

THIẾT KẾ VÀ THI CÔNG XE BẮM ĐỐI TƯỢNG

Lê Khắc Sinh⁽¹⁾

Trường Đại học Công nghiệp Thực phẩm TPHCM

Ngày nhận bài: 17/03/2022; Ngày gửi phản biện: 7/4/2022; Chấp nhận đăng: 20/5/2022

Liên hệ Email: sinhlk@hufi.edu.vn

<https://doi.org/10.37550/tdmu.VJS/2022.03.301>

Tóm tắt

Bài báo này nghiên cứu ứng dụng xử lý ảnh để xác định đối tượng và từ đó, định vị đối tượng. Bài báo cũng ứng dụng bộ điều khiển vi tích phân tỷ lệ PID (Proportional Integral Derivative) để điều khiển tốc độ của động cơ trên xe theo giá trị được đặt trước nhằm bám chính xác theo đối tượng mong muốn. Phần xử lý ảnh sẽ ứng dụng thị giác máy tính để phát hiện đối tượng trên Kit Raspberry Pi 3. Dựa vào video đầu vào, Kit Raspberry Pi 3 sẽ tính toán các thông số cần thiết cho Kit Arduino thông qua chuẩn giao tiếp UART (Universal Asynchronous Receiver/Transmitter). Sau đó, Kit Arduino sẽ xuất các xung điều chế độ rộng xung PWM (Pulse Width Modulation) từ bộ điều khiển PID để điều khiển động cơ cho xe bám theo đối tượng khi đối tượng di chuyển theo thời gian thực. Nhiều kết quả minh họa cho thấy xe bám theo đối tượng với độ chính xác cao.

Từ khóa: Arduino, PID, PWM, Raspberry Pi 3, xử lý ảnh

Abstract

DESIGN AND IMPLEMENT THE OBJECT-TRACKING VEHICLE

This article researches the application of image processing to identify the object and then, determine the location of the object. This article also applies the Proportional Integral Derivative (PID) controller to adjust the speed of the vehicle's engine according to a preset value to precisely track the desired object. The image processing part will apply computer vision to detect the object on Kit Raspberry Pi 3. Based on the input video, the Kit Raspberry Pi 3 will calculate necessary parameters for Kit Arduino through UART (Universal Asynchronous Receiver/Transmitter) communication standard. Then, the Kit Arduino will output PWM (Pulse Width Modulation) pulses from the PID controller to control motors for the vehicle to track the object when the object is moving in real time. The results in this article expose that the vehicle tracks the object with high accuracy.

1. Mở đầu

Các nhà nghiên cứu trong và ngoài nước đang quan tâm rất nhiều về xử lý ảnh và thị giác máy tính trong những năm gần đây (Habib và cs., 2022; Sheela, 2022; Kan và cs.,

2022; Wei và cs., 2022). Ngoài ra, thị giác máy tính có thể kết hợp với các kỹ thuật khác như điện tử, tự động hóa, cơ khí, truyền thông, công nghệ thông tin, ... để tạo ra vô số ứng dụng cho từ đời sống hàng ngày đến quân sự, y học, an ninh, ... (Pratapagiri và cs., 2022; Dinesh và cs., 2022; Mujaddid và cs., 2022). Bài báo này sẽ ứng dụng thị giác máy tính để xác định đối tượng và từ đó, điều khiển xe bám theo đối tượng mong muốn.

Xe bám đối tượng đã được nghiên cứu trong nhiều công trình đã công bố (Cowarobot, 2022; Samuel, 2018; Guillen-Garcia và cs., 2022; Liu & Lin, 2022; Li và cs., 2022; Lee và cs., 2022; Guo & Zhao, 2022). Các công trình này được tóm tắt như sau. Cowarobot (xem hình 1) là vali robot sử dụng trí tuệ nhận tạo đầu tiên trên thế giới có thể tự động bám theo chủ nhân của mình (Cowarobot USA, 2022). Không cần thiết bị đeo hoặc điều khiển từ xa từ chủ nhân của mình, Cowarobot với khả năng tự động di chuyển và được hỗ trợ bởi AI sẽ nhận ra chủ nhân của nó một cách tự động. Ngoài ra, Cowarobot còn có khả năng tránh chướng ngại vật thông minh giúp Cowarobot tránh chướng ngại vật và con người để nhanh chóng trở lại đúng đường đi. Cách thức theo dõi thông minh giúp

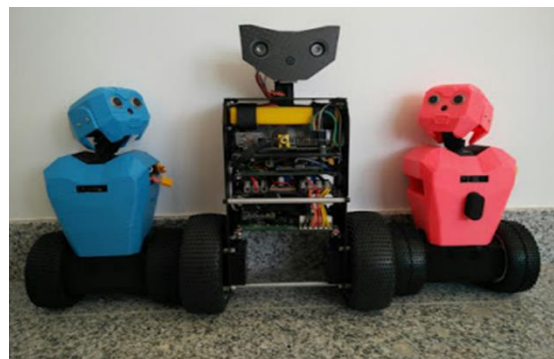
Ngoài Cowarobot thì hệ thống tự cân bằng và bám theo đối tượng mang tên RS5 Robot đã được nêu ra trong (Samuel, 2018). RS5 Robot này có khả năng tự cân bằng với hai bánh xe và di chuyển theo một đối tượng được chọn trước (xem hình 2). Hơn thế nữa, RS5 Robot di chuyển khá mượt mà trong điều kiện ánh sáng ổn định và đối tượng cần bám theo di chuyển với tốc độ vừa phải. Ngoài các sản phẩm thương mại (Cowarobot, 2022; Samuel, 2018), các công trình nghiên cứu khác như (Guillen-Garcia và cs., 2022; Liu & Lin, 2022; Li và cs., 2022; Lee và cs., 2022; Guo & Zhao, 2022) cũng đề xuất các giải

Cowarobot biết bạn đang ở đâu và gửi cảnh báo khi bạn ở cách xa hơn 6 feet. Cowarobot đã trở thành sản phẩm thương mại và có được nâng cấp thêm nhiều chức năng khác để thỏa mãn các yêu cầu của người dùng.



Hình 1. Cowarobot (Cowarobot USA, 2022)

pháp về bám theo đối tượng. Tuy nhiên, (Guillen-Garcia và cs., 2022; Liu & Lin, 2022; Li và cs., 2022; Lee và cs., 2022; Guo & Zhao, 2022) không đề cập đến giải pháp bám theo đối tượng dựa vào màu sắc như trong bài báo này.



Hình 2. RS5 Robot (Samuel, 2018)

Bài báo này thiết kế và thi công xe bám theo đối tượng. Xe bám theo đối tượng dựa vào hình ảnh thu được của đối tượng từ camera được gắn trên xe. Tuy nhiên, khi đối tượng chuyển động liên tục thì việc phát hiện và bám theo đối tượng là rất khó khăn. Điều này bởi vì thiết kế hệ thống xử lý ảnh dựa trên vi điều khiển rất khó hiện thực. Ngoài ra, việc lập trình và sử dụng các phần mềm và thư viện hỗ trợ có sẵn như libuvc, OpenCV, libjpg là không khả thi. Bài báo này đề xuất giải pháp cho vấn đề trên là sử dụng máy tính nhúng linux để thu thập và xử lý video đầu vào. Với giải pháp này, chúng ta có thể sử dụng dễ dàng các thư viện hỗ trợ xử lý ảnh có sẵn. Bài báo này sử dụng Kit Raspberry Pi 3 có gắn module camera để thu thập video và khai thác thư viện OpenCV để xử lý hình ảnh và tính toán mà từ đó trả về các thông số cần thiết về cho Arduino. Từ các thông số này, chúng ta có thể tính toán và lập trình cho Arduino cấp xung phù hợp để điều khiển các động cơ để xe di chuyển bám theo đối tượng mong muốn.

Bài báo này mang lại các đóng góp chính sau đây:

- Chúng tôi thiết kế thành công xe bám theo đối tượng với kết cấu đơn giản, nhỏ gọn, hoạt động ổn định trong điều kiện ánh sáng không quá tối và môi trường xung quanh không có vật cùng màu với đối tượng cần bám theo.

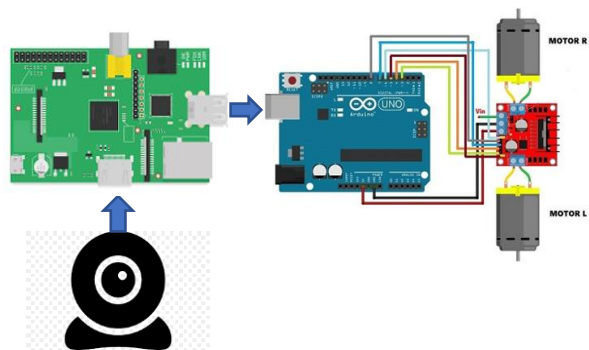
- Kết quả thiết kế của chúng tôi có thể hoàn thiện thêm thành một giải pháp IoT (Internet of Things) để phục vụ đời sống hàng ngày (vali thông minh) lẫn công nghiệp (xe đẩy hay kéo hàng thông minh).

Phần tiếp theo của bài báo trình bày thiết kế phần cứng trong khi đó phần 3 trình bày thi công hệ thống. Phần 4 trình bày các kết quả minh họa trong khi đó phần 5 nêu ra các kết luận cho bài báo này.

2. Thiết kế phần cứng

Thiết kế phần cứng của xe bám theo đối tượng được mô tả ở hình 3. Phần cứng của xe bao gồm một camera để thu nhận video đầu vào. Sau đó, video đầu vào được tách thành các khung ảnh để từ đó xử lý và phát hiện đối tượng trong khung ảnh sử dụng Kit Raspberry Pi 3 (Priyanka và cs., 2021). Sau khi xác định được đối tượng cần bám theo thì Kit Raspberry Pi 3 sẽ trả về các thông số như diện tích của đối tượng trong hình và độ lệch của đối tượng so với điểm trung tâm của khung ảnh cho Arduino theo chuẩn giao tiếp UART. Arduino cũng ứng dụng giải

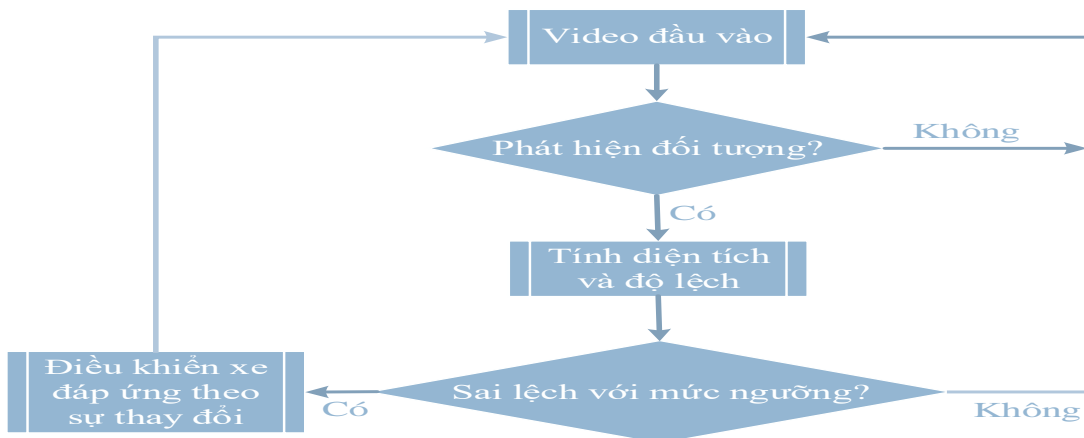
thuật điều khiển PID để điều khiển bốn bánh xe di chuyển tiến, lùi hay qua trái, qua phải để bám theo đối tượng dựa vào các thông số về vị trí được gửi xuống từ Kit Raspberry Pi 3 trước đó.



Hình 3. Sơ đồ khối phần cứng của hệ thống xe bám theo đối tượng

3. Thi công hệ thống

Lưu đồ giải thuật vận hành của xe bám theo đối tượng được trình bày như hình 4. Video đầu vào thu được từ webcam bao gồm các khung ảnh. Trước tiên, kích thước khung ảnh được thay đổi để có chiều rộng 400 pixel. Việc giảm kích thước khung ảnh như vậy cho phép xử lý khung ảnh nhanh hơn, dẫn đến tăng số khung ảnh trong một giây FPS (Frame Per Second) do giới hạn phần cứng của Kit Raspberry Pi 3. Sau đó, khung ảnh sẽ được làm mờ để giảm nhiễu tần số cao và để tập trung vào các đối tượng có cấu trúc bên trong khung ảnh, ví dụ như quả bóng. Cuối cùng, khung ảnh sẽ được chuyển đổi sang hệ màu HSV (Hue, Saturation, Value). Khi khởi tạo hệ thống, các mức giới hạn dưới và giới hạn trên cho màu sắc cần xác định đều được định nghĩa. Trong bài báo này, chúng tôi minh họa đối tượng là quả bóng màu xanh lá. Hệ thống sẽ xử lý và tính toán ra các vùng ảnh có chứa màu xanh lá. Khi tìm thấy vị trí quả bóng, một hình tròn bao quanh nó sẽ được xác định. Từ đó, hệ thống sẽ tính được vị trí của quả bóng và độ lệch của nó để trả xuống Arduino. Arduino dùng các tín hiệu vị trí này để cấp xung PWM phù hợp để điều khiển xe bám theo đối tượng.

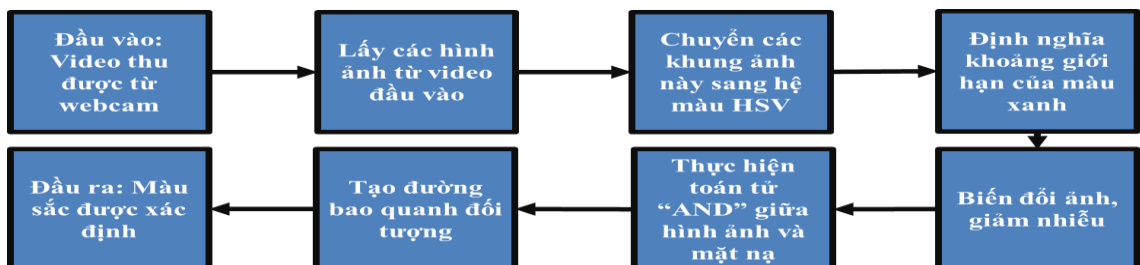


Hình 4. Lưu đồ giải thuật vận hành của xe bám theo đối tượng

Để hiểu một cách tường minh hơn, chúng ta có thể tách hệ thống làm hai phần như sau:

3.1. Phần 1: thu nhận và xử lý hình ảnh

Phần này để phát hiện trái bóng, từ đó lấy ra các thông số vị trí gửi xuống cho Arduino. Phần này gồm tám bước được liệt kê như sau:



Hình 5. Quy trình thu nhận và xử lý ảnh

Bước 1: Thu video đầu vào từ webcam.

```
vs = VideoStream(src=0).start()
time.sleep(2.0)
```

Bước 2: Lấy các khung ảnh từ video đầu vào để tiến hành xử lý bước tiếp theo.

```
frame = vs.read()
```

Bước 3: Định lại kích thước khung ảnh để tăng tốc độ xử lý do hạn chế về phần cứng của Kit Raspberry Pi 3. Sau đó, chuyển các khung ảnh từ hệ màu RGB (Red, Green, Blue) sang hệ màu HSV để dễ dàng xử lý hơn (Luis, 2015).

```
frame = imutils.resize(frame, width=450)
blurred = cv2.GaussianBlur(frame, (11, 11), 0)
hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
```

Bước 4: Xác định phạm vi của màu và tạo mặt nạ tương ứng với khoảng màu được định nghĩa từ trước. Trong bài báo này, như là ví dụ minh họa, chúng tôi chọn màu xanh lá là màu cần xác định.

```
greenLower = (29, 86, 6)
greenUpper = (64, 255, 255)
```

Bước 5: Biến đổi ảnh để loại bỏ những nhiễu trong ảnh (Python cv2.erode() Examples, 2022).

```
mask = cv2.erode(mask, None, iterations=2)
mask = cv2.dilate(mask, None, iterations=2)
```

Bước 6: Thực hiện phép “AND” bit giữa khung ảnh đầu vào và mặt nạ để xác định vùng màu xanh trong ảnh.

```
mask = cv2.inRange(hsv, greenLower, greenUpper)
```

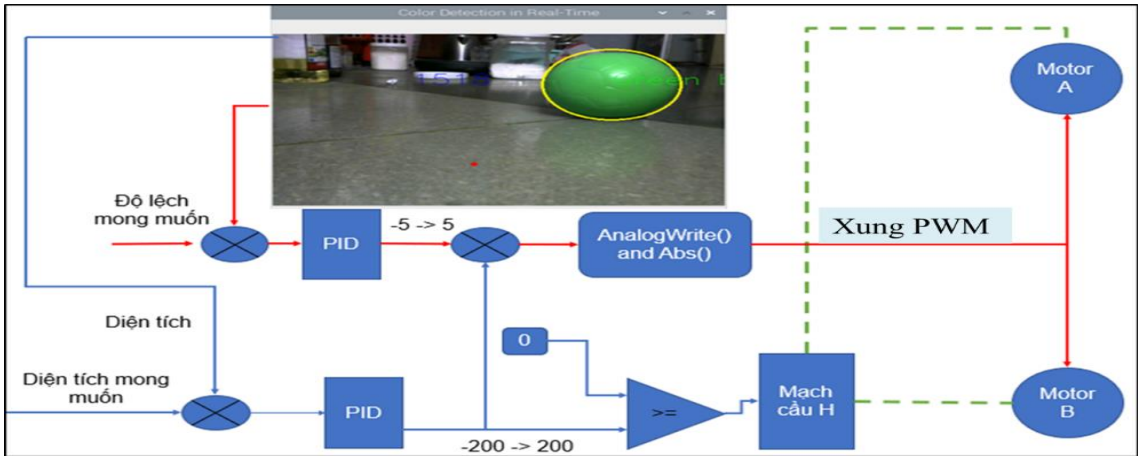
Bước 7: Tạo đường viền cho vùng màu xanh sau khi đã xác định.

```
cv2.circle(frame, (int(x), int(y)), int(radius), (0, 255, 255), 2)
```

Bước 8: Xuất ra các thông số cần thiết để điều khiển đối tượng.

```
def tranfer():
    start_buff = "9876\r"
    ser.write(start_buff.encode())
    str_area = str(S) + "\r"
    ser.write(str_area.encode())
    str_x = str(int(x)) + "\r"
    ser.write(str_x.encode())
```

3.2. Phần 2: điều khiển xe di chuyển theo đối tượng



Hình 6. Bộ điều khiển PID điều khiển hai động cơ chạy bám theo đối tượng

Các thông số về diện tích và độ lệch sau khi được tính toán sẽ được truyền xuống Arduino qua chuẩn giao tiếp UART như Hình 6. Để bám theo đối tượng thì chúng tôi đề xuất giải thuật có hai ràng buộc. Ràng buộc thứ nhất là khoảng cách của đối tượng và xe phải là cố định. Dựa vào hình ảnh thu về từ webcam, Kit Raspberry Pi 3 sẽ tính toán khoảng cách và trả xuống Arduino. Từ đó, sai lệch giữa giá trị đặt và giá trị đo sẽ được tính ra. Giá trị sai lệch này được đưa qua bộ điều khiển PID để tính toán đầu ra với giá trị đã được chặn trong khoảng từ -200 đến +200 (Arduino PID Library, 2022). Chiều quay của động cơ sẽ phụ thuộc vào dấu của giá trị sai lệch và được điều khiển bởi mạch cầu H. Ràng buộc thứ hai là luôn giữ hoành độ điểm tâm của đối tượng nằm ở chính giữa khung ảnh. Từ hoành độ tâm của đối tượng, chúng ta cũng tính được giá trị sai lệch hoành độ. Tương tự như cách làm với sai lệch về diện tích thì sai lệch về hoành độ cũng được đưa qua bộ điều khiển PID, rồi tính ra đầu ra để cấp xung PWM cho motor B. Khi đối tượng lệch sang trái thì giá trị này sẽ dương và vì vậy, bánh xe bên phải sẽ quay nhanh hơn để đưa đối tượng trở về vị trí chính giữa. Tương tự khi đối tượng lệch qua phải thì giá trị sai lệch sẽ âm, từ đó bánh xe bên phải sẽ quay chậm lại để đưa đối tượng về chính giữa khung ảnh.

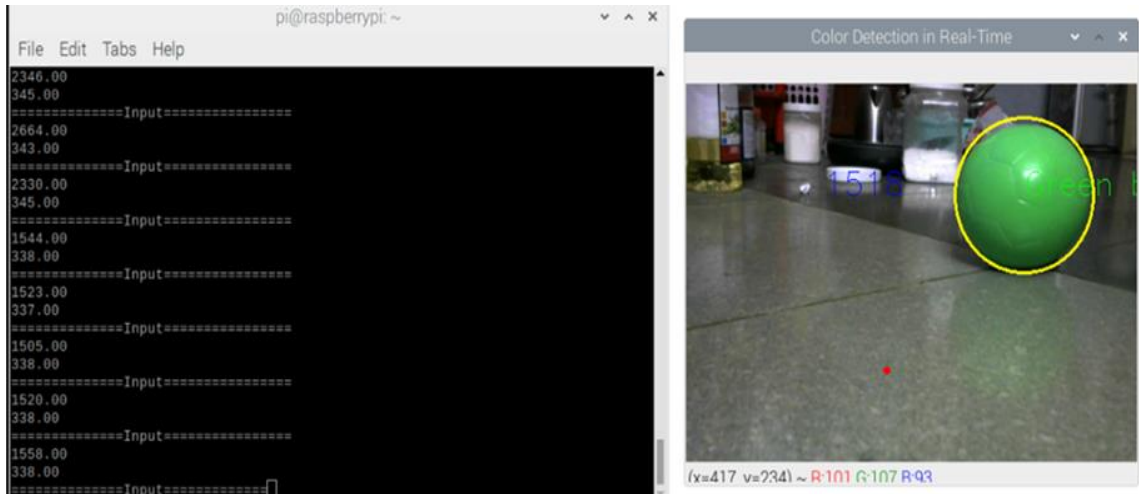
4. Kết quả minh họa

Hình 7 cho thấy mô hình xe bám đối tượng khi hoàn thành. Xe nhỏ gọn và khi thực thi chương trình thì xe có thể hoạt động độc lập. Nguồn cấp từ pin sạc dự phòng cho Kit Raspberry Pi 3 và Arduino. Pin Lipo dùng cấp nguồn cho hai động cơ có thể hoạt động khoảng một giờ đồng hồ.



Hình 7. Mô hình xe bám đối tượng khi hoàn thành

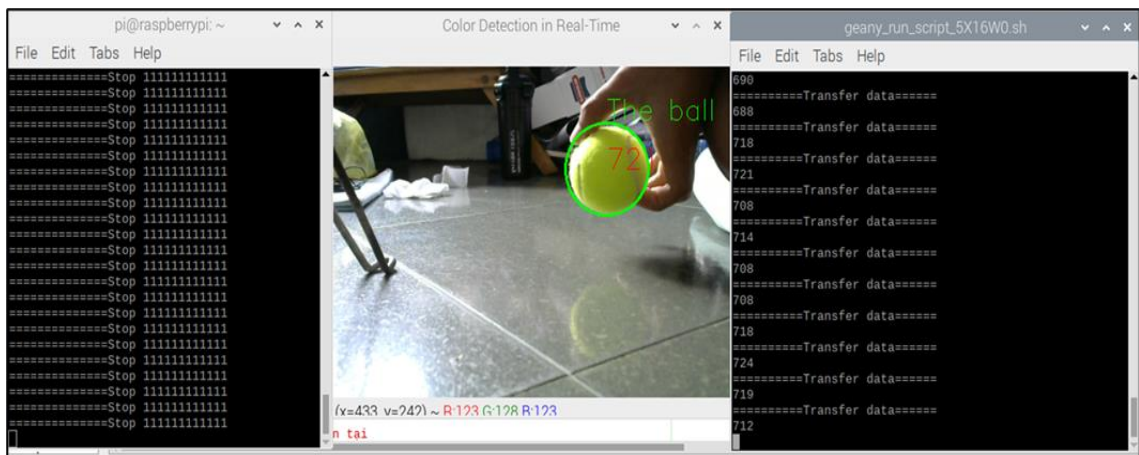
Môi trường chạy thử nghiệm của xe là trong phòng, cả ban ngày với ánh sáng tự nhiên và ban đêm khi bật đèn thì hệ thống vẫn hoạt động ổn định. Xe hoạt động gần như tương đồng trong cả hai điều kiện ánh sáng trên nên chúng tôi chỉ trình bày phần kết quả khi xe chạy trong điều kiện ánh sáng ban ngày để tránh trùng lặp. Xe có thể di chuyển bám theo trái bóng khi trái bóng di chuyển theo quỹ đạo đường thẳng ra xa hoặc lại gần xe trong khoảng từ 15cm đến 120cm.



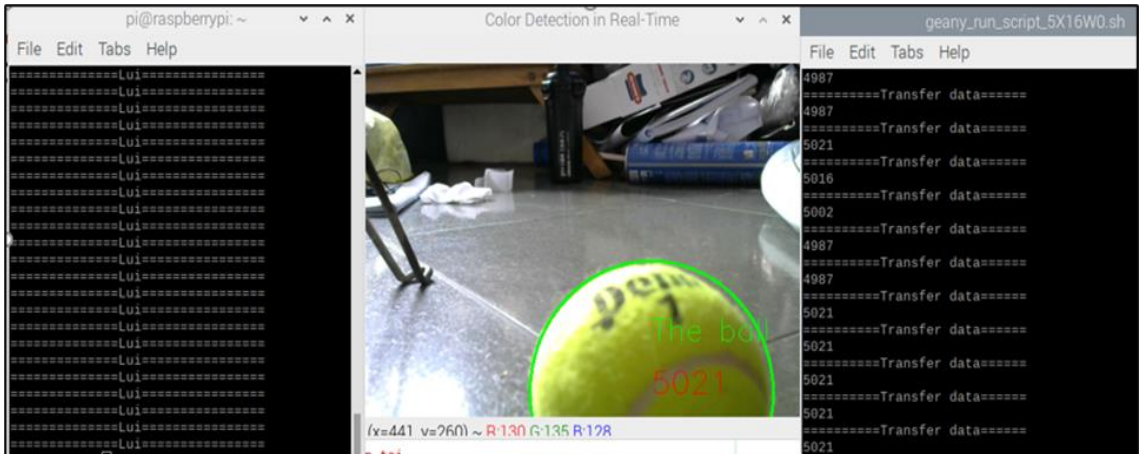
Hình 8. Hệ thống phát hiện trái bóng

Hệ thống phát hiện trái bóng, vẽ đường bao quanh, tính diện tích và tọa độ x được minh họa trên Hình 8. Diện tích và tọa độ được truyền từ Kit Raspberry Pi 3 qua Arduino theo chuẩn giao tiếp UART rất thành công.

Sau đây, một vài kết quả minh họa quá trình hoạt động của xe trong điều kiện ánh sáng tự nhiên trong phòng sẽ được trình bày. Hình 9 minh họa trường hợp xe đứng im khi quả bóng ở khoảng cách mong muốn so với xe. Mức ngưỡng ban đầu chúng tôi đặt cho diện tích quả bóng trong hình là 700 và cho phép chênh lệch 20% (tương đương với khoảng cách sấp xỉ 25-35cm trong thực tế), và giá trị đọc về từ camera là 721. Lúc này xe sẽ đứng yên vì thỏa mãn khoảng cách với đối tượng cần bám theo.

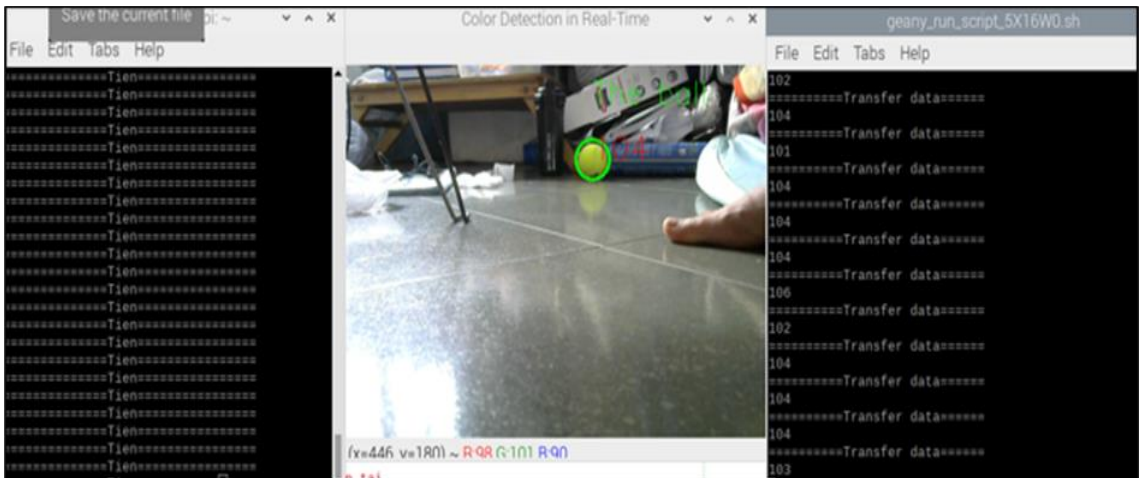


Hình 9. Xe đứng im khi quả bóng ở khoảng cách mong muốn so với xe



Hình 10. Xe đi lùi khi trái bóng gần xe hơn mức ngưỡng

Hình 10 minh họa trường hợp xe sẽ đi lùi khi trái bóng gần xe hơn mức ngưỡng. Diện tích đọc được từ hệ thống xử lý ảnh là 5021 (tương đương với khoảng cách thực tế xấp xỉ 15cm) lớn hơn so với ngưỡng ban đầu và do đó, xe sẽ đi ra xa trái bóng cho tới khi đọc được giá trị diện tích thỏa mãn mức ngưỡng ban đầu.



Hình 11. Xe đi tiến khi trái bóng xa xe hơn mức ngưỡng

Hình 11 minh họa trường hợp xe đi tiến khi trái bóng xa xe hơn mức ngưỡng. Diện tích đọc được từ hệ thống xử lý ảnh là 104 (tương đương với khoảng cách thực tế xấp xỉ 120cm) lớn hơn so với ngưỡng ban đầu và do đó, xe sẽ tiến lại gần trái bóng cho tới khi đọc được giá trị diện tích thỏa mãn mức ngưỡng ban đầu.

Bài báo đã thi công được mô hình xe tự động bám theo đối tượng. Phần xử lý ảnh trên Kit Raspberry Pi 3 qua thử nghiệm đã phát hiện mục tiêu chính xác với tốc độ nhanh và trả về các hệ số cần thiết cho Arduino Uno R3 qua chuẩn giao tiếp UART rất thành công. Tuy nhiên, việc xử lý ảnh bị ảnh hưởng nhiều bởi ánh sáng nên khi ánh sáng không ổn định, ví dụ như thiếu ánh sáng hay ánh sáng màu chiếu lên đối tượng, thì đối tượng sẽ bị thay đổi màu sắc trong khung ảnh và dẫn đến nhiễu tăng lên, có thể gây ra nhận dạng bị sai. Hoặc khi có vật màu xanh khác có kích thước lớn đi vào khung hình cũng làm hệ

thống xác định sai đối tượng. Hơn nữa, các kết quả thực nghiệm cho thấy Arduino Uno R3 nhận chính xác các thông số được Kit Raspberry Pi 3 truyền xuống theo thời gian thực, từ đó cấp xung PWM điều khiển chính xác các động cơ.

5. Kết luận

Bài báo này đã thiết kế và thi công thành công xe bám theo đối tượng. Xe được thiết kế có kết cấu nhỏ gọn, đơn giản, và hoạt động ổn định trong các điều kiện ánh sáng khác nhau. Xe di chuyển khá mượt khi đối tượng cần bám theo di chuyển với tốc độ vừa phải. Do đó, mô hình xe đã được thi công có thể phát triển và hoàn thiện thêm để trở thành sản phẩm thương mại. Các hướng phát triển để hoàn thiện mô hình xe hiện hữu là cải thiện tốc độ xử lý bằng cách nâng cấp bộ xử lý trung tâm để có thể huấn luyện phát hiện các đối tượng đặc thù, từ đó đi theo một đối tượng theo lựa chọn của người dùng. Ngoài ra, bài báo này có thể phát triển thêm tính năng tránh vật cản để hệ thống có thể hoạt động trong môi trường có nhiều vật cản hay phát triển trên các ứng dụng điện thoại để cảnh báo khi mục tiêu đi quá xa hệ thống cũng như theo dõi được camera của hệ thống trên ứng dụng của điện thoại. Hơn thế nữa, bài báo này cũng có thể thi công thêm hệ thống định vị để tránh bị thất lạc xe. Cuối cùng, ý tưởng của bài báo này có thể hoàn thiện hơn bằng cách thay đổi kết cấu cơ khí của xe để mở rộng ứng dụng như xe vận chuyển hàng hóa hay vali thông minh.

TÀI LIỆU THAM KHẢO

- [1] Arduino PID Library (2022). <https://playground.arduino.cc/Code/PIDLibrary/>.
- [2] Cowarobot R1 Black (2022). Copyright © 2022 COWAROBOT USA Powered by Shopif. <https://cowarobotusa.com/collections/frontpage/products/cowarobot-r1-black>.
- [3] Dinesh, G., Singh, A., Shinz, N., Sravanya, M. V., & Agrawal, G. (2022). Mandatory Automated Safety Kernel (MASK): Face Mask Recognition Using Deep Learning & Computer Vision. *Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 423-429.
- [4] Guillen-Garcia, J., Palacios-Alonso, D., Cabello, E., & Conde, C. (2022). Unsupervised Adaptive Multi-Object Tracking-by-Clustering Algorithm With a Bio-Inspired System. *IEEE Access*, 10, 24895-24908.
- [5] Guo, G., & Zhao, S. (2022). 3D Multi-Object Tracking with Adaptive Cubature Kalman Filter for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 1. <https://doi.org/10.1109/TIV.2022.3158419>
- [6] Habib, M. T., Raza, D. M., Islam, M. M., Victor, D. B., & Arif, M. A. I. (2022). Applications of Computer Vision and Machine Learning in Agriculture: A State-of-the-Art Glimpse. *International Conference on Innovative Trends in Information Technology (ICITIT)*, 1-5.
- [7] Kan, S., Cen, Y., Li, Y., Vladimir, M., & He, Z. (2022). Local Semantic Correlation Modeling over Graph Neural Networks for Deep Feature Embedding and Image Retrieval. *IEEE Transactions on Image Processing*, 1. <https://doi.org/10.1109/TIP.2022.3163571>

- [8] Lee, H., Lee, H., Shin, D., & Yi, K. (2022). Moving Objects Tracking Based on Geometric Model-Free Approach With Particle Filter Using Automotive LiDAR. *IEEE Transactions on Intelligent Transportation Systems*, 1-10. <https://doi.org/10.1109/TITS.2022.3155828>
- [9] Li, B., Fu, C., Ding, F., Ye, J., & Lin, F. (2022). All-Day Object Tracking for Unmanned Aerial Vehicle. *IEEE Transactions on Mobile Computing*, 1. <https://doi.org/10.1109/TMC.2022.3162892>
- [10] Liu, C.-J., & Lin, T.-N. (2022). DET: Depth-Enhanced Tracker to Mitigate Severe Occlusion and Homogeneous Appearance Problems for Indoor Multiple-Object Tracking. *IEEE Access*.
- [11] Luis (2015). <https://pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/>
- [12] Mujaddid, S., Zainuddin, Z., & Tahir, Z. (2022). Detecting Indiscriminate Disposal of Waste Using Computer Vision. *International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*, 114-119.
- [13] Pratapagiri, S., Madugula, S., Kiran, S., Rao, V. C. S., Venkatramulu, S., & Phridviraj, M. S. B. (2022). ML based Implementation for Documents Forensic and Prediction of Forgery using Computer Vision Framework. *Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 280-283.
- [14] Priyanka, E. B., Thangavel, S., Tharun, S., Saravanan, S. N., Sankar, S. R., Kumar, B. B., & Pugazhenth, C. (2021). IoT based Rash Braking Data Analysis and Plotting in Google Maps Using Raspberry Pi. *International Conference on Data Analytics for Business and Industry (ICDABI)*, 320-325.
- [15] Python cv2.erode() Examples (2022). <https://www.programcreek.com/python/example/89362/cv2.erode>.
- [16] Samuel (2018). <http://roboticssamy.blogspot.com/2018/01/old-vs-new-robot.html>.
- [17] Sheela, T. (2022). Cloud Based E-Commerce Application For Organic Fertilizers, Pesticides And Other Products And Crop Disease Identification Using Computer Vision. *International Conference on Computer Communication and Informatics (ICCCI)*, 1-3.
- [18] Wei, X., Yang, J., Lv, M., Chen, W., & Ma, X. (2022). Translational Motion Compensation for ISAR Imaging Based on Range Joint Fast Orthogonal Matching Pursuit Algorithm. *IEEE Access*, 1. <https://doi.org/10.1109/ACCESS.2022.3165020>.