BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM

**KHOA CÔNG NGHỆ ĐIỆN – ĐIỆN TỬ**

**KHÓA LUẬN TỐT NGHIỆP**

**ĐỀ TÀI: SỬ DỤNG CÔNG NGHỆ XỬ LÝ ẢNH DỰA TRÊN NỀN TẢNG RASPBERRY CHO HỆ THỐNG PHÂN LOẠI SẢN PHẨM THEO MÀU SẮC**

 DANH MỤC BẢNG BIỂU

Bảng 4.1. Bảng kết quả thực nghiệm...................................... 43

DANH MỤC VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Tiếng anh** | **Tiếng việt** |
| RPI | Raspberry Pi |  |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| OEM | Original Equipment Manufacturer | Nhà sản xuất thiết bị gốc |
| PCM | Pulse-Code Modulation | Bộ mã hóa mã xung |
| HDMI | High Definition Multimedia Interface | Giao diện đa phương tiện |
| GPIO | General purpose input/output | Cổng vào ra vạn năng |
| MP | Megapixel |  |
| IDE | Integrated Development Environment | Môi trường tích hợp để viết code |
| DC | Direct Current | Điện một chiều |
| AC | Alternating Current | Điện xoay chiều |
| PLC | Programmable Logic Controller | Bộ điều khiển logic lập trình được |
| LCD | Liquid Crystal Display | Màn hình tinh thể lỏng |
| OpenCV | Open Computer Vision | Thư viện mã nguồn mở |

**Chương 1.     GIỚI THIỆU ĐỀ TÀI**

**1.1      Đặt vấn đề.**

Ngày nay cùng với sự phát triển của các ngành khoa học kỹ thuật, kỹ thuật điện- điện tử và điều khiển tự động đóng vai trò hết sức quan trọng trong mọi lĩnh vực khoa học, quản lý, công nghiệp tự động hóa, cung cấp thông tin… Do đó chúng ta phải nắm bắt và vận dụng nó một cách có hiệu quả nhằm góp phần vào sự nghiệp phát triển nền khoa học kỹ thuật thế giới nói chung và trong sự phát triển kỹ thuật điều khiển tự động nói riêng. Với những kỹ thuật tiên tiến như vi xử lý, PLC, vi mạch số… được ứng dụng vào lĩnh vực điều khiển, thì các hệ thống điều khiển thô sơ, với tốc độ xử lý chậm chạp ít chính xác được thay thế bằng các hệ thống điều khiển tự động với các lệnh chương trình đã được thiết lập trước.

Trong quá trình hoạt động ở các nhà xưởng, xí nghiệp hiện nay, việc tiết kiệm điện năng là nhu cầu rất cần thiết, bên cạnh đó ngành công nghiệp ngày càng phát triển các công ty xí nghiệp đã đưa tự động hóa và sản xuất để tiện ích cho việc quản lý dây chuyền và sản phẩm cho toàn bộ hệ thống một cách hợp lý là yêu cầu thiết yếu, tiết kiệm được nhiều thời gian cũng như quản lý một cách dễ dàng. Để đáp ứng yêu cầu đó, em đã tiến hành nghiên cứu tài liệu, thiết kế mô hình hệ thống phân loại sản phẩm theo màu sắc.

Những lợi ích mà hệ thống phân loại sản phẩm đem lại cho chúng ta là rất lớn, cụ thể như:

-       Giảm sức lao động, tránh được sự nhàm chán trong công việc, cải thiện được điều kiện làm việc của con người, tạo chongười lao động tiếp cận với sự tiến bộ của khoa học kỹ thuật và được làm việc trong môi trường ngày càng tốt hơn.

-       Nâng cao năng suất lao động, tạo tiền đề cho việc giảm giá thành sản phẩm, cũng như thay đổi mẫu mã một cách nhanh chóng và khoa học.

-       Giúp cho việc quản lý và giám sát trở nên rất đơn giản, bởi vì nó không những thay đổi điều kiện làm việc của công nhân mà còn có thể giảm số lượng công nhân đến mức tối đa…

**1.2      Các nghiên cứu liên quan trong và ngoài nước.**

**1.2.1Bài báo “*Detecting colors (Hsv Color Space) – Opencv with Python*”**

-     Tác giả thực hiện: Sergio Canu.

-     Thời gian đăng trực tuyến: 15/02/2019.

**Tóm tắt nội dung:**Phân loại màu sắc dựa trên OpenCV với ngôn ngữ lập trình Python.

**Kết quả đạt được:**Màu được tách một cách rõ ràng.

**Hình 1.1. Tách màu đỏ từ môi trường**

**1.2.2Bài báo “*Adruino Color Detection*”**

-       Tác giả thực hiện: Marcelo.

-       Thời gian đăng trực tuyến: 12/07/2017.

**Tóm tắt nội dung:**sử dụng Adruino phát hiện màu sắc với cảm biến màu sắc TSC 3200.

**Kết quả thực hiện:**màu được phát hiện và hiện kết quả tên màu sắc và hệ màu RGB lên màn hình LCD.

**Hình 1.2. Phát hiện màu vàng**

**Hình 1.3. Phát hiện màu đỏ**

**1.2.3Bài báo “*Hệ thống phân loại theo màu sắc xử dụng PLC S7 - 1200*”**

-       Tác giả thực hiện: Trần Khánh Dưng, Hoàng Như Ý.

-       Nơi thực hiện: Trường Đại Học Sư Phạm Kỹ Thuật, 03/07/2018, TP. Hồ Chí Minh.

**Tóm tắt nội dung:**sử dụng cảm biến TCS 3200 để phân loại màu sắc gửi tín hiệu cho PLC, xử dụng cảm biến hồng ngoại nhận biết vị trí cà chua và xử dụng xilanh khí nén để đẩy.

**Kết quả thực hiện:**mô hình được hoàn thiện, tồn tại một số hạn chế.

**Hình 1.4. Kết quả thi công phần cứng**

**1.2.4 Bài báo “*Delta Robot*”**

-       Tác giả thực hiện: Giáo sư Reymond Clave.

-       Nơi thực hiện: Tại EPFL - École Polytechnique Fédérale de Lausanne.

**Tóm tắt nội dung:** Ông đưa ra ý tưởng về việc sử dụng các khối hình bình hành để chế tạo một robot song song với ba phép tịnh tiến và một bậc tự do quay.

**Kết quả thực hiện:**Là một trong những thiết kế robot song song thành công nhất, với hàng trăm robot đang hoạt động trên toàn thế giới.

Hình 4. Thiết kế của Delta Robot.

**1.3      Mục tiêu**

Mục tiêu nghiên cứu của đề tài: “Thiết kế và thi công hệ thống phân loại sản phẩm theo màu sắc sử dụng công nghệ xử lý ảnh dựa trên nền tảng Raspberry” được chia làm các phần:

-       Tìm hiểu về các mô hình phân loại màu sắc có sẵn dựa theo đó chọn các thiết bị phù hợp cho mô hình.

-       Tìm hiểu về Raspberry và các thiết bị được chọn.

-       Tìm hiểu về cách kết nối giữa Raspberry và các thiết bị được chọn.

-       Tìm hiểu về ngôn ngữ lập trình Python.

-       Từ những kiến thức tìm hiểu được thiết kế mô hình tối ưu nhất trong khả năng.

**1.4      Phạm vi của đề tài.**

Về nội dung nghiên cứu: luận văn nghiên cứu xoay quanh tìm hiểu về nhận dạng, phân loại sản phẩm bằng phương pháp xử lý ảnh thông qua Kit Raspberry Pi 4 Model B.

Về không gian nghiên cứu: tập trung nghiên cứu xoay quanh xử lý ảnh, thực hiện phân loại bằng mô hình Delta Robot.

Về thời gian nghiên cứu: 4 tháng.

**Chương 2 CƠ SỞ LÝ THUYẾT**

**2.1.     Các linh kiện được sử dụng trong đề tài.**

**2.1.1     Kit Raspberry Pi 4 model B.**

**2.1.1.1    Giới thiệu về Raspberry Pi và Raspberry pi 4 Model B.**

Raspberry Pi là từ để chỉ các máy tính chỉ có một board mạch (hay còn gọi là máy tính nhúng) kích thước chỉ bẳng một thẻ tín dụng, được phát triển tại Anh bởi Raspberry Pi Foundation với mục đích ban đầu là thúc đẩy việc giảng dạy về khoa học máy tính cơ bản trong các trường học và các nước đang phát triển.

Nhiệm vụ ban đầu của dự án Raspberry Pi là tạo ra máy tính rẻ tiền có khả năng lập trình cho những sinh viên , nhưng Pi đã được sự quan tầm từ nhiều đối tượng khác nhau. Đặc tính của Raspberry Pi xây dựng xoay quanh bộ xử lí SoC Broadcom BCM2835 (là chip xử lí mobile mạnh mẽ có kích thước nhỏ hay được dùng trong điện thoại di động) bao gồm CPU , GPU , bộ xử lí âm thanh /video , và các tính năng khác … tất cả được tích hợp bên trong chip có điện năng thấp này .

Raspberry Pi sản xuất bởi 3 OEM: Sony, Qsida, Egoman. Và được phân phối chính bởi Element14, RS Components và Egoman.

**Hình 2.1. Raspberry Pi 4 Model B.**

Raspberry Pi 4 Model B nổi bật với chip 4 nhân 64-bit có tốc độ 1.4GHz. Phiên bản mới còn hỗ trợ Wifi Dual-band 2.4GHz và 5GHz, Bluetooth 5.0/Bluetooth Low Energy, cổng Ethernet tốc độ cao (300Mbps) và Power over Ethernet (PoE) thông qua PoE HAT.

Ngoài những thay đổi đó, kích thước bề ngoài cũng như phần mềm của Raspberry Pi 4 Model B không thay đổi. Nghĩa là bạn hoàn toàn có thể sử dụng các phụ kiện cũ như vỏ, thẻ nhớ, adapter (nguồn điện) hoặc các phần mềm đang chạy trên Raspbian/Ubuntu hiện tại. Tuy nhiên, về hệ điều hành thì các bạn cần tải bản mới nhất để có thể hỗ trợ được bo mạch mới nhất này.

**2.1.1.2    Thông số kỹ thuật của Raspberry Pi 4 Model B.**

**Hình 2.2. Thông số phần cứng của Raspberry Pi 3 Model B**

**Thông số kỹ thuật:**

-       Vi xử lý: Broadcom BCM2711,quad-core Cortex-A72 (ARMv8) 64-bit SoC @1.5GHz.

-       RAM: 2GB, 4GB hoặc 8GB LPDDR4 (tùy nhu cầu sử dụng).

-       Kết nối: 2.4GHz và 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 5.0, BLE, Gigabit Ethernet với 5 cổng USB 2.0.

-       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.265 (4Kp60 decode).

-       Lưu trữ: MicroSD.

-       Nguồn điện sử dụng: 5V/2.5A DC cổng microUSB 5V, DC trên chân GPIO, Power over Ethernet (PoE)  (yêu cầu thêm PoE HAT).

**2.1.1.3       Những lưu ý khi sử dụng Raspberry.**

Nguồn sử dụng cho Board là nguồn 5VDC, dòng điện từ 2,5A trở lên, chất lượng tốt và ổn định, trường hợp nguồn không đủ dòng sẽ dẫn đến hiện tượng sụt áp cháy máy.

Các chân GPIO của Raspberry Pi (40 chân cắm Header) nếu giao tiếp điện áp trên (lớn hơn) 3.3V, chân cấp nguồn 3.3V và 5V nếu chạm với GND (Mass) sẽ làm chập nguồn cháy máy ngay lập tức, nếu không có nhu cầu, vui lòng không sử dụng và nên mua thêm vỏ bảo vệ để tránh chập, chạm.

Nếu bạn cần sử dụng các chân GPIO của Raspberry Pi để giao tiếp với các mạch có mức điện áp giao tiếp lớn hơn 3.3V hoặc sử dụng với mạch điều khiển động cơ thì cần dùng thêm mạch chuyển tín hiệu.

**Hình 2.3. Sơ đồ chân của Raspberry Pi 4 Model B.**

**2.1.2     Module điều khiển động cơ A4988.**

A4988 là bo mạch điều của Allegro A4988 DMOS với tính năng bảo vệ động cơ khi giới hạn điện áp và quá dòng. Module cho phép điều khiểnđộng cơ bước lưỡng cực với dòng ra tối đa là 2A. Module điều khiển bước và hướng đơn giản cùng năm bước khác nhau là 1 bước, 1/2 bước, 1/4 bước, 1/8 bước và 1/16 bước. Điều chỉnh dòng điện theo dòng định mức của động cơ, dòng điện cực đại khi điều chỉnh chiết áp, cho phép bạn sử dụng điện áp cao hơn điện áp định mức của động cơ bước để đạt được tốc độ bước cao hơn. Điều khiển băm xung thông minh tự động chọn chế độ di chuyển chính xác (di chuyển nhanh hoặc di chuyển chậm). Ngắt điện khi quá nhiệt, khóa điện áp ngược và bảo vệ dòng điện. Bảo vệ ngắn mạch và quá tải. A4988 cũng dễ dàng kết hợp với arduino thay vì sử dụng các module điều khiển Servo đắt tiền khác.

**Hình 2.4. Module điều khiển động cơ bước A4988.**

**Thông số kỹ thuật:**

|  |  |
| --- | --- |
| **Điện áp định mức:** | 8 V. |
| **Điện áp cực đại:** | 35 V. |
| **Dòng điện mỗi pha:** | 1 A. |
| **Dòng cực đại mỗi pha:** | 2 A. |
| **Điện áp hoạt động tối thiểu:** | 3 V. |
| **Điện áp hoạt động định mức:** | 5.5 V. |
| **Số bước điều khiển:** | 1, 1/2, 1/4, 1/8, 1/16. |
| **Kích thước:** | 0.6 x 0.8cm. |
| **Trọng lượng:** | 1.3 g. |

**2.1.3     Băng tải**

Băng tải hay còn gọi băng chuyền là một thiết bị dùng vận chuyển, chuyền tải vật liệu, hàng hóa từ vị trí này đến vị trí kia. So với các phương thức vận tải khác như vận tải đường bộ và đường sắt, hệ thống băng tải có nhiều lợi thế như công suất cao hơn, chi phí thấp hơn, hiệu quả cao hơn, ít sự tham gia của con người và độ tin cậy đã được chứng minh.

**Hình 2.6. Băng tải mini sử dụng cho các mô hình.**

Cấu tạo băng tải được sử dụng trong mô hình:

-       Khung băng tải được thiết kế từ nhôm định hình.

-       Dây băng tải là bằng dây PVC,có độ dày 2mm, có màu xanh lá.

-       Hệ thống con lăn đỡ dây băng tải.

-       Động cơ bướcNEMA 17.

**Hình 2.7. Động cơ bước NEMA 17.**

**Thông số kỹ thuật của động cơ:**

-       Điện áp: 12VDC.

-       Dòng định mức: 2.5A.

-       Tốc độ không tải: 52RPM (52 vòng 1 phút).

-       Đường kính trục: 5mm.

-       Chiều dài trục: 14mm.

**2.1.5   Module Arduino Nano.**

**Hình 2.12. Module Arduino Nano.**

Arduino Nano là một bo mạch nhỏ, hoàn chỉnh với bo mạch, dựa trên ATmega328 (Arduino Nano 3.x). Module chỉ thiếu giắc cắm nguồn DC và chỉ hoạt động khi được cấp nguồn với cáp USB Mini-B thay vì cáp tiêu chuẩn. Arduino Nano cũng dễ dàng kết hợp với Raspberry theo phương pháp Master - Slaves.

BẢNG 7. Thông số kỹ thuật của Arduino Nano:

|  |  |
| --- | --- |
| **Bộ vi điều khiển:** |  |
| **Thiết kế:** |  |
| **Điện áp hoạt động:** |  |
| **Bộ nhớ hoạt động:** | 32 KB (2 KB dùng cho bộ nạp). |
| **SRAM( bộ nhớ lưu trữ):** |  |
| **Tốc độ truy cập**: | 16 MHz. |
| **Cổng kết nối:** |  |
| **Bộ nhớ trong:** |  |
| **Dòng điện mỗi chân:** |  |
| **Dòng hoạt động:** | 7 – 12V. |
| **Chân kết nối:** | 22 (6 chân xung PWM). |
| **Chân xung PWM:** |  |
| **Dòng tiêu thụ:** |  |
| **Kích thước:** | 18 x 45mm. |
| **Nặng:** |  |
| **Mã sản phẩm:** |  |

**2.1.6   Động cơ bước Nema 17.**

**Hình 2.14. Động cơ bước Nema 17.**

Nema 17 là động cơ bước lưỡng cực với góc bước 1.8° (200 bước/vòng). Nema 17 thường được sử dụng trong máy in, máy CNC và Máy cắt Laser. Mặc dù Nema 17 không có chức năng phản hồi tín hiệu và độ chính xác thấp hơn những động cơ Servo hoặc động cơ bước có bộ xử lý nhạy khác nhưng giá thành rẻ hơn các loại khác và dễ tìm mua trên thị trường.

BẢNG 9. Thông số kỹ thuật của Nema-17 42MM:

|  |  |
| --- | --- |
| **Thuộc tính:** | **Nema-17 Bipolar 42mm** |
| Loại động cơ: | Lưỡng cực. |
| Góc bước: | 1.8°. |
| Bước chính xác: | 5%. |
| Mô-men xoắn: | 4.1 kg/cm. |
| **Thuộc tính điện:** | |
| Điện áp định mức: | 12 VDC. |
| Dòng điện định mức: |  |
| Trở kháng: | 1.1 Ω. |
| Điện cảm: | 2.6mH ± 20% (1KHz). |
| **Thuộc tính thiết kế:** | |
| Đường kính trục: | 5 mm. |
| Khối lượng: | 300 g. |
| Số chân: |  |

**2.1.7 Động cơ bơm chân không JH12-65 12VDC.**

Động cơ được cấp nguồn để khởi động động cơ DC nhỏ, chuyển động cơ học bên trong là chuyển động lệch tâm, được dẫn động bằng chuyển động lệch tâm bên trong pit tông màng. Buồng bơm cố định thể tích nén khí (cổng hút nén đóng, cổng xả để xả áp suất không khí). Khi ở trạng thái hút khí (cổng xả khí nén đóng, cổng hút áp suất mở ra), cổng hút khí có sự chênh lệch áp suất với môi trường bên ngoài và xả ra khi có tín hiệu xa.

Máy bơm mini được sử dụng phổ biến trong máy lấy mẫu không khí, dụng cụ và thiết bị, ngành y tế và hóa chất, thiết bị gia dụng và các lĩnh vực khác.

BẢNG 11. Thông số kỹ thuật của động cơ bơm chân không JH12-65 12VDC:

|  |  |
| --- | --- |
| **Vật liệu** | Nhựa & Kim loại |
| **Kích thước:** | Xấp xỉ. 6 x 3.8 x 4.5 cm |
| **Điện áp hoạt động:** | DC 9 V ~ 14V |
| **Điện áp định mức:** | 12 VDC. |
| **Dòng điện định mức:** |  |
| **Công suất định mức:** |  |
| **Công suất hút:** | 12L/phút. |
| **Áp suất chân không:** | ~ 80 kpa |
| **Áp suất cực đại:** | 120 kpa. |
| **Tiếng ồn:** | < 65db. |

**2.1.8 Nguồn xung**

Nguồn tổ ong là cách gọi khác của nguồn xung. Cái tên nguồn tổ ong bắt nguồn từ hình dạng các lỗ thông hơi thoát nhiệt của bộ nguồn xung được đục lỗ lục giác giống với cấu tạo của tổ ong nên dân gian gọi vậy cho thân thuộc dễ nhớ.

Nguồn xung là bộ nguồn có tác dụng biến đổi từ nguồn điện xoay chiều sang nguồn điện một chiều bằng chế độ dao động xung tạo bằng mạch điện tử kết hợp với một biến áp xung.

**Hình 2.19. Sơ đồ khối nguồn xung**

**Cấu tạo của nguồn xung:**

**Biến áp xung:** cũng cấu tạo gồm các cuộn dây quán trên một lõi từ giống như biến áp thông thường chỉ có điều biến áp này sử dụng lõi ferit còn biến áp thường sử dụng lỗi thép kỹ thuật điện . Với cùng một kích thước thì biến áp xung cho công suất lớn hơn biến áp thường rất nhiều lần. Ngoài ra biến áp xung hoạt động tốt ở dải tần cao còn biến áp thường chỉ hoạt động ở dải tần thấp.

**Cầu chì:** bảo vệ mạch nguồn bị ngắn mạch.

**Cuộn chống nhiễu, tụ lọc sơ cấp, điode chỉnh lưu:**có nhiệm vụ biến đổi điện áp xoay chiều 220V thành điện áp một chiều tích trữ trên tụ lọc sơ cấp để cung cấp năng lượng cho cuộn sơ cấp của máy biến áp xung.

**Sò công suất:** đây là một linh kiện bán dẫn dùng như một công tắc chuyển mạch , đó có thể là transistor, mosfet, IC tích hợp, IGBT có nhiệm vụ đóng cắt điện từ chân (+) của tụ lọc sơ cấp vào cuộn dây sơ cấp của biến áp xung rồi cho xuống mass.

**Tụ lọc nguồn thứ cấp:** dùng để tích trữ năng lượng điện từ cuộn thứ cấp của biến áp xung để cấp cho tải tiêu thụ. Chúng ta biết rằng khi cuộn sơ cấp của biến áp được đóng cắt điện liên tục bằng sò công suất thì xuất hiện từ trường biến thiên dẫn đến cuộn thứ cấp của biến áp cũng xuất hiện một điện áp ra. Điện áp này được chỉnh lưu qua một vài diode rồi đưa ra tụ lọc (tụ điện) thứ cấp để san phẳng điện áp.

**IC quang và IC TL431:** có nhiệm vụ tạo ra một điện áp cố định để khống chế điện áp ra bên thứ cấp ổn định theo mong muốn. Chúng sẽ làm nhiệm vụ khống chế dao dộng đóng cắt điện vào cuộn sơ cấp của biến áp xung sao cho điện áp ra bên thứ cấp đạt yêu cầu.

**Hình 2.20. Cấu tạo của nguồn xung**

**2.2 Ngôn ngữ Python và thư viện OpenCV**

**2.2.1 Ngôn ngữ Python**

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

**2.2.2   Thư viện OpenCV**

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.

Các tính năng của thư viện OpenCV:

-       Đối với hình ảnh chúng ta có thể đọc và lưu hay ghi chúng.

-       Có thể ghi và đọc video.

-       Xử lý ảnh có thể lọc nhiễu hoặc chuyển đổi ảnh.

-       Thực hiện nhận dạng đặc điểm của hình dạng ảnh

-       Phát hiện các đối tượng được xác định trước như khuôn mặt, mắt, … trong video hoặc hình ảnh.

-       Phân tích video, ước lượng chuyển động của đối tượng xuất hiện trong video, tách nền và đối tượng riêng biệt, …

**2.3  Phương pháp phân loại màu sắc**

**2.3.1.  Hệ màu RGB (Red – Green – Blue)**

Mô hình màu RGB sử dụng mô hình bổ sung trong đó ánh sáng đỏ, xanh lá cây và xanh lam được tổ hợp với nhau theo nhiều phương thức khác nhau để tạo thành các màu khác.

Từ viết tắt RGB trong tiếng Anh có nghĩa:

-       R: Red (màu đỏ).

-       G: Green (màu xanh lá cây).

-       B: (blue (màu xanh lam)

Khi biểu diễn dưới dạng số, các giá trị RGB trong mô hình 24 bpp thông thường được ghi bằng cặp ba số nguyên giữa 0 và 255, mỗi số đại diện cho cường độ của màu đỏ, xanh lá cây, xanh lam trong trật tự như thế.

Ví dụ:

-       (0, 0, 0) là màu đen.

-       (255, 255, 255) là màu trắng.

-       (255, 0, 0) là màu đỏ.

-       (0, 255, 0) là màu xanh lá cây.

-       (0, 0, 255) là màu xanh lam.

-       (255, 255, 0) là màu vàng.

-       (0, 255, 255) là màu xanh ngọc.

-       (255, 0, 255) là màu hồng sẫm.

**Hình 2.21. Mô hình không gian màu RGB**

**2.3.2.  Hệ màu HSV**

Thực chất của không gian này là sự biến đổi của không gian RGB. Không gian HSV được mô tả bằng lệnh lập phương RGB quay trên đỉnh Black. H (Hue) là góc quay trục V (value) qua 2 đỉnh Black và White.

**Trong đó:**

-       H (Hue) chỉ sắc thái có giá trị từ 0o – 360o.

-       S (Saturation) chỉ độ bão hoà. Là mức độ thuần khiến của màu, có thể hiểu là có bao nhiêu màu trắng được thêm vào màu thuần khiết này. Giá trị S nằm trong đoạn [0, 255], trong đó S = 255 là màu tinh khiết nhất, hoàn toàn không pha trắng. Nói cách khác, S càng lớn thì màu càng tinh khiết, đậm.

-       V (Value) Độ sáng của màu. Còn được gọi là Intensity, Lightness, có giá trị dao động trong đoạn [0, 255], trong đó V= 0 là hoàn toàn tối (đen), V = 255 là hoàn toàn sáng. Nói cách khác, V càng lớn thì màu càng sáng.

**Hình 2.22. Mô hình màu HSV**

**Hình 2.23. Bản đồ màu HSV**

Ví dụ: Trích xuất màu xanh lá và màu vàng trong hình dưới đây:

**Hình 2.24. Hoa hướng dương khi chưa tách màu**

Ta sử dụng bản đồ màu HSV trên để xác định vùng màu xanh lá của bông hoa hướng dương trên (36,0,0) ~ (75,255,255) sẻ được kết quả như hình sau:

**Hình 2.25. Tách màu xanh ra khỏi bức hình**

Bản đồ màu trên giúp ta tìm được vùng màu HSV một cách nhanh chóng:

**Hình 2.26. Một số ví dụ khác khi sử dụng bản đồ màu**

**Chương 3.     NỘI DUNG THỰC HIỆN**

**3.1.     Mô hình hệ thống**

**3.1.1  Giới thiệu về hệ thống**

Hệ thống phân loại sẽ thực hiện nhận biết màu sắc của cà chua ở đầu vào sau đó truyền tín hiệu màu đã qua xử lý đến khối xử lý để thực hiện tác vụ phân loại. Trong suốt quá trình phân loại thì sự tác động của các xi lanh, số lượng quả đã được tác động sẽ được thể hiện trên giao diện điều khiển. Dưới đây là mô hình chung của hệ thống:

**Hình 3.1. Mô hình chung của hệ thống**

**3.1.2Chức năng từng bộ phận**

Xử lý màu sắc: xử lý tín hiệu nhận được từ camera và tạo ra tín hiệu đưa về Raspberry để thực hiện các tác vụ khác.

Phân loại cà chua: nhận các tín hiệu từ khối xử lý màu sắc để thực hiện các tác vụ phân loại cà chua dựa trên chương trình được viết theo các yêu cầu được đưa ra.

Giao diện điều khiển: Thực hiện việc điều khiển tắt mở hệ thống và xử lý và hiển thị dữ liệu.

**3.2.     Tính toán và thiết kế hệ thống**

**3.2.1  Thiết kế sơ đồ khối hệ thống**

Với yêu cầu của đề tài, em tiến hành thiết kế sơ đồ với các và chức năng của các khối như sau:

**Hình 3.2. Sơ đồ khối của hệ thống**

**Chức năng các khối:**

-       Khối xử lý màu sắc: có chức năng xử lý tín hiệu từ cảm biến màu sắc và gửi tín hiệu đến khối xử lý trung tâm.

-       Khối cảm biến: cảm biến nhận biết cà chua khi chạy qua băng tải.

-       Khối băng chuyền: có chức năng tải cà chua đến các khu vực khác trong hệ thống.

-       Khối điều khiển khí nén: có chức năng đẩy cà chua được phân loại ra khỏi băng chuyền.

-       Khối xử lý trung tâm: có chức năng nhận, xử lý thông tin và điều khiển các khối khác.

-       Khối nguồn: có chức năng cấp nguồn cho các thiết bị trong hệ thống.

**3.2.2Thiết kế các khối**

**3.2.2.1  Khối cảm biến**

Sử dụng cảm biến vật cản hồng ngoại E18 – D80NK để nhận biết có cà chua đến vị trí xilanh khí nén, cho dừng băng tải, xilanh ở vị trí màu đó hoạt động.

Do mức tín hiệu của cảm biến là 5VDC mà mức điện áp tối đa của chân GPIO của Raspberry chỉ có 3.3VDC nên em sử dụng bộ chuyển đổi 2 mức tín hiệu (Level Converter) để chuyển mức tín hiệu của cảm biến xuống 3.3VDC. Nếu không chuyển mức tính hiệu của cảm biến xuống 3.3VDC sẽ dẫn đến chập, chạm cháy board mạch Raspberry ngay.

**Sơ đồ kết nối cảm biến với Raspberry:**

**Hình 3.3. Sơ đồ kết nối 3 cảm biến với Raspberry**

Dây nguồn (màu nâu) của cảm biến được nối với nguồn 5VDC, dây GND (màu xanh) được nối với 0VDC. Dây tín hiệu (màu đen) được nối vào bộ chuyển đổi (Level Converter) theo thứ tự là: CB1 nối HV1, CB2 nối HV2, CB3 nối HV3.

Chân HV của bộ chuyển đổi nối nguồn 5VDC chân LV nối nguồn 3.3VDC của Raspberry, 2 chân GND được nối chung giữa nguồn 5VDC và chân GND của Raspberry.

Các tín hiệu của cảm biến đã được chuyển đổi từ 5VDC xuống 3.3VDC ở các chân đối diện: HV1 – LV1, HV2 – LV2, HV3 – LV3. Nối dây từ các chân LV vào Raspberry để thực hiện viết chương trình.

**3.2.2.2  Khối băng chuyền**

Khối băng chuyền sẽ thực hiện việc chuyển cà chua đến vị trí thực hiện phân loại (camera Rasperry) và vị trí của cảm biến vật cản. Do là mô hình nhỏ nên sử dụng động cơ giảm tốc 37GB520R với điện áp 12VDC.

Băng tải được nối với 1 relay trong module relay 4 kênh để phục vụ cho việc dừng và chạy tiếp: Dây nối từ nguồn 12VDC chân COM của relay, chân NO được nối với chân (+) của băng tải, chân (-) của băng tải được nối với 0VDC.

**Hình 3.4. Sơ đồ kết nối của băng tải**

**3.2.2.3  Khối hệ thống điều khiển khí nén**

Hệ thống điều khiển khí nén sẽ thực hiện nhiệm vụ đưa cà chua ra khỏi băng tải để hoàn thành việc phân loại. Hệ thống sẽ gồm 2 phần chính là xi lanh và van điện từ.

Để điều khiển được hành trình của xilanh thì ta sẽ điều khiển các cuộn dây của van điện từ thông qua relay và Raspberry. Dưới đây là sơ đồ kết nối của van với relay và Raspberry:

**Hình 3.5 Sơ đồ kết nối của 3 van khí nén với module relay 4 kênh và Raspberry**

Chân (+) của van được nối vào chân NO của relay, chân COM của relay nối vào V+ của nguồn 24VDC.

Chân (-) của van được nối thẳng vào V- của nguồn 24VDC.

Module relay được cấp nguồn 5VDC nối đất chung với Raspberry.

Chân kích của các relay (IN1, IN2, …) được nối vào Raspberry để viết chương trình.

**3.2.2.4 Khối xử lý trung tâm**

Ở đây khối xử lý trung tâm được sử dụng là Rasperry Pi 3 Model B+.

Khối xử lý trung tâm thực hiện việc lọc, phân loại các khoảng màu chín, gần chín, còn xanh của cà chua thông qua hình ảnh lấy được từ camera bằng chương trình Python đã được viết.

Raspberry nhận tín hiệu từ các cảm biến hồng ngoại để xác định vị trí của cà chua và thực hiện việc điều khiển các van điện từ để đẩy cà chua được phân loại về đúng vị trí được chọn.

**Hình 3.6. Camera kết nối với Raspberry Pi 3 Model B+**

**3.2.2.5  Khối nguồn**

Đối với khối nguồn do ở đây chúng ta vừa cần dùng nguồn AC và DC nên nguồn AC 220V chúng ta sẽ lấy trực tiếp từ lưới điện để cấp cho nguồn xung.

Nguồn DC để cấp cho relay, van điện từ, được lấy từ nguồn tổ ong 24VDC và 5VDC, riêng băng tải sử dụng mạch giảm áp LM2596 để giảm từ 24VDC xuống 12VDC.

**Hình 3.7. Mạch giảm áp LM2596**

**3.2.2.6  Khối xử lý màu sắc**

Sử dụng Camera Raspberry pi V1 5MP để thu nhận hình ảnh từ buồng sáng.

Khi ta chiếu ánh sáng trắng vào một vật thể bất kì, tại bề mặt vật thể sẽ xảy ra hiện tượng hấp thụ và phản xạ ánh sáng. Dựa trên nguyên lý sự phản xạ, hấp thụ ánh sáng trắng của vật thể và sự phối trộn màu sắc bởi 3 màu cơ bản Blue, Green, Red. Ta chuyển đổi hình ảnh nhận được từ hệ màu RGB sang HSV dựa vào thư viện OpenCV trong ngôn ngữ Python bằng câu lệnh:

hsv\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

**Cách chuyển đổi RGB sang HSV**

Các giá trị màu của hệ màu RGB được chia cho 255 để chuyển đổi giá trị từ 0…255 cho 0…1:

                                                               (2.1)

                                                                (2.2)

                                                              (2.3)

                                           (2.4)

                                            (2.5)

                                                    (2.6)

Trong đó:

R’, G’, B’ là giá trị được chuyển đổi từ 0…255 sang 0…1.

Cmax, Cmin là giá trị lớn nhất, nhỏ nhất được chọn từ R’, G’, B'.

Giá trị H trong hệ màu HSV được tính theo công thức sau:

                        (2.7)

Giá trị S được tính dựa trên công thức:

                                             (2.8)

Giá trị V được chọn theo Cmax:

                                                                 (2.9)

Các công thức được sử dụng tương ứng với giá trị Cmax tìm được ở (2.4).

**Ví dụ:**

Chuyển màu đỏ từ hệ màu RGB sang HSV

**Hình 3.8. Xác định hệ màu RGB thông qua ứng dụng Paint**

Màu đỏ trong hệ màu RGB (255,0,0)

Ta thấy

Vậy màu đỏ được chuyển từ hệ màu RGB sang HSV có kết quả là (0, 1, 1) tức là (0, 255, 255) đem kết quả so sánh với hình 2.20 ta thấy đúng.

**3.3.         Thực hiện phần mềm**

**3.3.1     Cài đặt hệ điều hành cho Raspberry**

**Bước 1:**Tải hệ điều hành và các phân mềm cần thiết

-       Tải hệ điều hành Raspian mới nhất về máy.

-       Tải SD Memory Card Formatter mới nhất để format thẻ nhớ.

-       Tải Win32DiskImager mới nhất để chép Raspbian vào thẻ nhớ.

-       Tải về Wireless Network Watcher để tìm IP của Pi.

-       Tải Putty (SSH – telnet client)

**Bước 2:** Format thẻ nhớ

Lưu ý: cần chọn đúng ổ đĩa để format, việc chọn nhầm ổ đĩa sẻ dẫn đến việc mất toàn bộ dữ liệu có trong ổ đĩa đó.

**Hình 3.9. Format thẻ nhớ**

**Bước 3:**Mở Win32DiskImager ra, chọn đến file chứa hệ điều hành Raspbian .img đã tải về ở bước trên, chọn ổ đĩa thẻ nhớ ở ô Device, sau đó bấm Write để ghi vào thẻ nhớ.

**Hình 3.10. Ghi file hệ điều hành vào thẻ nhớ SD**

**Bước 4:**Mở putty và đăng nhập vào Raspberry (dựa trên ip đã quét).

**Hình 3.11. Mở putty và nhập ip để đăng nhập**

**Bước 5:**Gõ các dòng lệnh:

$ sudo apt-get install xrdp (để cài đặt XRDP)

$ sudo apt-get update

$ sudo apt-get upgrade

Nếu muốn kết nối với màn hình laptop cần cài screen và htop:

$ sudo apt-get install screen

$ sudo apt-get install htop

Như vậy là chúng ta đã hoàn thành quá trình cài hệ điều hành cho Raspberry, chúng ta có thể kết nối với màn hình laptop thông qua remote desktop hay một số phần mềm khác.

**3.3.2Cài đặt thư viện OpenCV cho hệ điều hành Raspian Buster**

**Bước 1:** Mở rộng hệ thống và lấy lại không gian

Nhập lệnh ($ sudo raspi-config)   chọn Advanced Options  chọn A1 Expand filesystem   chọn Finish   nhập lệnh ($ sudo reboot).

**Hình 3.12. Chọn Advanced Options để có thể mở rộng bộ nhớ**

Mở rộng không gian thẻ nhớ bằng cách nhập các lệnh:

$ sudo apt-get purge wolfram-engine

$ sudo apt-get purge libreoffice\*

$ sudo apt-get clean

$ sudo apt-get autoremove

**Bước 2:**Cài đặt các gói phụ thuộc

Cài đặt các gói của nhà phát triển: $sudo apt-get install build-essential cmake pkg-config

Chúng ta cần cài đặt một số gói I/O hình ảnh cho phép chúng ta tải các định dạng tệp hình ảnh khác nhau từ đĩa:

$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev

Gói I/O video:

$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

$ sudo apt-get install libxvidcore-dev libx264-dev

Tiếp theo đó chúng ta cũng cần cài đặt thư viện phát triển GTK để chúng ta có thể biên soạn modun phụ của OpenCV, cho phép chúng ta hiển thị hình ảnh trên màn hình và xây dựng các GUI đơn giản:

$ sudo apt-get install libfontconfig1-dev libcairo2-dev

$ sudo apt-get install libgdk-pixbuf2.0-dev libpango1.0-dev

$ sudo apt-get install libgtk2.0-dev libgtk-3-dev

Cài đặt gói tối ưu hóa (cải thiện ma trận cho OpenCV):

  $ sudo apt-get install libatlas-base-dev gfortran

Bộ dữ liệu HDF5:

  $ sudo apt-get install libhdf5-dev libhdf5-serial-dev libhdf5-103

$ sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5

Cài Python 3:

  $ sudo apt-get install python3-dev

Cài numpy:

  $ wget https://bootstrap.pypa.io/get-pip.py

$ sudo python get-pip.py

$ sudo python3 get-pip.py

$ sudo rm -rf ~/.cache/pip

Cài PiCamera API:

  $ pip install "picamera[array]"

**Bước 3:**Tải OpenCV và OpenCV contrib:

  $ cd ~

$ wget -O opencv.zip https://github.com/opencv/opencv/archive/4.1.1.zip

$ wget -O opencv\_contrib.zip

$ unzip opencv.zip

$ unzip opencv\_contrib.zip

$ mv opencv-4.1.1 opencv

$ mv opencv\_contrib-4.1.1 opencv\_contrib

**Bước 4:**Thay đổi kích thước không gian (100  2048)

  $ sudo nano /etc/dphys-swapfile

  $ sudo /etc/init.d/dphys-swapfile stop

$ sudo /etc/init.d/dphys-swapfile start

**Bước 5:**Biên dịch và cài đăt Opencv 4

  $ cd ~/opencv

$ mkdir build

$ cd build

$ cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

    -D CMAKE\_INSTALL\_PREFIX=/usr/local \

    -D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_contrib/modules \

    -D ENABLE\_NEON=ON \

    -D ENABLE\_VFPV3=ON \

    -D BUILD\_TESTS=OFF \

    -D INSTALL\_PYTHON\_EXAMPLES=OFF \

    -D OPENCV\_ENABLE\_NONFREE=ON \

    -D CMAKE\_SHARED\_LINKER\_FLAGS=-latomic \

    -D BUILD\_EXAMPLES=OFF ..

**Hình 3.13. Sau khi biên dịch xong**

Đợi biên dịch xong và nhấn Make để chạy trình cài đặt.

**Bước 6:**Chuyển lại kích thước không gian (2048  100)

**Bước 7:**Cài đặt OpenCV vào hệ thống

  $ sudo make install

$ sudo ldconfig

**Bước 8:** Kiểm tra

$ cd ~

$ workon cv

$ python

>>> import cv2

>>> cv2.\_\_version\_\_

Nếu hiện bản 4.1.1 là ta đã cài thành công.

**3.3.3   Phần mềm lập trình ngôn ngữ Python**

Trong đề tài phần mềm được sử dụng để viết chương trình là Python IDE.

IDE là viết tắt của Integrated Development Environment (môi trường phát triển tích hợp) được định nghĩa là một công cụ mã hóa giúp tự động hóa quá trình chỉnh sửa, biên dịch, kiểm thử mã nguồn và nó giúp nhà phát triển dễ dàng chạy, viết và debug code.

Hướng dẫn sử dụng Python IDE:

Bước 1: Khởi động raspberry  nhấn vào biểu tượng có hình logo của Raspberry  chọn lập trình  chọn Thonny Python IDE.

**Hình 3.14. Khởi động Python IDE**

Bước 2: Click chọn file new để viết một chương trình mới hoặc file open để mở một chương trình có sẵn.

**Hình 3.15. Chương trình đã viết**

Bước 3: Click vào nút play màu xanh lá hoặc nhập lệnh %run “tên file” để chạy một chương trình đã viết.

Nếu không quen với giao diện của Python IDE ta có thể tải phần mềm Sublime text để viết chương trình. Đối với phần mềm Sublime text ta cần lưu file với tên.py để tạo một file Python, muốn chạy một file Python trong Sublime text ta cần chạy file đó trên terminal.

**3.4.     Nguyên lý hoạt động của hệ thống**

Raspberry đóng vai trò là khối xử lý trung tâm nhận hình ảnh từ camera và xử lý hình ảnh theo chương trình được viết với ngôn ngữ Python. Chương trình dựa vào thư viện OpenCV để phân biệt được màu sắc khác nhau.

Sau khi phân loại được màu sắc, Raspberry thu nhận tín hiệu từ cảm biến hồng ngoài và xuất các tín hiệu khiến van điện từ điều khiển các xilanh khí nén đẩy cà chua vào các máng theo màu sắc.

**3.5.     Lưu đồ giải thuật**

Lưu đồ là một loại sơ đồ biểu diễn một thuật toán hoặc một quá trình, biểu hiện các bước công việc dưới dạng các loại hình hộp khác nhau theo thứ tự được biểu diễn bởi các mũi tên. Sơ đồ này có thể thể hiện giải pháp cho vấn đề cần giải quyết từng bước từng bước một.

**3.5.1Lưu đồ giải thuật cho chương trình chính**

**Hình 3.16.Lưu đồ giải thuật chương trình chính của hệ thống**

Ở chương trình chính sau khi hệ thống hoạt động sẽ kiểm tra màu nhận được từ quả cà chua được cấp. Cà chua được băng tải vận chuyển đi qua camera và các cảm biến hồng ngoại và xi lanh sẽ được tác động để thực hiện phân loại. Sau đó xilanh cấp phôi sẽ được khởi động lại để đẩy quả cà chua tiếp theo.

**3.5.2Lưu đồ giải thuật của khối hình ảnh**

**Hình 3.17. Lưu đồ giải thuật của khối xử lý hình ảnh**

Chương trình xử lý màu sắc sẽ thực hiện kiểm tra màu mà camera nhận được với mỗi màu sẽ xuất tín hiệu ra chân tín hiệu tương ứng với màu đó.

**Chương 4.     KẾT QUẢ THỰC HIỆN**

**4.1.     Kết quả**

**4.1.1  Kết quả nghiên cứu**

Trong quá trình thực hiện khóa luận tốt nghiệp em đã nghiên cứu và đạt được các kết quả sau:

-      Lập trình giao tiếp Raspberry với các thiết bị ngoại vi (camera, cảm biến hồng ngoại, relay, …).

-      Lập trình hệ thống sử dụng ngôn ngữ lập trình Python.

-      Sử dụng các phần mềm như: Sublime Text, Python IDE, Microsoft Visio trong quá trình thực hiện đề tài.

**4.1.2Kết quả thi công**

**4.1.2.1  Kết quả phần cứng**

Qua quá trình thiết kế phần cứng, chọn lựa linh kiện, em đã tiến hành kết nối các module, linh kiện với nhau và cho ra được mô hình phân loại cà chua theo màu sắc sử dụng Raspberry Pi 3 Module B+.

**Hình 4.1. Mô hình phân loại cà chua dùng công nghệ xử lý ảnh**

**4.1.2.2  Các bước thực nghiệm mô hình**

Phân biệt được 3 màu cà chua là chín, gần chín và xanh được hiển thị và khoanh vùng màu trên hai khung label color và mask.

**Bước 1:**kiểm tra bằng cà chua chín

**Hình 4.2. Kết quả phân loại cà chua chín**

**Bước 2:**kiểm tra bằng cà chua gần chín

**Hình 4.3. Kết quả phân loại cà chua gần chín**

**Bước 3:**kiểm tra bằng cà chua xanh

**Hình 4.4. Kết quả phân loại cà chua xanh**

**4.1.2.3 Kết quả thực nghiệm**

Chạy chương trình ở điều kiện ánh sáng tốt không bị ngược sáng, mẫu vật không bị trói sáng, bị che lấp. Đây là môi trường lý tưởng giúp chương trình cho độ chính xác cao. Tuy nhiên khi chạy mô hình trong thực tế có sự ảnh hưởng của các yếu tố bên ngoài như: ánh sáng chiếu vào camera, mẫu vật bị che lấp, … sẽ làm cho độ chính xác của chương trình bị giảm đi tương đối.

So sánh chương trình được chạy ở điều kiện lý tưởng và chương trình chạy ở điều kiện không lý tưởng:

-         Khi điều kiện ánh sáng phù hợp, mẫu vật không bị che lấp chương trình cho kết quả chính xác của 3 màu sắc: cà chua chín, gần chín, còn xanh.

-         Khí điều kiện ánh sáng quá chói, mẫu vật bị che lấp hoặc bị lệch chương trình cho kết quả không được chính xác.

Cách giải quyết:

Làm một buồng camrera gắn led trắng để cung cấp ánh sáng. Buồng sáng giúp cố định môi trường xung quanh camera giúp camera luôn có được hình ảnh chính xác giúp mô hình chạy với hiệu suất 90% trong bất kỳ điều kiện môi trường ánh sáng nào.

**Hình 4.5. Buồng camera được gắn led trắng**

**Bảng 4.1. Bảng kết quả thực nghiệm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Số lần   Màu sắc | Tổng số lần thực nghiệm | Số lần phân loại đúng | Số lần phân loại sai | Hiệu suất |
| Màu chín | 50 | 46 | 4 | 92% |
| Màu gần chín | 50 | 43 | 7 | 86% |
| Còn xanh | 50 | 50 | 0 | 100% |

**4.2.     Biện luận**

Về cơ bản chương trình đã thực hiện được các yêu cầu đã đề ra là phân loại trái cây (cà chua) theo 3 cấp bậc là chín, gần chín và còn xanh. Chúng ta có thể phân loại các loại trái cây khác thông qua việc thay đổi vùng màu.

Đây là đề tài xử lý ảnh nền tảng giúp ta đặt bước đi đầu tiên trong việc tìm hiểu về xử lý ảnh ở các cập bậc cao hơn.

Đề tài được ứng dụng một cách rộng rãi, việc thực hiện đề tài này giúp ta có nhiều kinh nghiệm hơn trong lĩnh vực xử lý ảnh phân loại sản phẩm giúp cho việc làm sau này.

**Chương 5.     KẾT LUẬN**

**5.1.     Kết luận**

Đây là một đề tài hấp dẫn, đã nhận được nhiều sự quan tâm của các nhà nghiên cứu vì được lợi ích to lớn trong việc phân loại sản phẩm. Khóa luận đã trình bày tổng quan được về phương pháp xử lý ảnh bằng Raspberry kết hợp với ngôn ngữ Python và thư viên OpenCV.

Phương pháp xử lý ảnh trực tiếp bằng hình anh thu được từ camera mang lại kết quả tốt khi so sánh độ chính xác với các phương pháp sử dụng cảm biến màu TCS 3200,… So với phương pháp thực hiện thủ công thì đó là một bước tiến rất lớn, tốc độ cho kết quả cao đặc biệt khi xử lý số lượng lớn các vật có màu và hình dạng khác nhau, không thể kiểm định chính xác bằng mắt thường. Kết quả cho thấy điều kiện ánh sáng có ảnh hưởng lớn đến việc phân loại vì vậy cần phải có nguồn sáng ổn định.

**5.2.     Những thuận lợi và khó khăn khi thực hiện**

**5.2.1  Thuận lợi**

-         Tài liệu nghiên cứu có sẵn trên mạng.

-         Các thiết bị dễ tìm.

-         Cách kết nối giữa các thiết bị với Raspberry có thể tham khảo trên mạng.

-         Các lỗi trong quá trình thực hiện đều được giải đáp trên các trang web tiếng anh (github.com, stackoverflow, …).

**5.2.2Khó khăn**

-         Là lần đầu sử dụng Raspberry.

-         Lần đầu tiếp xúc với ngôn ngữ lập trình Python.

-         Vùng màu của trái cây thực tế khó xác định.

-         Có ít tài liệu tiếng việt.

-         Mức điện áp tối đa (3.3V) của chân GPIO gây khó khan trong việc nhận tín hiệu từ các thiết bị.

**5.3.     Các hạn chế của mô hình**

-         Khâu cấp phôi chưa được tối ưu.

-         Chưa thực hiện chính xác 100%.

-         Cần kết nối với laptop để khởi động chương trình.

-         Tốc độ phân loại chậm.

**5.4.     Hướng phát triển hoàn thiện đề tài**

-         Có giao diện để điều chỉnh vùng màu phù hợp với từng loại trái cây.

-         Thiết kế giao diện điều khiển từ xa.

-         Mở rộng thêm các khâu như đóng gói sản phẩm.

-         Tăng hiệu suất cũng như tốc độ của hệ thống.

-         Mở rộng phân loại nhiều loại trái cây.

-         Có thể thêm kích thước cân năng để phân loại.

-         Thêm màn hình LCD để khởi động và chạy trưc tiếp trên màn hình.

**TÀI LIỆU THAM KHẢO**

**Tiếng việt:**

[1] Trần Khánh Dưng và Hoàng Như Ý, “*Hệ thống phân loại cà chua theo màu sắc xử dụng PLC S7 – 1200”,*Trường Đại Học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh, 2018.

[2] Lại Tuấn Dũng, “*Lập trình cơ bản Python cho người mới”,*Trường Đại Học Swinburne of Technology, 2019.

[3] Nguyễn Quang Hoan,“*Xử lý ảnh*”, Học viện công nghệ bưu chính viễn thông, 2006.

**Tiếng anh:**

[1] Minichino Joe, “*Learning OpenCV 3 Computer Vision with Python*”, ebay England production, 2015.

[2] Sandipan Dey, “*Hands-On Image Processing with Python: Expert techniques for advanced image analysis and effective interpretation of image*”, Paperback England, 2018.

[3] Ashwin Pajankar, “*Python 3 Image Processing: Learn Image Processing with Python 3, NumPy, Matplotlib, and Scikit-image*”, Kindle editon Nashik India, 2016.

[4] Ray Lawnthorn, “*Color Detection for Robots*”, Kindle edition England, 2012.

**PHỤ LỤC**

import cv2

import numpy as np

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

#Chan 19 R1 kich bang tai

GPIO.setup(19, GPIO.OUT)

#chan 13 R2 kich mau xanh

GPIO.setup(13, GPIO.OUT)

#chan 6 R3 kich mau do

GPIO.setup(6, GPIO.OUT)

#chan 26 R4 kich cap phoi

GPIO.setup(26, GPIO.OUT)

#chan 8 = sensor 1

GPIO.setup(8, GPIO.IN)

#chan 25 = sensor 2

GPIO.setup(25, GPIO.IN)

#chan 7 = sensor 3

GPIO.setup(7, GPIO.IN)

b = 0

GPIO.output(19,1)

GPIO.output(13,1)

from picamera.array import PiRGBArray

from picamera import PiCamera

def nothing(x):

    pass

camera = PiCamera()

camera.resolution = (480, 368)

camera.framerate = 24

rawCapture = PiRGBArray(camera, size=(480, 368))

for frame in camera.capture\_continuous(rawCapture, format="bgr", use\_video\_port=True):

    frame = frame.array

    hsv\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

    #vung mau

    low\_red = np.array([170, 155, 180])

    high\_red = np.array([5, 255, 255])

    red\_mask = cv2.inRange(hsv\_frame, low\_red, high\_red)

    #red = cv2.bitwise\_and(frame, frame, mask=red\_mask)

    (contours,hierarchy)=cv2.findContours(red\_mask,cv2.RETR\_TREE,cv2.CHAIN\_APPROX\_SIMPLE)

    for piv, contour in enumerate(contours):

        area = cv2.contourArea(contour)

        if(area>300):

            x,y,w,h = cv2.boundingRect(contour)

            img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),1)

            cv2.putText(frame,"Mau do",(x,y),cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0,0,255))

            b = 1

    #vung mau

    low\_yellow = np.array([25, 100, 180])

    high\_yellow = np.array([44, 255, 255])

    yellow\_mask = cv2.inRange(hsv\_frame, low\_yellow, high\_yellow)

    yellow = cv2.bitwise\_and(frame, frame, mask=yellow\_mask)

        (contours,hierarchy)=cv2.findContours(yellow\_mask,cv2.RETR\_TREE,cv2.CHAIN\_APPROX\_SIMPLE)

    for piv, contour in enumerate(contours):

        area = cv2.contourArea(contour)

        if(area>300):

            x,y,w,h = cv2.boundingRect(contour)

            img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),1)

            cv2.putText(frame,"Mau vang",(x,y),cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0,0,255))

            b = 3

    # vung mau

    low\_green = np.array([45, 40,100])

    high\_green = np.array([65, 255, 255])

    #hien mask

    green\_mask = cv2.inRange(hsv\_frame, low\_green, high\_green)

    #green = cv2.bitwise\_and(frame, frame, mask=green\_mask)

    (contours,hierarchy)=cv2.findContours(green\_mask,cv2.RETR\_TREE,cv2.CHAIN\_APPROX\_SIMPLE)

    for piv, contour in enumerate(contours):

        area = cv2.contourArea(contour)

        if(area>300):

            x,y,w,h = cv2.boundingRect(contour)

            img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),1)

            cv2.putText(frame,"Mau xanh",(x,y),cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0,0,255))

            b = 2

#maudo

    if b == 1 :

        cb1 = GPIO.input(8)

        if cb1 == 0:

            GPIO.output(19, 0)

            GPIO.output(6, 1)

            GPIO.output(26, 0)

            print("da day sp mau do")

        elif cb1 == 1:

            GPIO.output(19, 1)

            GPIO.output(6, 0)

            GPIO.output(26, 1)

            print("sp mau do")

 #mau xanh

    elif b == 2:

           cb1 = GPIO.input(8)

        cb2 = GPIO.input(25)

        if cb1 == 1:

           GPIO.output(26, 0)

        if cb2 == 1:

            GPIO.output(19, 0)

            GPIO.output(13, 1)

            print ("da day sp mau xanh")

        elif cb2 == 0:

            GPIO.output(26, 1)

            GPIO.output(19, 1)

            GPIO.output(13, 0)

#mauvang

    elif b == 3:

           cb1 = GPIO.input(8)

           cb3 = GPIO.input(7)

           if cb1 == 1:

                     GPIO.output(26,0)

           if cb3 == 1:

                     GPIO.output(26,1)

    thresh = red\_mask + green\_mask

    kernel = np.ones((5,5),np.uint8)

    mask = cv2.morphologyEx(thresh, cv2.MORPH\_OPEN, kernel)

    mask = cv2.morphologyEx(mask, cv2.MORPH\_CLOSE, kernel)

    mau = cv2.bitwise\_and(frame, frame, mask=mask)

    cv2.imshow("color tracking", frame)

    cv2.imshow("mask", mask)

    key = cv2.waitKey(1)

    rawCapture.truncate(0)

    if key == 27:

        break

**KHÓA LUẬN TỐT NGHIỆP**

**ĐỀ TÀI: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG PHÂN LOẠI SẢN PHẨM THEO MÀU SẮC SỬ DỤNG CÔNG NGHỆ XỬ LÝ ẢNH DỰA TRÊN NỀN TẢNG RASPBERRY**

**TÓM TẮT ĐỀ TÀI**

Cùng với sự phát triển của thế giới và xu hướng hội nhập kinh tế quốc tế, đất nước ta đang đổi mới và bước vào thời kì công nghiệp hóa, hiện đại hóa, vừa xây dựng cơ sở vật chất kỹ thuật hiện đại, vừa phát triển nền kinh tế đất nước. Từ đó, chúng ta đòi hỏi phải nghiên cứu và áp dụng những dây chuyền, máy móc và thiết bị tiên tiến hiện đại, cókhả năng tự động hóa cao để đưa công nghệ vào mọi lĩnh vực của cuộc sống. Trong đó ngành công nghiệp tự động hóa đóng một vai trò rất quan trọng trong quá trình phát triển của đất nước. Để đáp ứng nhu cầu to lớn của việc phát triển ngành Tự động hóa nói chung, đòi hỏi phải có đội ngũ cán bộ, nhân viên kỹ thuật có khả năng, đủ năng lực và trình độ chuyên môn để kịp thời giải quyết mọi vấn đề liên quan đến kỹ thuật cơ khí, điện-điện tử và kỹ thuật phần mềm.

Từ những thực tế nói trên, bản thân là sinh viên của ngành **Điện Tử,**từ những kiến thức được học, em đã lựa chọn đề tài: **“Thiết kế và thi công hệ thống phân loại sản phẩm theo màu sắc sử dụng công nghệ xử lý ảnh dựa trên nền tảng Raspberry”.**Việc tạo ra một hệ thống như vậy để thay thế con người trong công việc là vấn đề hết sức cần thiết.

Luận văn này được chia làm 8 chương chính:

Chương 1: Giới thiệu đề tài.

Chương 2: Cơ sở lý thuyết

Chương 3: Nội dung thực hiện

Chương 4: Kết quả thực hiện

Chương 5: Kết luận

**MỤC LỤC**

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.. i

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN.. ii

LỜI CẢM ƠN   iii

TÓM TẮT ĐỀ TÀIiv

MỤC LỤC.. v

MỤC LỤC HÌNH ẢNH.. viii

DANH MỤC BẢNG BIỂU.. xi

DANH MỤC VIẾT TẮT. xii

Chương 1.     GIỚI THIỆU ĐỀ TÀI1

1.1        Đặt vấn đề. 1

1.2        Các nghiên cứu liên quan trong và ngoài nước. 1

1.2.1           Bài báo “*Detecting colors (Hsv Color Space) – Opencv with Python*”. 1

1.2.2           Bài báo “*Adruino Color Detection*”. 2

1.2.3           Bài báo “*Hệ thống phân loại theo màu sắc xử dụng PLC S7 - 1200*”. 3

1.3        Mục tiêu. 3

1.4        Phạm vi của đề tài4

Chương 2.     CƠ SỞ LÝ THUYẾT. 5

2.1.       Các linh kiện được xử dụng trong đề tài5

2.1.1           Kit Raspberry Pi 3 model B. 5

2.1.1.1        Giới thiệu về Raspberry Pi và Raspberry pi 3 Model B  5

2.1.1.2        Thông số kỹ thuật của Raspberry Pi 3 Model B. 6

2.1.1.3        Những lưu ý khi xử dụng Raspberry. 7

2.1.2           Module Camera Raspberry V1 5MP. 8

2.1.3           Băng tải9

2.1.4           Xilanh khí nén và Van điện từ khí nén. 10

2.1.4.1        Xilanh khí nén. 10

2.1.4.2        Van điện từ khí nén. 11

2.1.5           Module relay 5V 4 kênh. 14

2.1.6     Bộ chuyển đổi mức logic hai hướng 4 kênh. 15

2.1.7     Cảm biến hồng ngoại E18 – D80NK.. 16

2.1.8     Nguồn xung. 18

2.2.       Ngôn ngữ Python và thư viện OpenCV.. 20

2.2.1     Ngôn ngữ Python. 20

2.2.2     Thư viện OpenCV.. 21

2.3.       Phương pháp phân loại màu sắc. 21

2.3.1.    Hệ màu RGB (Red – Green – Blue)21

2.3.2.    Hệ màu HSV.. 22

Chương 3..... NỘI DUNG THỰC HIỆN.. 25

3.1.       Mô hình hệ thống. 25

3.1.1     Giới thiệu về hệ thống. 25

3.1.2     Chức năng từng bộ phận. 25

3.2.       Tính toán và thiết kế hệ thống. 25

3.2.1     Thiết kế sơ đồ khối hệ thống. 25

3.2.2     Thiết kế các khối26

3.2.2.1  Khối cảm biến. 26

3.2.2.2  Khối băng chuyền. 27

3.2.2.3  Khối hệ thống điều khiển khí nén. 28

3.2.2.4 Khối xử lý trung tâm.. 29

3.2.2.5  Khối nguồn. 29

3.2.2.6  Khối xử lý màu sắc. 30

3.3.       Thực hiện phần mềm.. 32

3.3.1     Cài đặt hệ điều hành cho Raspberry. 32

3.3.2     Cài đặt thư viện OpenCV cho hệ điều hành Raspian Buster35

3.3.3     Phần mềm lập trình ngôn ngữ Python. 39

3.4.       Nguyên lý hoạt động của hệ thống. 40

3.5.       Lưu đồ giải thuật40

3.5.1     Lưu đồ giải thuật cho chương trình chính. 41

3.5.2     Lưu đồ giải thuật của khối hình ảnh. 42

Chương 4..... KẾT QUẢ THỰC HIỆN.. 43

4.1.       Kết quả. 43

4.1.1     Kết quả nghiên cứu. 43

4.1.2     Kết quả thi công. 43

4.1.2.1  Kết quả phần cứng. 43

4.1.2.2  Các bước thực nghiệm mô hình. 44

4.1.2.3 Kết quả thực nghiệm.. 45

4.2.       Biện luận. 46

Chương 5..... KẾT LUẬN.. 48

5.1.       Kết luận. 48

5.2.       Những thuận lợi và khó khăn khi thực hiện. 48

5.2.1     Thuận lợi48

5.2.2     Khó khăn. 48

5.3.       Các hạn chế của mô hình. 48

5.4.       Hướng phát triển hoàn thiện đề tài49

TÀI LIỆU THAM KHẢO.. 50

PHỤ LỤC.. 51

MỤC LỤC HÌNH ẢNH

Hình 1.1. Tách màu đỏ từ môi trường. 2

Hình 1.2. Phát hiện màu vàng. 2

Hình 1.3. Phát hiện màu đỏ. 3

Hình 1.4. Kết quả thi công phần cứng. 3

Hình 2.1. Raspberry Pi 3 Model B. 5

Hình 2.2. Thông số phần cứng của Raspberry Pi 3 Model B. 6

Hình 2.3. Sơ đồ chân của Raspberry Pi 3 Model B. 7

Hình 2.4. Camera Raspberry V1 5MP. 8

Hình 2.5 Camera được gắn vào kit Raspberry. 8

Hình 2.6. Băng tải mini sử dụng cho các mô hình. 9

Hình 2.7. Động cơ 37GB520R. 10

Hình 2.8. Xilanh SMC SDJ2D16 – 100B. 11

Hình 2.9. Cổng P của van 5/2. 12

Hình 2.10. Hai cổng A và B của van 5/2. 13

Hình 2.11. Hai cổng xả R vs S của van 5/2. 13

Hình 2.12. Module Relay 5VDC 4 kênh. 14

Hình 2.13. Sơ đồ nguyên lý của Module Relay. 14

Hình 2.14. Bộ chuyển đổi mức tín hiệu. 15

Hình 2.15. Sơ đồ nguyên lý của Level Coverter MH.. 15

Hình 2.16. Cảm biến hồng ngoại E18 – D80NK.. 16

Hình 2.17. Sơ đồ nguyên lý của cảm biến hồng ngoại17

Hình 2.18. Mô tả hoạt động của cảm biến hồng ngoại18

Hình 2.19. Sơ đồ khối nguồn xung. 19

Hình 2.20. Cấu tạo của nguồn tổ ong. 20

Hình 2.21. Mô hình không gian màu RGB. 22

Hình 2.22. Mô hình màu HSV.. 23

Hình 2.23. Bản đồ màu HSV.. 23

Hình 2.24. Hoa hướng dương khi chưa tách màu. 24

Hình 2.25. Tách màu xanh ra khỏi bức hình. 24

Hình 2.26. Một số ví dụ khác khi sử dụng bản đồ màu. 24

Hình 3.1. Mô hình chung của hệ thống. 25

Hình 3.2. Sơ đồ khối của hệ thống. 26

Hình 3.3. Sơ đồ kết nối 3 cảm biến với Raspberry. 27

Hình 3.4. Sơ đồ kết nối của băng tải28

Hình 3.5 Sơ đồ kết nối của 3 van khí nén với module relay 4 kênh và Raspberry. 28

Hình 3.6. Camera kết nối với Raspberry Pi 3 Model B+. 29

Hình 3.7. Mạch giảm áp LM2596. 30

Hình 3.8. Xác định hệ màu RGB thông qua ứng dụng Paint31

Hình 3.9. Format thẻ nhớ. 33

Hình 3.10. Ghi file hệ điều hành vào thẻ nhớ SD.. 33

Hình 3.11. Mở putty và nhập ip để đăng nhập. 34

Hình 3.12. Chọn Advanced Options để có thể mở rộng bộ nhớ. 35

Hình 3.13. Sau khi biên dịch xong. 38

Hình 3.14. Khởi động Python IDE. 39

Hình 3.15. Chương trình đã viết40

Hình 3.16. Lưu đồ giải thuật chương trình chính của hệ thống. 41

Hình 3.17. Lưu đồ giải thuật của khối xử lý hình ảnh. 42

Hình 4.1. Mô hình phân loại cà chua dùng công nghệ xử lý ảnh. 43

Hình 4.2. Kết quả phân loại cà chua chín. 44

Hình 4.3. Kết quả phân loại cà chua gần chín. 44

Hình 4.4. Kết quả phân loại cà chua xanh. 45

Hình 4.5. Buồng camera được gắn led trắng. 46

DANH MỤC VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Tiếng anh** | **Tiếng việt** |
| RPI | Raspberry Pi |  |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| OEM | Original Equipment Manufacturer | Nhà sản xuất thiết bị gốc |
| PCM | Pulse-Code Modulation | Bộ mã hóa mã xung |
| HDMI | High Definition Multimedia Interface | Giao diện đa phương tiện |
| GPIO | General purpose input/output | Cổng vào ra vạn năng |
| MP | Megapixel |  |
| IDE | Integrated Development Environment | Môi trường tích hợp để viết code |
| DC | Direct Current | Điện một chiều |
| AC | Alternating Current | Điện xoay chiều |
| PLC | Programmable Logic Controller | Bộ điều khiển logic lập trình được |
| LCD | Liquid Crystal Display | Màn hình tinh thể lỏng |
| OpenCV | Open Computer Vision | Thư viện mã nguồn mở |

**Chương 1.     GIỚI THIỆU ĐỀ TÀI**

**1.1      Đặt vấn đề.**

Ngày nay cùng với sự phát triển của các ngành khoa học kỹ thuật, kỹ thuật điện- điện tử và điều khiển tự động đóng vai trò hết sức quan trọng trong mọi lĩnh vực khoa học, quản lý, công nghiệp tự động hóa, cung cấp thông tin… Do đó chúng ta phải nắm bắt và vận dụng nó một cách có hiệu quả nhằm góp phần vào sự nghiệp phát triển nền khoa học kỹ thuật thế giới nói chung và trong sự phát triển kỹ thuật điều khiển tự động nói riêng. Với những kỹ thuật tiên tiến như vi xử lý, PLC, vi mạch số… được ứng dụng vào lĩnh vực điều khiển, thì các hệ thống điều khiển thô sơ, với tốc độ xử lý chậm chạp ít chính xác được thay thế bằng các hệ thống điều khiển tự động với các lệnh chương trình đã được thiết lập trước.

Trong quá trình hoạt động ở các nhà xưởng, xí nghiệp hiện nay, việc tiết kiệm điện năng là nhu cầu rất cần thiết, bên cạnh đó ngành công nghiệp ngày càng phát triển các công ty xí nghiệp đã đưa tự động hóa và sản xuất để tiện ích cho việc quản lý dây chuyền và sản phẩm cho toàn bộ hệ thống một cách hợp lý là yêu cầu thiết yếu, tiết kiệm được nhiều thời gian cũng như quản lý một cách dễ dàng. Để đáp ứng yêu cầu đó, em đã tiến hành nghiên cứu tài liệu, thiết kế mô hình hệ thống phân loại sản phẩm theo màu sắc.

Những lợi ích mà hệ thống phân loại sản phẩm đem lại cho chúng ta là rất lớn, cụ thể như:

-       Giảm sức lao động, tránh được sự nhàm chán trong công việc, cải thiện được điều kiện làm việc của con người, tạo chongười lao động tiếp cận với sự tiến bộ của khoa học kỹ thuật và được làm việc trong môi trường ngày càng tốt hơn.

-       Nâng cao năng suất lao động, tạo tiền đề cho việc giảm giá thành sản phẩm, cũng như thay đổi mẫu mã một cách nhanh chóng và khoa học.

-       Giúp cho việc quản lý và giám sát trở nên rất đơn giản, bởi vì nó không những thay đổi điều kiện làm việc của công nhân mà còn có thể giảm số lượng công nhân đến mức tối đa…

**1.2      Các nghiên cứu liên quan trong và ngoài nước.**

**1.2.1Bài báo “*Detecting colors (Hsv Color Space) – Opencv with Python*”**

-     Tác giả thực hiện: Sergio Canu.

-     Thời gian đăng trực tuyến: 15/02/2019.

**Tóm tắt nội dung:**Phân loại màu sắc dựa trên OpenCV với ngôn ngữ lập trình Python.

**Kết quả đạt được:**Màu được tách một cách rõ ràng.

**Hình 1.1. Tách màu đỏ từ môi trường**

**1.2.2Bài báo “*Adruino Color Detection*”**

-       Tác giả thực hiện: Marcelo.

-       Thời gian đăng trực tuyến: 12/07/2017.

**Tóm tắt nội dung:**sử dụng Adruino phát hiện màu sắc với cảm biến màu sắc TSC 3200.

**Kết quả thực hiện:**màu được phát hiện và hiện kết quả tên màu sắc và hệ màu RGB lên màn hình LCD.

**Hình 1.2. Phát hiện màu vàng**

**Hình 1.3. Phát hiện màu đỏ**

**1.2.3Bài báo “*Hệ thống phân loại theo màu sắc xử dụng PLC S7 - 1200*”**

-       Tác giả thực hiện: Trần Khánh Dưng, Hoàng Như Ý.

-       Nơi thực hiện: Trường Đại Học Sư Phạm Kỹ Thuật, 03/07/2018, TP. Hồ Chí Minh.

**Tóm tắt nội dung:**sử dụng cảm biến TCS 3200 để phân loại màu sắc gửi tín hiệu cho PLC, xử dụng cảm biến hồng ngoại nhận biết vị trí cà chua và xử dụng xilanh khí nén để đẩy.

**Kết quả thực hiện:**mô hình được hoàn thiện, tồn tại một số hạn chế.

**Hình 1.4. Kết quả thi công phần cứng**

**1.2.4 Bài báo “*Delta Robot*”**

-       Tác giả thực hiện: Giáo sư Reymond Clave.

-       Nơi thực hiện: Tại EPFL - École Polytechnique Fédérale de Lausanne.

**Tóm tắt nội dung:** Ông đưa ra ý tưởng về việc sử dụng các khối hình bình hành để chế tạo một robot song song với ba phép tịnh tiến và một bậc tự do quay.

**Kết quả thực hiện:**Là một trong những thiết kế robot song song thành công nhất, với hàng trăm robot đang hoạt động trên toàn thế giới.

Hình 4. Thiết kế của Delta Robot.

**1.3      Mục tiêu**

Mục tiêu nghiên cứu của đề tài: “Thiết kế và thi công hệ thống phân loại sản phẩm theo màu sắc sử dụng công nghệ xử lý ảnh dựa trên nền tảng Raspberry” được chia làm các phần:

-       Tìm hiểu về các mô hình phân loại màu sắc có sẵn dựa theo đó chọn các thiết bị phù hợp cho mô hình.

-       Tìm hiểu về Raspberry và các thiết bị được chọn.

-       Tìm hiểu về cách kết nối giữa Raspberry và các thiết bị được chọn.

-       Tìm hiểu về ngôn ngữ lập trình Python.

-       Từ những kiến thức tìm hiểu được thiết kế mô hình tối ưu nhất trong khả năng.

**1.4      Phạm vi của đề tài.**

Về nội dung nghiên cứu: luận văn nghiên cứu xoay quanh tìm hiểu về nhận dạng, phân loại sản phẩm bằng phương pháp xử lý ảnh thông qua Kit Raspberry Pi 4 Model B.

Về không gian nghiên cứu: tập trung nghiên cứu xoay quanh xử lý ảnh, thực hiện phân loại bằng mô hình Delta Robot.

Về thời gian nghiên cứu: 4 tháng.

**Chương 2.     CƠ SỞ LÝ THUYẾT**

**2.1.     Các linh kiện được sử dụng trong đề tài.**

**2.1.1     Kit Raspberry Pi 4 model B.**

**2.1.1.1    Giới thiệu về Raspberry Pi và Raspberry pi 4 Model B.**

Raspberry Pi là từ để chỉ các máy tính chỉ có một board mạch (hay còn gọi là máy tính nhúng) kích thước chỉ bẳng một thẻ tín dụng, được phát triển tại Anh bởi Raspberry Pi Foundation với mục đích ban đầu là thúc đẩy việc giảng dạy về khoa học máy tính cơ bản trong các trường học và các nước đang phát triển.

Nhiệm vụ ban đầu của dự án Raspberry Pi là tạo ra máy tính rẻ tiền có khả năng lập trình cho những sinh viên , nhưng Pi đã được sự quan tầm từ nhiều đối tượng khác nhau. Đặc tính của Raspberry Pi xây dựng xoay quanh bộ xử lí SoC Broadcom BCM2835 (là chip xử lí mobile mạnh mẽ có kích thước nhỏ hay được dùng trong điện thoại di động) bao gồm CPU , GPU , bộ xử lí âm thanh /video , và các tính năng khác … tất cả được tích hợp bên trong chip có điện năng thấp này .

Raspberry Pi sản xuất bởi 3 OEM: Sony, Qsida, Egoman. Và được phân phối chính bởi Element14, RS Components và Egoman.

**Hình 2.1. Raspberry Pi 4 Model B.**

Raspberry Pi 4 Model B nổi bật với chip 4 nhân 64-bit có tốc độ 1.4GHz. Phiên bản mới còn hỗ trợ Wifi Dual-band 2.4GHz và 5GHz, Bluetooth 5.0/Bluetooth Low Energy, cổng Ethernet tốc độ cao (300Mbps) và Power over Ethernet (PoE) thông qua PoE HAT.

Ngoài những thay đổi đó, kích thước bề ngoài cũng như phần mềm của Raspberry Pi 4 Model B không thay đổi. Nghĩa là bạn hoàn toàn có thể sử dụng các phụ kiện cũ như vỏ, thẻ nhớ, adapter (nguồn điện) hoặc các phần mềm đang chạy trên Raspbian/Ubuntu hiện tại. Tuy nhiên, về hệ điều hành thì các bạn cần tải bản mới nhất để có thể hỗ trợ được bo mạch mới nhất này.

**2.1.1.2    Thông số kỹ thuật của Raspberry Pi 4 Model B.**

**Hình 2.2. Thông số phần cứng của Raspberry Pi 3 Model B**

**Thông số kỹ thuật:**

-       Vi xử lý: Broadcom BCM2711,quad-core Cortex-A72 (ARMv8) 64-bit SoC @1.5GHz.

-       RAM: 2GB, 4GB hoặc 8GB LPDDR4 (tùy nhu cầu sử dụng).

-       Kết nối: 2.4GHz và 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 5.0, BLE, Gigabit Ethernet với 5 cổng USB 2.0.

-       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.265 (4Kp60 decode).

-       Lưu trữ: MicroSD.

-       Nguồn điện sử dụng: 5V/2.5A DC cổng microUSB 5V, DC trên chân GPIO, Power over Ethernet (PoE)  (yêu cầu thêm PoE HAT).

**2.1.1.3       Những lưu ý khi sử dụng Raspberry.**

Nguồn sử dụng cho Board là nguồn 5VDC, dòng điện từ 2,5A trở lên, chất lượng tốt và ổn định, trường hợp nguồn không đủ dòng sẽ dẫn đến hiện tượng sụt áp cháy máy.

Các chân GPIO của Raspberry Pi (40 chân cắm Header) nếu giao tiếp điện áp trên (lớn hơn) 3.3V, chân cấp nguồn 3.3V và 5V nếu chạm với GND (Mass) sẽ làm chập nguồn cháy máy ngay lập tức, nếu không có nhu cầu, vui lòng không sử dụng và nên mua thêm vỏ bảo vệ để tránh chập, chạm.

Nếu bạn cần sử dụng các chân GPIO của Raspberry Pi để giao tiếp với các mạch có mức điện áp giao tiếp lớn hơn 3.3V hoặc sử dụng với mạch điều khiển động cơ thì cần dùng thêm mạch chuyển tín hiệu.

**Hình 2.3. Sơ đồ chân của Raspberry Pi 4 Model B.**

**2.1.2     Module điều khiển động cơ A4988.**

A4988 là bo mạch điều của Allegro A4988 DMOS với tính năng bảo vệ động cơ khi giới hạn điện áp và quá dòng. Module cho phép điều khiểnđộng cơ bước lưỡng cực với dòng ra tối đa là 2A. Module điều khiển bước và hướng đơn giản cùng năm bước khác nhau là 1 bước, 1/2 bước, 1/4 bước, 1/8 bước và 1/16 bước. Điều chỉnh dòng điện theo dòng định mức của động cơ, dòng điện cực đại khi điều chỉnh chiết áp, cho phép bạn sử dụng điện áp cao hơn điện áp định mức của động cơ bước để đạt được tốc độ bước cao hơn. Điều khiển băm xung thông minh tự động chọn chế độ di chuyển chính xác (di chuyển nhanh hoặc di chuyển chậm). Ngắt điện khi quá nhiệt, khóa điện áp ngược và bảo vệ dòng điện. Bảo vệ ngắn mạch và quá tải. A4988 cũng dễ dàng kết hợp với arduino thay vì sử dụng các module điều khiển Servo đắt tiền khác.

**Hình 2.4. Module điều khiển động cơ bước A4988.**

**Thông số kỹ thuật:**

|  |  |
| --- | --- |
| **Điện áp định mức:** | 8 V. |
| **Điện áp cực đại:** | 35 V. |
| **Dòng điện mỗi pha:** | 1 A. |
| **Dòng cực đại mỗi pha:** | 2 A. |
| **Điện áp hoạt động tối thiểu:** | 3 V. |
| **Điện áp hoạt động định mức:** | 5.5 V. |
| **Số bước điều khiển:** | 1, 1/2, 1/4, 1/8, 1/16. |
| **Kích thước:** | 0.6 x 0.8cm. |
| **Trọng lượng:** | 1.3 g. |

**2.1.3     Băng tải**

Băng tải hay còn gọi băng chuyền là một thiết bị dùng vận chuyển, chuyền tải vật liệu, hàng hóa từ vị trí này đến vị trí kia. So với các phương thức vận tải khác như vận tải đường bộ và đường sắt, hệ thống băng tải có nhiều lợi thế như công suất cao hơn, chi phí thấp hơn, hiệu quả cao hơn, ít sự tham gia của con người và độ tin cậy đã được chứng minh.

**Hình 2.6. Băng tải mini sử dụng cho các mô hình.**

Cấu tạo băng tải được sử dụng trong mô hình:

-       Khung băng tải được thiết kế từ nhôm định hình.

-       Dây băng tải là bằng dây PVC,có độ dày 2mm, có màu xanh lá.

-       Hệ thống con lăn đỡ dây băng tải.

-       Động cơ bướcNEMA 17.

**Hình 2.7. Động cơ bước NEMA 17.**

**Thông số kỹ thuật của động cơ:**

-       Điện áp: 12VDC.

-       Dòng định mức: 2.5A.

-       Tốc độ không tải: 52RPM (52 vòng 1 phút).

-       Đường kính trục: 5mm.

-       Chiều dài trục: 14mm.

**2.1.5   Module Arduino Nano.**

**Hình 2.12. Module Arduino Nano.**

Arduino Nano là một bo mạch nhỏ, hoàn chỉnh với bo mạch, dựa trên ATmega328 (Arduino Nano 3.x). Module chỉ thiếu giắc cắm nguồn DC và chỉ hoạt động khi được cấp nguồn với cáp USB Mini-B thay vì cáp tiêu chuẩn. Arduino Nano cũng dễ dàng kết hợp với Raspberry theo phương pháp Master - Slaves.

BẢNG 7. Thông số kỹ thuật của Arduino Nano:

|  |  |
| --- | --- |
| **Bộ vi điều khiển:** |  |
| **Thiết kế:** |  |
| **Điện áp hoạt động:** |  |
| **Bộ nhớ hoạt động:** | 32 KB (2 KB dùng cho bộ nạp). |
| **SRAM( bộ nhớ lưu trữ):** |  |
| **Tốc độ truy cập**: | 16 MHz. |
| **Cổng kết nối:** |  |
| **Bộ nhớ trong:** |  |
| **Dòng điện mỗi chân:** |  |
| **Dòng hoạt động:** | 7 – 12V. |
| **Chân kết nối:** | 22 (6 chân xung PWM). |
| **Chân xung PWM:** |  |
| **Dòng tiêu thụ:** |  |
| **Kích thước:** | 18 x 45mm. |
| **Nặng:** |  |
| **Mã sản phẩm:** |  |

**2.1.6   Động cơ bước Nema 17.**

**Hình 2.14. Động cơ bước Nema 17.**

Nema 17 là động cơ bước lưỡng cực với góc bước 1.8° (200 bước/vòng). Nema 17 thường được sử dụng trong máy in, máy CNC và Máy cắt Laser. Mặc dù Nema 17 không có chức năng phản hồi tín hiệu và độ chính xác thấp hơn những động cơ Servo hoặc động cơ bước có bộ xử lý nhạy khác nhưng giá thành rẻ hơn các loại khác và dễ tìm mua trên thị trường.

BẢNG 9. Thông số kỹ thuật của Nema-17 42MM:

|  |  |
| --- | --- |
| **Thuộc tính:** | **Nema-17 Bipolar 42mm** |
| Loại động cơ: | Lưỡng cực. |
| Góc bước: | 1.8°. |
| Bước chính xác: | 5%. |
| Mô-men xoắn: | 4.1 kg/cm. |
| **Thuộc tính điện:** | |
| Điện áp định mức: | 12 VDC. |
| Dòng điện định mức: |  |
| Trở kháng: | 1.1 Ω. |
| Điện cảm: | 2.6mH ± 20% (1KHz). |
| **Thuộc tính thiết kế:** | |
| Đường kính trục: | 5 mm. |
| Khối lượng: | 300 g. |
| Số chân: |  |

**2.1.7Động cơ bơm chân không JH12-65 12VDC.**

Động cơ được cấp nguồn để khởi động động cơ DC nhỏ, chuyển động cơ học bên trong là chuyển động lệch tâm, được dẫn động bằng chuyển động lệch tâm bên trong pit tông màng. Buồng bơm cố định thể tích nén khí (cổng hút nén đóng, cổng xả để xả áp suất không khí). Khi ở trạng thái hút khí (cổng xả khí nén đóng, cổng hút áp suất mở ra), cổng hút khí có sự chênh lệch áp suất với môi trường bên ngoài và xả ra khi có tín hiệu xa.

Máy bơm mini được sử dụng phổ biến trong máy lấy mẫu không khí, dụng cụ và thiết bị, ngành y tế và hóa chất, thiết bị gia dụng và các lĩnh vực khác.

BẢNG 11. Thông số kỹ thuật của động cơ bơm chân không JH12-65 12VDC:

|  |  |
| --- | --- |
| **Vật liệu** | Nhựa & Kim loại |
| **Kích thước:** | Xấp xỉ. 6 x 3.8 x 4.5 cm |
| **Điện áp hoạt động:** | DC 9 V ~ 14V |
| **Điện áp định mức:** | 12 VDC. |
| **Dòng điện định mức:** |  |
| **Công suất định mức:** |  |
| **Công suất hút:** | 12L/phút. |
| **Áp suất chân không:** | ~ 80 kpa |
| **Áp suất cực đại:** | 120 kpa. |
| **Tiếng ồn:** | < 65db. |

**2.1.8Nguồn xung**

Nguồn tổ ong là cách gọi khác của nguồn xung. Cái tên nguồn tổ ong bắt nguồn từ hình dạng các lỗ thông hơi thoát nhiệt của bộ nguồn xung được đục lỗ lục giác giống với cấu tạo của tổ ong nên dân gian gọi vậy cho thân thuộc dễ nhớ.

Nguồn xung là bộ nguồn có tác dụng biến đổi từ nguồn điện xoay chiều sang nguồn điện một chiều bằng chế độ dao động xung tạo bằng mạch điện tử kết hợp với một biến áp xung.

**Hình 2.19. Sơ đồ khối nguồn xung**

**Cấu tạo của nguồn xung:**

**Biến áp xung:** cũng cấu tạo gồm các cuộn dây quán trên một lõi từ giống như biến áp thông thường chỉ có điều biến áp này sử dụng lõi ferit còn biến áp thường sử dụng lỗi thép kỹ thuật điện . Với cùng một kích thước thì biến áp xung cho công suất lớn hơn biến áp thường rất nhiều lần. Ngoài ra biến áp xung hoạt động tốt ở dải tần cao còn biến áp thường chỉ hoạt động ở dải tần thấp.

**Cầu chì:** bảo vệ mạch nguồn bị ngắn mạch.

**Cuộn chống nhiễu, tụ lọc sơ cấp, điode chỉnh lưu:**có nhiệm vụ biến đổi điện áp xoay chiều 220V thành điện áp một chiều tích trữ trên tụ lọc sơ cấp để cung cấp năng lượng cho cuộn sơ cấp của máy biến áp xung.

**Sò công suất:** đây là một linh kiện bán dẫn dùng như một công tắc chuyển mạch , đó có thể là transistor, mosfet, IC tích hợp, IGBT có nhiệm vụ đóng cắt điện từ chân (+) của tụ lọc sơ cấp vào cuộn dây sơ cấp của biến áp xung rồi cho xuống mass.

**Tụ lọc nguồn thứ cấp:** dùng để tích trữ năng lượng điện từ cuộn thứ cấp của biến áp xung để cấp cho tải tiêu thụ. Chúng ta biết rằng khi cuộn sơ cấp của biến áp được đóng cắt điện liên tục bằng sò công suất thì xuất hiện từ trường biến thiên dẫn đến cuộn thứ cấp của biến áp cũng xuất hiện một điện áp ra. Điện áp này được chỉnh lưu qua một vài diode rồi đưa ra tụ lọc (tụ điện) thứ cấp để san phẳng điện áp.

**IC quang và IC TL431:** có nhiệm vụ tạo ra một điện áp cố định để khống chế điện áp ra bên thứ cấp ổn định theo mong muốn. Chúng sẽ làm nhiệm vụ khống chế dao dộng đóng cắt điện vào cuộn sơ cấp của biến áp xung sao cho điện áp ra bên thứ cấp đạt yêu cầu.

**Hình 2.20. Cấu tạo của nguồn xung**

**2.2.     Ngôn ngữ Python và thư viện OpenCV**

**2.2.1Ngôn ngữ Python**

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

**2.2.2   Thư viện OpenCV**

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.

Các tính năng của thư viện OpenCV:

-       Đối với hình ảnh chúng ta có thể đọc và lưu hay ghi chúng.

-       Có thể ghi và đọc video.

-       Xử lý ảnh có thể lọc nhiễu hoặc chuyển đổi ảnh.

-       Thực hiện nhận dạng đặc điểm của hình dạng ảnh

-       Phát hiện các đối tượng được xác định trước như khuôn mặt, mắt, … trong video hoặc hình ảnh.

-       Phân tích video, ước lượng chuyển động của đối tượng xuất hiện trong video, tách nền và đối tượng riêng biệt, …

**2.3.     Phương pháp phân loại màu sắc**

**2.3.1.  Hệ màu RGB (Red – Green – Blue)**

Mô hình màu RGB sử dụng mô hình bổ sung trong đó ánh sáng đỏ, xanh lá cây và xanh lam được tổ hợp với nhau theo nhiều phương thức khác nhau để tạo thành các màu khác.

Từ viết tắt RGB trong tiếng Anh có nghĩa:

-       R: Red (màu đỏ).

-       G: Green (màu xanh lá cây).

-       B: (blue (màu xanh lam)

Khi biểu diễn dưới dạng số, các giá trị RGB trong mô hình 24 bpp thông thường được ghi bằng cặp ba số nguyên giữa 0 và 255, mỗi số đại diện cho cường độ của màu đỏ, xanh lá cây, xanh lam trong trật tự như thế.

Ví dụ:

-       (0, 0, 0) là màu đen.

-       (255, 255, 255) là màu trắng.

-       (255, 0, 0) là màu đỏ.

-       (0, 255, 0) là màu xanh lá cây.

-       (0, 0, 255) là màu xanh lam.

-       (255, 255, 0) là màu vàng.

-       (0, 255, 255) là màu xanh ngọc.

-       (255, 0, 255) là màu hồng sẫm.

**Hình 2.21. Mô hình không gian màu RGB**

**2.3.2.  Hệ màu HSV**

Thực chất của không gian này là sự biến đổi của không gian RGB. Không gian HSV được mô tả bằng lệnh lập phương RGB quay trên đỉnh Black. H (Hue) là góc quay trục V (value) qua 2 đỉnh Black và White.

**Trong đó:**

-       H (Hue) chỉ sắc thái có giá trị từ 0o – 360o.

-       S (Saturation) chỉ độ bão hoà. Là mức độ thuần khiến của màu, có thể hiểu là có bao nhiêu màu trắng được thêm vào màu thuần khiết này. Giá trị S nằm trong đoạn [0, 255], trong đó S = 255 là màu tinh khiết nhất, hoàn toàn không pha trắng. Nói cách khác, S càng lớn thì màu càng tinh khiết, đậm.

-       V (Value) Độ sáng của màu. Còn được gọi là Intensity, Lightness, có giá trị dao động trong đoạn [0, 255], trong đó V= 0 là hoàn toàn tối (đen), V = 255 là hoàn toàn sáng. Nói cách khác, V càng lớn thì màu càng sáng.

**Hình 2.22. Mô hình màu HSV**

**Hình 2.23. Bản đồ màu HSV**

Ví dụ: Trích xuất màu xanh lá và màu vàng trong hình dưới đây:

**Hình 2.24. Hoa hướng dương khi chưa tách màu**

Ta sử dụng bản đồ màu HSV trên để xác định vùng màu xanh lá của bông hoa hướng dương trên (36,0,0) ~ (75,255,255) sẻ được kết quả như hình sau:

**Hình 2.25. Tách màu xanh ra khỏi bức hình**

Bản đồ màu trên giúp ta tìm được vùng màu HSV một cách nhanh chóng:

**Hình 2.26. Một số ví dụ khác khi sử dụng bản đồ màu**

**Chương 3.     NỘI DUNG THỰC HIỆN**

**3.1.     Mô hình hệ thống**

**3.1.1  Giới thiệu về hệ thống**

Hệ thống phân loại sẽ thực hiện nhận biết màu sắc của cà chua ở đầu vào sau đó truyền tín hiệu màu đã qua xử lý đến khối xử lý để thực hiện tác vụ phân loại. Trong suốt quá trình phân loại thì sự tác động của các xi lanh, số lượng quả đã được tác động sẽ được thể hiện trên giao diện điều khiển. Dưới đây là mô hình chung của hệ thống:

**Hình 3.1. Mô hình chung của hệ thống**

**3.1.2Chức năng từng bộ phận**

Xử lý màu sắc: xử lý tín hiệu nhận được từ camera và tạo ra tín hiệu đưa về Raspberry để thực hiện các tác vụ khác.

Phân loại cà chua: nhận các tín hiệu từ khối xử lý màu sắc để thực hiện các tác vụ phân loại cà chua dựa trên chương trình được viết theo các yêu cầu được đưa ra.

Giao diện điều khiển: Thực hiện việc điều khiển tắt mở hệ thống và xử lý và hiển thị dữ liệu.

**3.2.     Tính toán và thiết kế hệ thống**

**3.2.1  Thiết kế sơ đồ khối hệ thống**

Với yêu cầu của đề tài, em tiến hành thiết kế sơ đồ với các và chức năng của các khối như sau:

**Hình 3.2. Sơ đồ khối của hệ thống**

**Chức năng các khối:**

-       Khối xử lý màu sắc: có chức năng xử lý tín hiệu từ cảm biến màu sắc và gửi tín hiệu đến khối xử lý trung tâm.

-       Khối cảm biến: cảm biến nhận biết cà chua khi chạy qua băng tải.

-       Khối băng chuyền: có chức năng tải cà chua đến các khu vực khác trong hệ thống.

-       Khối điều khiển khí nén: có chức năng đẩy cà chua được phân loại ra khỏi băng chuyền.

-       Khối xử lý trung tâm: có chức năng nhận, xử lý thông tin và điều khiển các khối khác.

-       Khối nguồn: có chức năng cấp nguồn cho các thiết bị trong hệ thống.

**3.2.2Thiết kế các khối**

**3.2.2.1  Khối cảm biến**

Sử dụng cảm biến vật cản hồng ngoại E18 – D80NK để nhận biết có cà chua đến vị trí xilanh khí nén, cho dừng băng tải, xilanh ở vị trí màu đó hoạt động.

Do mức tín hiệu của cảm biến là 5VDC mà mức điện áp tối đa của chân GPIO của Raspberry chỉ có 3.3VDC nên em sử dụng bộ chuyển đổi 2 mức tín hiệu (Level Converter) để chuyển mức tín hiệu của cảm biến xuống 3.3VDC. Nếu không chuyển mức tính hiệu của cảm biến xuống 3.3VDC sẽ dẫn đến chập, chạm cháy board mạch Raspberry ngay.

**Sơ đồ kết nối cảm biến với Raspberry:**

**Hình 3.3. Sơ đồ kết nối 3 cảm biến với Raspberry**

Dây nguồn (màu nâu) của cảm biến được nối với nguồn 5VDC, dây GND (màu xanh) được nối với 0VDC. Dây tín hiệu (màu đen) được nối vào bộ chuyển đổi (Level Converter) theo thứ tự là: CB1 nối HV1, CB2 nối HV2, CB3 nối HV3.

Chân HV của bộ chuyển đổi nối nguồn 5VDC chân LV nối nguồn 3.3VDC của Raspberry, 2 chân GND được nối chung giữa nguồn 5VDC và chân GND của Raspberry.

Các tín hiệu của cảm biến đã được chuyển đổi từ 5VDC xuống 3.3VDC ở các chân đối diện: HV1 – LV1, HV2 – LV2, HV3 – LV3. Nối dây từ các chân LV vào Raspberry để thực hiện viết chương trình.

**3.2.2.2  Khối băng chuyền**

Khối băng chuyền sẽ thực hiện việc chuyển cà chua đến vị trí thực hiện phân loại (camera Rasperry) và vị trí của cảm biến vật cản. Do là mô hình nhỏ nên sử dụng động cơ giảm tốc 37GB520R với điện áp 12VDC.

Băng tải được nối với 1 relay trong module relay 4 kênh để phục vụ cho việc dừng và chạy tiếp: Dây nối từ nguồn 12VDC chân COM của relay, chân NO được nối với chân (+) của băng tải, chân (-) của băng tải được nối với 0VDC.

**Hình 3.4. Sơ đồ kết nối của băng tải**

**3.2.2.3  Khối hệ thống điều khiển khí nén**

Hệ thống điều khiển khí nén sẽ thực hiện nhiệm vụ đưa cà chua ra khỏi băng tải để hoàn thành việc phân loại. Hệ thống sẽ gồm 2 phần chính là xi lanh và van điện từ.

Để điều khiển được hành trình của xilanh thì ta sẽ điều khiển các cuộn dây của van điện từ thông qua relay và Raspberry. Dưới đây là sơ đồ kết nối của van với relay và Raspberry:

**Hình 3.5 Sơ đồ kết nối của 3 van khí nén với module relay 4 kênh và Raspberry**

Chân (+) của van được nối vào chân NO của relay, chân COM của relay nối vào V+ của nguồn 24VDC.

Chân (-) của van được nối thẳng vào V- của nguồn 24VDC.

Module relay được cấp nguồn 5VDC nối đất chung với Raspberry.

Chân kích của các relay (IN1, IN2, …) được nối vào Raspberry để viết chương trình.

**3.2.2.4 Khối xử lý trung tâm**

Ở đây khối xử lý trung tâm được sử dụng là Rasperry Pi 3 Model B+.

Khối xử lý trung tâm thực hiện việc lọc, phân loại các khoảng màu chín, gần chín, còn xanh của cà chua thông qua hình ảnh lấy được từ camera bằng chương trình Python đã được viết.

Raspberry nhận tín hiệu từ các cảm biến hồng ngoại để xác định vị trí của cà chua và thực hiện việc điều khiển các van điện từ để đẩy cà chua được phân loại về đúng vị trí được chọn.

**Hình 3.6. Camera kết nối với Raspberry Pi 3 Model B+**

**3.2.2.5  Khối nguồn**

Đối với khối nguồn do ở đây chúng ta vừa cần dùng nguồn AC và DC nên nguồn AC 220V chúng ta sẽ lấy trực tiếp từ lưới điện để cấp cho nguồn xung.

Nguồn DC để cấp cho relay, van điện từ, được lấy từ nguồn tổ ong 24VDC và 5VDC, riêng băng tải sử dụng mạch giảm áp LM2596 để giảm từ 24VDC xuống 12VDC.

**Hình 3.7. Mạch giảm áp LM2596**

**3.2.2.6  Khối xử lý màu sắc**

Sử dụng Camera Raspberry pi V1 5MP để thu nhận hình ảnh từ buồng sáng.

Khi ta chiếu ánh sáng trắng vào một vật thể bất kì, tại bề mặt vật thể sẽ xảy ra hiện tượng hấp thụ và phản xạ ánh sáng. Dựa trên nguyên lý sự phản xạ, hấp thụ ánh sáng trắng của vật thể và sự phối trộn màu sắc bởi 3 màu cơ bản Blue, Green, Red. Ta chuyển đổi hình ảnh nhận được từ hệ màu RGB sang HSV dựa vào thư viện OpenCV trong ngôn ngữ Python bằng câu lệnh:

hsv\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

**Cách chuyển đổi RGB sang HSV**

Các giá trị màu của hệ màu RGB được chia cho 255 để chuyển đổi giá trị từ 0…255 cho 0…1:

Trong đó:

R’, G’, B’ là giá trị được chuyển đổi từ 0…255 sang 0…1.

Cmax, Cmin là giá trị lớn nhất, nhỏ nhất được chọn từ R’, G’, B'.

Giá trị H trong hệ màu HSV được tính theo công thức sau:

                        (2.7)

Giá trị S được tính dựa trên công thức:

                                             (2.8)

Giá trị V được chọn theo Cmax:

                                                                 (2.9)

Các công thức được sử dụng tương ứng với giá trị Cmax tìm được ở (2.4).

**Ví dụ:**

Chuyển màu đỏ từ

......

**TÓM TẮT ĐỀ TÀI**

Cùng với sự phát triển của thế giới và xu hướng hội nhập kinh tế quốc tế, đất nước ta đang đổi mới và bước vào thời kì công nghiệp hóa, hiện đại hóa, vừa xây dựng cơ sở vật chất kỹ thuật hiện đại, vừa phát triển nền kinh tế đất nước. Từ đó, chúng ta đòi hỏi phải nghiên cứu và áp dụng những dây chuyền, máy móc và thiết bị tiên tiến hiện đại, cókhả năng tự động hóa cao để đưa công nghệ vào mọi lĩnh vực của cuộc sống. Trong đó ngành công nghiệp tự động hóa đóng một vai trò rất quan trọng trong quá trình phát triển của đất nước. Để đáp ứng nhu cầu to lớn của việc phát triển ngành Tự động hóa nói chung, đòi hỏi phải có đội ngũ cán bộ, nhân viên kỹ thuật có khả năng, đủ năng lực và trình độ chuyên môn để kịp thời giải quyết mọi vấn đề liên quan đến kỹ thuật cơ khí, điện-điện tử và kỹ thuật phần mềm.

Từ những thực tế nói trên, bản thân là sinh viên của ngành **Điện Tử,**từ những kiến thức được học, em đã lựa chọn đề tài: **“Thiết kế và thi công hệ thống phân loại sản phẩm theo màu sắc sử dụng công nghệ xử lý ảnh dựa trên nền tảng Raspberry”.**Việc tạo ra một hệ thống như vậy để thay thế con người trong công việc là vấn đề hết sức cần thiết.

Luận văn này được chia làm 8 chương chính:

Chương 1: Giới thiệu đề tài.

Chương 2: Cơ sở lý thuyết

Chương 3: Nội dung thực hiện

Chương 4: Kết quả thực hiện

Chương 5: Kết luận

**MỤC LỤC**

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.. i

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN.. ii

LỜI CẢM ƠN   iii

TÓM TẮT ĐỀ TÀIiv

MỤC LỤC.. v

MỤC LỤC HÌNH ẢNH.. viii

DANH MỤC BẢNG BIỂU.. xi

DANH MỤC VIẾT TẮT. xii

Chương 1.     GIỚI THIỆU ĐỀ TÀI1

1.1        Đặt vấn đề. 1

1.2        Các nghiên cứu liên quan trong và ngoài nước. 1

1.2.1           Bài báo “*Detecting colors (Hsv Color Space) – Opencv with Python*”. 1

1.2.2           Bài báo “*Adruino Color Detection*”. 2

1.2.3           Bài báo “*Hệ thống phân loại theo màu sắc xử dụng PLC S7 - 1200*”. 3

1.3        Mục tiêu. 3

1.4        Phạm vi của đề tài4

Chương 2.     CƠ SỞ LÝ THUYẾT. 5

2.1.       Các linh kiện được xử dụng trong đề tài5

2.1.1           Kit Raspberry Pi 3 model B. 5

2.1.1.1        Giới thiệu về Raspberry Pi và Raspberry pi 3 Model B  5

2.1.1.2        Thông số kỹ thuật của Raspberry Pi 3 Model B. 6

2.1.1.3        Những lưu ý khi xử dụng Raspberry. 7

2.1.2           Module Camera Raspberry V1 5MP. 8

2.1.3           Băng tải9

2.1.4           Xilanh khí nén và Van điện từ khí nén. 10

2.1.4.1        Xilanh khí nén. 10

2.1.4.2        Van điện từ khí nén. 11

2.1.5           Module relay 5V 4 kênh. 14

2.1.6     Bộ chuyển đổi mức logic hai hướng 4 kênh. 15

2.1.7     Cảm biến hồng ngoại E18 – D80NK.. 16

2.1.8     Nguồn xung. 18

2.2.       Ngôn ngữ Python và thư viện OpenCV.. 20

2.2.1     Ngôn ngữ Python. 20

2.2.2     Thư viện OpenCV.. 21

2.3.       Phương pháp phân loại màu sắc. 21

2.3.1.    Hệ màu RGB (Red – Green – Blue)21

2.3.2.    Hệ màu HSV.. 22

Chương 3..... NỘI DUNG THỰC HIỆN.. 25

3.1.       Mô hình hệ thống. 25

3.1.1     Giới thiệu về hệ thống. 25

3.1.2     Chức năng từng bộ phận. 25

3.2.       Tính toán và thiết kế hệ thống. 25

3.2.1     Thiết kế sơ đồ khối hệ thống. 25

3.2.2     Thiết kế các khối26

3.2.2.1  Khối cảm biến. 26

3.2.2.2  Khối băng chuyền. 27

3.2.2.3  Khối hệ thống điều khiển khí nén. 28

3.2.2.4 Khối xử lý trung tâm.. 29

3.2.2.5  Khối nguồn. 29

3.2.2.6  Khối xử lý màu sắc. 30

3.3.       Thực hiện phần mềm.. 32

3.3.1     Cài đặt hệ điều hành cho Raspberry. 32

3.3.2     Cài đặt thư viện OpenCV cho hệ điều hành Raspian Buster35

3.3.3     Phần mềm lập trình ngôn ngữ Python. 39

3.4.       Nguyên lý hoạt động của hệ thống. 40

3.5.       Lưu đồ giải thuật40

3.5.1     Lưu đồ giải thuật cho chương trình chính. 41

3.5.2     Lưu đồ giải thuật của khối hình ảnh. 42

Chương 4..... KẾT QUẢ THỰC HIỆN.. 43

4.1.       Kết quả. 43

4.1.1     Kết quả nghiên cứu. 43

4.1.2     Kết quả thi công. 43

4.1.2.1  Kết quả phần cứng. 43

4.1.2.2  Các bước thực nghiệm mô hình. 44

4.1.2.3 Kết quả thực nghiệm.. 45

4.2.       Biện luận. 46

Chương 5..... KẾT LUẬN.. 48

5.1.       Kết luận. 48

5.2.       Những thuận lợi và khó khăn khi thực hiện. 48

5.2.1     Thuận lợi48

5.2.2     Khó khăn. 48

5.3.       Các hạn chế của mô hình. 48

5.4.       Hướng phát triển hoàn thiện đề tài49

TÀI LIỆU THAM KHẢO.. 50

PHỤ LỤC.. 51

MỤC LỤC HÌNH ẢNH

Hình 1.1. Tách màu đỏ từ môi trường. 2

Hình 1.2. Phát hiện màu vàng. 2

Hình 1.3. Phát hiện màu đỏ. 3

Hình 1.4. Kết quả thi công phần cứng. 3

Hình 2.1. Raspberry Pi 3 Model B. 5

Hình 2.2. Thông số phần cứng của Raspberry Pi 3 Model B. 6

Hình 2.3. Sơ đồ chân của Raspberry Pi 3 Model B. 7

Hình 2.4. Camera Raspberry V1 5MP. 8

Hình 2.5 Camera được gắn vào kit Raspberry. 8

Hình 2.6. Băng tải mini sử dụng cho các mô hình. 9

Hình 2.7. Động cơ 37GB520R. 10

Hình 2.8. Xilanh SMC SDJ2D16 – 100B. 11

Hình 2.9. Cổng P của van 5/2. 12

Hình 2.10. Hai cổng A và B của van 5/2. 13

Hình 2.11. Hai cổng xả R vs S của van 5/2. 13

Hình 2.12. Module Relay 5VDC 4 kênh. 14

Hình 2.13. Sơ đồ nguyên lý của Module Relay. 14

Hình 2.14. Bộ chuyển đổi mức tín hiệu. 15

Hình 2.15. Sơ đồ nguyên lý của Level Coverter MH.. 15

Hình 2.16. Cảm biến hồng ngoại E18 – D80NK.. 16

Hình 2.17. Sơ đồ nguyên lý của cảm biến hồng ngoại17

Hình 2.18. Mô tả hoạt động của cảm biến hồng ngoại18

Hình 2.19. Sơ đồ khối nguồn xung. 19

Hình 2.20. Cấu tạo của nguồn tổ ong. 20

Hình 2.21. Mô hình không gian màu RGB. 22

Hình 2.22. Mô hình màu HSV.. 23

Hình 2.23. Bản đồ màu HSV.. 23

Hình 2.24. Hoa hướng dương khi chưa tách màu. 24

Hình 2.25. Tách màu xanh ra khỏi bức hình. 24

Hình 2.26. Một số ví dụ khác khi sử dụng bản đồ màu. 24

Hình 3.1. Mô hình chung của hệ thống. 25

Hình 3.2. Sơ đồ khối của hệ thống. 26

Hình 3.3. Sơ đồ kết nối 3 cảm biến với Raspberry. 27

Hình 3.4. Sơ đồ kết nối của băng tải28

Hình 3.5 Sơ đồ kết nối của 3 van khí nén với module relay 4 kênh và Raspberry. 28

Hình 3.6. Camera kết nối với Raspberry Pi 3 Model B+. 29

Hình 3.7. Mạch giảm áp LM2596. 30

Hình 3.8. Xác định hệ màu RGB thông qua ứng dụng Paint31

Hình 3.9. Format thẻ nhớ. 33

Hình 3.10. Ghi file hệ điều hành vào thẻ nhớ SD.. 33

Hình 3.11. Mở putty và nhập ip để đăng nhập. 34

Hình 3.12. Chọn Advanced Options để có thể mở rộng bộ nhớ. 35

Hình 3.13. Sau khi biên dịch xong. 38

Hình 3.14. Khởi động Python IDE. 39

Hình 3.15. Chương trình đã viết40

Hình 3.16. Lưu đồ giải thuật chương trình chính của hệ thống. 41

Hình 3.17. Lưu đồ giải thuật của khối xử lý hình ảnh. 42

Hình 4.1. Mô hình phân loại cà chua dùng công nghệ xử lý ảnh. 43

Hình 4.2. Kết quả phân loại cà chua chín. 44

Hình 4.3. Kết quả phân loại cà chua gần chín. 44

Hình 4.4. Kết quả phân loại cà chua xanh. 45

Hình 4.5. Buồng camera được gắn led trắng. 46