẽ có các thành phần bao gồm:

True Positive (TP): Số lượng đối tượng được nhận dạng đúng là biển báo tương ứng.

False Positive (FP): Số lượng đối tượng bị nhận diện sai là biển báo của lớp khác.

True Negative (TN): Số lượng đối tượng bị nhận diện đúng là không phải biển báo của lớp đó.

False Negative (FN): Số lượng đối tượng bị bỏ qua không nhận diện là biển báo của lớp đó.

Dựa vào thông tin này, tác giả bài báo tính toán được các giá trị sau:

Số lượng nhận dạng sai (False Positive): $FP = \text{Số lượng phát hiện} - \text{Số lượng nhận dạng đúng}$.

Số lượng không nhận dạng (False Negative): $FN = \text{Số lượng kiểm tra} - \text{Số lượng nhận dạng đúng}$.

True Negative (TN): Số lượng đối tượng mà không phải biển báo của lớp đó sẽ được tính là đối tượng không bị nhận dạng sai cho lớp này, nhưng trong trường hợp này chúng ta không có đủ dữ liệu để tính toán số lượng TN cho mỗi lớp riêng biệt, nên TN sẽ được hiểu là tổng của các đối tượng không thuộc lớp đó và không bị nhận dạng sai.

Bảng 2. Bảng ma trận nhầm lẫn cho mỗi loại biển báo

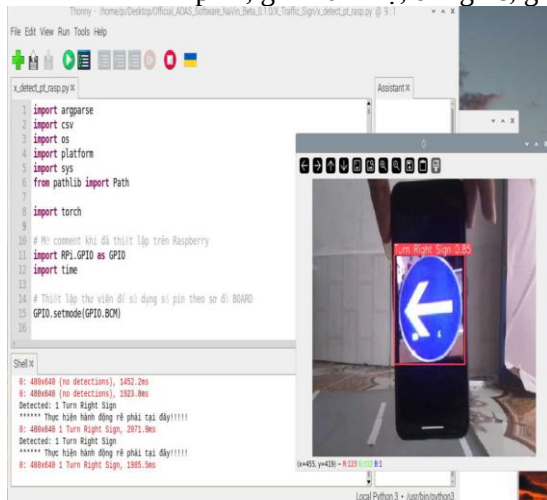
Lớp biển báo	Biển báo rẽ trái	Biển báo rẽ phải	Biển dừng	Biển báo hạn chế 50 km/h
Số lượng kiểm tra (Ground truth)	55	55	55	55
Số lượng phát hiện (Predicted)	60	59	66	65
Số lượng nhận dạng đúng (TP)	52	51	49	50
False Positive (FP)	$60 - 52 = 8$	$59 - 51 = 8$	$66 - 49 = 17$	$65 - 50 = 15$
False Negative (FN)	$55 - 52 = 3$	$55 - 51 = 4$	$55 - 49 = 6$	$55 - 50 = 5$

Dựa vào Bảng 1 về kết quả huấn luyện phân lớp biển báo bằng thuật toán Cascade Adaboost và Bảng 2 về ma trận nhầm lẫn cho mỗi loại biển báo chúng ta thấy rằng các biển báo rẽ trái và rẽ phải có tỷ lệ nhận dạng đúng khá cao, gần 94,5% và 92,7%, trong khi các biển báo dừng và hạn chế tốc độ có tỷ lệ thấp hơn (89,1% và 90,1%). Điều này có thể chỉ ra rằng hệ thống Cascade Adaboost hoạt động tốt đối với một số loại biển báo nhưng chưa hiệu quả với những

biển báo phức tạp hoặc có sự tương đồng với các biển báo khác. Có thể bằng cách tăng cường dữ liệu huấn luyện hoặc sử dụng các kỹ thuật tiền xử lý tốt hơn để giảm thiểu các nhầm lẫn, đặc biệt là đối với biển báo dừng và hạn chế tốc độ. Nói chung hệ thống đã hoạt động khá tốt, nhưng vẫn còn những điểm cần cải thiện để đạt được độ chính xác cao hơn và giảm thiểu các lỗi nhầm lẫn.

3. Kết quả mô phỏng và thực nghiệm hệ thống

Hình 16 là kết quả mô phỏng và Hình 17 là mô hình hoàn chỉnh của hệ thống xe tự hành ứng dụng các công nghệ xử lý ảnh, trí tuệ nhân tạo để nhận diện biển báo giao thông. Hệ thống xe tự hành sau khi cấp nguồn xe sẽ hoạt động. Xe sẽ đi theo đường line, khi đến đoạn gặp biển báo, camera sẽ nhận diện biển báo. Sau đó Raspberry Pi 4B sẽ thực hiện xử lý ảnh và gửi tín hiệu về arduino để điều khiển hoạt động của xe. Ví dụ, nếu camera nhận diện được biển báo rẽ trái, xe sẽ thực hiện rẽ trái. Hoặc khi xe gặp vật cản trên đường line xe sẽ dừng lại. Tương tự như vậy với các biển báo: rẽ phải, giảm tốc độ, dừng xe, giao nhau đường sắt, biển người đi bộ...



Hình 16. Kết quả thực nghiệm



Hình 17. Mô hình hệ thống hoàn chỉnh

Nguyên lý hoạt động của hệ thống có thể được hiểu tóm lược như sau: Sau khi cấp nguồn cho xe hoạt động, xe sẽ di chuyển theo một đường line đã được xác định từ trước, sử dụng các cảm biến line follower để nhận diện và theo dõi vạch đường. Khi xe tiến đến một đoạn có biển báo, một camera được lắp đặt trên xe sẽ hoạt động để nhận diện các biển báo giao thông, như biển báo rẽ trái, rẽ phải hoặc dừng lại. Camera sẽ chụp và truyền hình ảnh về Raspberry Pi, nơi thực hiện quá trình xử lý ảnh. Quá trình này sử dụng các thuật toán Adaboost với cấu trúc cascaded giúp cải thiện độ chính xác và tốc độ nhận dạng biển báo đảm bảo kết quả nhận diện chính xác.

Sau khi Raspberry Pi xử lý ảnh và xác định loại biển báo, nó sẽ gửi tín hiệu về Arduino thông qua một giao tiếp UART. Arduino sẽ nhận tín hiệu này và dựa trên lệnh nhận được, điều khiển các động cơ của xe. Ví dụ, nếu camera nhận diện biển báo "rẽ trái", Raspberry Pi sẽ gửi lệnh cho Arduino để điều khiển động cơ quay sang trái. Nếu xe gặp biển báo "dừng lại", Arduino sẽ nhận tín hiệu dừng và ngắt động cơ, khiến xe dừng lại ngay lập tức. Ngoài ra, nếu xe gặp vật cản trên đường line, cảm biến va chạm hoặc khoảng cách sẽ phát hiện vật cản và gửi tín hiệu về Arduino để dừng xe lại hoặc thay đổi hướng đi. Arduino sẽ xử lý tín hiệu và điều khiển động cơ, đảm bảo xe không tiếp tục di chuyển khi gặp vật cản. Toàn bộ quá trình này yêu cầu sự phối hợp chính xác giữa Raspberry Pi, camera, và Arduino để đảm bảo xe hoạt động đúng theo các tình huống giao thông và an toàn trên đường. Nhìn chung, mô hình hoạt động ổn định tốt, sử dụng nguồn cấp điện nhỏ, an toàn với người sử dụng. Hệ thống điều khiển vận hành ổn định, tốc độ truyền nhận dữ liệu chính xác trong điều kiện ít vật cản. Mức tiêu thụ điện năng thấp, phù hợp cho các ứng dụng trong giao thông, công nghiệp và tự động hóa.

4. Kết luận

Bài báo nghiên cứu thiết kế mô hình xe tự hành ứng dụng các công nghệ xử lý ảnh và trí tuệ nhân tạo để nhận diện biển báo giao thông, những kỹ thuật và công nghệ tiên tiến đã được áp dụng giải quyết được nhiều hạn chế, khuyết điểm của các phương pháp thiết kế xe tự hành trước đây đã làm. Thuật toán Cascade Adaboost ứng dụng cho bài toán nhận dạng biển báo giao thông đảm bảo tính chính xác cao, dễ dàng cho việc triển khai trên các nền tảng máy tính nhúng. Các kết quả kiểm thử phương pháp trên trình giả lập Unity cho thấy tính hiệu quả, khả thi của phương pháp điều hướng cho xe tự hành. Mở ra nhiều hướng phát triển và ứng dụng mới mẻ trong ngành công nghiệp, tự động hóa ở Việt Nam hiện nay. Hệ thống được thiết kế với độ ổn định cao, độ chính xác tin cậy, giao diện trực quan với bảng số liệu được thể hiện ở Bảng 1 về kết quả huấn luyện phân lớp biển báo bằng thuật toán Cascade Adaboost và Bảng 2 về ma trận nhầm lẫn cho mỗi loại biển báo. Kết quả thực nghiệm và khảo sát mô hình hệ thống trên nhiều môi trường tương đối chính xác và phù hợp. Các sai số về độ trễ trong khâu xử lý ảnh xảy ra có thể từ nhiều nguyên nhân khác nhau nhưng nguyên nhân chính vẫn là do thiết bị nhận diện do xe vừa di chuyển vừa xử lý ảnh.

Ứng dụng xử lý ảnh dựa trên mã nguồn mở đã nhận diện biển báo một cách chính xác. Mô hình hệ thống có tính ứng dụng cao trong cuộc cách mạng công nghệ khoa học kỹ thuật hiện đại ngày nay. Vì vậy, nghiên cứu có thể được phát triển, mở rộng cải tiến thêm với nguồn xử lý dữ liệu ảnh lớn hơn. Hệ thống có thể được ứng dụng trong giao thông nói riêng và trong nhiều sản phẩm hữu ích khác.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] T. Inagaki and T. B. Sheridan, "A critique of the SAE conditional driving automation definition, and analyses of options for improvement," *Cogn. Tech Work*, vol. 21, no. 4, pp. 569–578, Nov. 2019.
- [2] P. Srinivas *et al.*, "Raspberry pi based personal voice assistant using python," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 11, pp. 105–108, 2020.
- [3] C. Li, J. Wang, X. Wang, and Y. Zhang, "A model based path planning algorithm for self-driving cars in dynamic environment in 2015," *Chinese Automation Congress (CAC)*, Wuhan, China, Nov. 2015, pp. 1123–1128.
- [4] R. K. Satzoda, S. Sathyanarayana, T. Srikanthan, and S. Sathyanarayana, "Hierarchical Additive Hough Transform for Lane Detection," *IEEE Embedded Syst. Lett.*, vol. 2, no. 2, pp. 23–26, Jun. 2010.
- [5] S. Patil, P. Ovhal, S. Jigale, and R. Aneesh, "Robotic Car for Avoidance and Detection of obstacles using IR Sensor," *IRJET*, vol. 7, no. 4, pp. 808-812, Apr. 2020.
- [6] A. Alghmgham, G. Latif, J. Alghazo, and L. Alzubaidi, "Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN," *Procedia Computer Science*, vol. 163, pp. 266–274, 2019.
- [7] V. A. M. M. Vujović, "Raspberry Pi as a Sensor Web node for home automation," *Computers & Electrical Engineering*, vol. 44, pp. 153-171, 2015.
- [8] L. Kruno, K. Andrej, C. Robert, and P. Ivan, "Fast planar surface 3D SLAM using LIDAR," *Robotics and Autonomous Systems*, vol. 92, pp. 197-220, 2017.
- [9] Y. Chang, C. Hui, L. Ju, Z. Di, and X. Yanyan, "Robust Lane Detection for Complicated Road Environment Based on Normal Map," *IEEE Access*, vol. 6, pp. 49679-49689, 2018, doi: 10.1109/ACCESS.2018.2868976.
- [10] E. Yurtsever *et al.*, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, pp. 58443-58469, 2020.
- [11] H. Zhao *et al.*, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE transactions on intelligent transportation systems*, vol. 21, pp. 1086-1095, 2019.