

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**



## ĐỒ ÁN TỐT NGHIỆP

## ỨNG DỤNG XỬ LÝ ẢNH VÀ YOLO TRONG HỆ THỐNG

## MÁY THANH TOÁN TỰ ĐỘNG

**SVTH : TRƯỜNG QUỐC TÍN MSSV: 20161045**

DOÃN ĐỨC MINH MSSV: 20161231

KHÓA : 2020

## **NGÀNH : CNKT Điện tử -Viễn thông**

**GVHD** : Ths. Nguyễn Ngô Lâm

Tp. Thủ Đức, tháng 06 năm 2024

## BỘ GIÁO DỤC VÀ ĐÀO TẠO

# TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

## KHOA ĐIỆN - ĐIỆN TỬ



## **ĐỒ ÁN TỐT NGHIỆP**

## ỨNG DỤNG XỬ LÝ ẢNH VÀ YOLO TRONG HỆ THỐNG

## MÁY THANH TOÁN TỰ ĐỘNG

**SVTH : TRƯỜNG QUỐC TÍN MSSV: 20161045**

DOÃN ĐÚC MINH MSSV: 20161231

KHÓA : 2020

## **NGÀNH : CNKT Điện tử -Viễn thông**

**GVHD** : Ths. Nguyễn Ngô Lâm

Tp. Thủ Đức, tháng 06 năm 2024

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM



Độc lập - Tự do - Hạnh phúc

---\*\*\*---

SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

HCMC University of Technology and Education

Tp. Thủ Đức, ngày 21 tháng 06 năm 2024

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**Họ và tên sinh viên:** Trương Quốc Tín **MSSV:** 20161045

Doãn Đức Minh **MSSV:** 20161231

**Ngành:** Công Nghệ Kỹ Thuật Điện Tử - Viễn Thông **Lớp:** 20161CLVT1A,B

**Giảng viên hướng dẫn:** Ths. Nguyễn Ngô Lâm

**Ngày nhận đề tài:** 28/02/2024 **Ngày nộp đề tài:** 21/06/2024

**1. Tên đề tài:** Ứng dụng xử lý ảnh và Yolo trong hệ thống máy tính toán tự động

**2. Các số liệu, tài liệu ban đầu:**

- Kiến thức về các môn: Vật lý, Xử lý ảnh, Mạch điện, Hệ thống nhúng, IoTs
- Các tài liệu liên quan về lập trình Esp32, Raspberry, Python và Django Framework.
- Các kiến thức về ngôn ngữ lập trình C, Linux.
- Kết hợp với sáng tạo, vận dụng kiến thức từ các đồ án môn học trước

**3. Nội dung thực hiện đề tài:**

- Tính toán, lựa chọn linh kiện và thiết kế mô hình hệ thống
- Vẽ sơ đồ khối, sơ đồ nguyên lý thông qua phần mềm Proteus
- Lập trình cho hệ thống thông qua các phần mềm
- Triển khai mô hình hoàn chỉnh
- Viết báo cáo
- Bảo vệ đồ án tốt nghiệp

**4. Sản phẩm:** Mô hình hệ thống ứng dụng xử lý ảnh và Yolo trong hệ thống máy tính toán tự động.

TRƯỞNG NGÀNH

GIẢNG VIÊN HƯỚNG DẪN

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**Họ và tên sinh viên:** Trương Quốc Tín

**MSSV:** 20161045

Doãn Đức Minh

**MSSV:** 20161231

**Ngành:** Công Nghệ Kỹ Thuật Điện Tử - Viễn Thông    **Lớp:** 20161CLVT1A,B

**5. Tên đề tài:** Ứng dụng xử lý ảnh và Yolo trong hệ thống máy tính toán tự động

**Họ và tên Giáo viên hướng dẫn:** Ths. Nguyễn Ngô Lâm

**NHẬN XÉT:**

1. Về nội dung đề tài & khối lượng thực hiện:.....

.....

2. Ưu điểm: .....

.....

3. Khuyết điểm:.....

.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:.....

6. Điểm:.....(Bằng chữ: .....)

Tp. Thủ Đức, ngày 21 tháng 06 năm 2024

Giáo viên hướng dẫn

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

**Họ và tên sinh viên:** Trương Quốc Tín

**MSSV:** 20161045

Doãn Đức Minh

**MSSV:** 20161231

**Ngành:** Công Nghệ Kỹ Thuật Điện Tử - Viễn Thông    **Lớp:** 20161CLVT1A,B

**6. Tên đề tài:** Ứng dụng xử lý ảnh và Yolo trong hệ thống máy tính toán tự động

**Họ và tên Giáo viên hướng dẫn:** Ths. Nguyễn Ngô Lâm

**NHẬN XÉT:**

1. Về nội dung đề tài & khối lượng thực hiện: .....

.....

2. Ưu điểm: .....

.....

3. Khuyết điểm: .....

.....

4. Đề nghị cho bảo vệ hay không ?

.....

5. Đánh giá loại: .....

6. Điểm: .....(Bằng chữ: .....)

Tp. Thủ Đức, ngày 21 tháng 06 năm 2024

Giáo viên phản biện

## LỜI CAM ĐOAN

Chúng tôi xin khẳng định rằng bài luận văn tốt nghiệp này là kết quả làm việc nghiêm túc và trung thực của nhóm chúng tôi, gồm các thành viên Trương Quốc Tín và Doãn Đức Minh. Nội dung và thành phẩm trong luận văn đều là thành quả từ sự nỗ lực không ngừng nghỉ của chúng tôi. Từ những chỉ dẫn chu đáo và đóng góp hết sức quý báu từ Ths. Nguyễn Ngô Lâm, chúng tôi đã hoàn thành bài luận văn này dựa trên năng lực và sự cố gắng của chính mình, không lấy cắp từ công trình nghiên cứu hay bài viết nào khác.

Tác giả thực hiện đề tài

Trương Quốc Tín      Doãn Đức Minh

## LỜI CẢM ƠN

Để đồ án tốt nghiệp ngành Kỹ thuật Điện tử - Viễn thông được hoàn thành, nhóm xin bày tỏ lòng tri ân sâu sắc tới đội ngũ giảng viên tại trường Đại học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh. Đặc biệt, nhóm muốn gửi lời cảm ơn chân thành đến thầy Nguyễn Ngô Lâm, người đã nhiệt tình chỉ dẫn, cung cấp những kiến thức thiết yếu và tạo môi trường thuận lợi cho nhóm trong suốt quá trình thực hiện đề tài.

Nhóm cũng muốn bày tỏ sự biết ơn đối với bạn bè đã hỗ trợ, chia sẻ kinh nghiệm quý giá và đóng góp những ý kiến hữu ích, giúp nhóm hoàn thành đề tài một cách tốt nhất. Bên cạnh đó, nhóm vô cùng cảm kích sự hỗ trợ từ gia đình, những người đã tạo điều kiện thuận lợi về cả tinh thần lẫn vật chất, để mỗi thành viên có thể tập trung hoàn thành nhiệm vụ của mình.

Mặc dù đã cố gắng hết sức, nhóm nhận thức rằng vẫn có thể còn những thiếu sót. Do đó, nhóm rất mong nhận được những góp ý quý báu từ quý Thầy Cô để có thể cải thiện và tích lũy kinh nghiệm cho các dự án tương lai.

Nhóm xin gửi lời chúc sức khỏe dồi dào, nhiệt huyết và thành công trong sự nghiệp giáo dục cao quý đến quý Thầy Cô.

Nhóm xin chân thành cảm ơn!

Nhóm thực hiện đề tài

Trương Quốc Tín      Doãn Đức Minh

## TÓM TẮT

Với sự phát triển không ngừng của xã hội, nhu cầu mua sắm của người dân ngày càng tăng cao, dẫn đến số lượng người đến siêu thị ngày càng nhiều. Tuy nhiên, một vấn đề thường gặp khiến nhiều khách hàng không thoái mái là phải xếp hàng dài chờ thanh toán. Có hai nguyên nhân chính gây ra tình trạng này: Thứ nhất, vào các giờ cao điểm, siêu thị quá đông khách, dù đã bố trí nhiều máy tính tiền nhưng vẫn không kịp phục vụ nhanh chóng cho tất cả mọi người. Thứ hai, vào các giờ thấp điểm, siêu thị giảm số lượng máy tính tiền hoạt động để tiết kiệm chi phí thuê nhân viên thu ngân, dẫn đến việc dù lượng khách ít nhưng vẫn phải chờ lâu để thanh toán.

Nhận thấy vấn đề trên, nhóm chúng tôi quyết định nghiên cứu và thiết kế một hệ thống máy tính tiền tự động. Hiện tại, hệ thống này có khả năng nhận diện ba loại quả: táo, chanh và lê. Khi khách hàng đặt sản phẩm lên bàn cân, hệ thống sẽ tự động hiển thị thông tin sản phẩm như: tên, khối lượng, giá niêm yết và tổng số tiền cần thanh toán. Sau khi quét tất cả các sản phẩm qua camera, khách hàng chỉ cần nhấn nút thanh toán trên màn hình, mã QR sẽ xuất hiện để khách hàng thanh toán qua mạng mà không cần sử dụng tiền mặt.

Hệ thống này không chỉ giúp khách hàng thanh toán một cách nhanh chóng và tiện lợi hơn mà còn giúp các siêu thị và cửa hàng tiện lợi tiết kiệm chi phí thuê nhân viên thu ngân. Đây sẽ là tiền đề cho các máy tính tiền tự động với nhiều cải tiến hơn trong tương lai, phù hợp cho các siêu thị và cửa hàng tiện lợi, giúp khách hàng có trải nghiệm mua sắm thoải mái và thuận tiện hơn.

## MỤC LỤC

DANH MỤC HÌNH ẢNH .....	9
DANH MỤC BẢNG .....	12
DANH MỤC TỪ VIẾT TẮT .....	13
CHƯƠNG 1: TỔNG QUAN .....	15
1.1 Lý do chọn đề tài .....	15
1.2 Tính cấp thiết của đề tài .....	15
1.3 Mục tiêu nghiên cứu.....	16
1.4 Nhiệm vụ nghiên cứu.....	16
1.5 Đối tượng và phạm vi nghiên cứu.....	16
1.5.1 Đối tượng nghiên cứu .....	16
1.5.2 Phạm vi nghiên cứu .....	17
1.6 Phương pháp nghiên cứu.....	17
1.7 Bố cục đồ án.....	18
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	19
2.1. Các công nghệ liên quan.....	19
2.1.1. Nhận dạng hình ảnh .....	19
2.1.2. Xử lý ảnh.....	19
2.2. YOLO.....	21
2.2.1. Khái niệm [5].....	21
2.2.2. Đặc điểm .....	21
2.2.3. Đánh giá hiệu suất.....	22

2.2.4. Kiến trúc YOLO .....	23
2.2.5. Quy trình nhận dạng hộp giới hạn.....	24
2.3 Sự phát triển của YOLO.....	25
2.4 Các linh kiện sử dụng cho phần cứng.....	31
2.4.1 Raspberry Pi 4 [7].....	31
2.4.2 ESP32 [10] .....	33
2.4.3 Các cảm biến và linh kiện sử dụng .....	34
2.5 Các chuẩn giao tiếp.....	43
2.5.1 Chuẩn giao tiếp UART [24] .....	43
2.5.2 Chuẩn giao tiếp I2C [24].....	45
2.6 Giới thiệu về IoTs và các ứng dụng.....	47
2.7 Nền tảng về Django Rest Framework.....	47
2.7.1 Tổng quan về Django Rest Framework [3].....	47
2.7.2 Ứng dụng Django vào việc xử lý khi người dùng yêu cầu truy cập bên trong ứng dụng [3].....	48
2.7.3 Giao tiếp giữa máy chủ với cơ sở dữ liệu [4] .....	49
2.8 Tổng quan về các công cụ hỗ trợ xây dựng vào lập trình cho hệ thống.....	50
2.8.1 Proteus.....	50
2.8.2 Arduino IDE .....	51
2.8.3 Visual Studio Code .....	52
CHƯƠNG 3: YOLOv8, THU VIỆN OPENCV VÀ PYZBAR .....	54
3.1. Giới thiệu về YOLOv8 .....	54

3.1.1. Lịch sử phát triển [5] .....	54
3.1.2. Ưu điểm và nhược điểm .....	54
3.1.3. Kiến trúc mạng [5] .....	55
3.1.4 Ứng dụng của YOLOv8 .....	55
3.2. Cấu trúc của YOLOv8 .....	55
3.2.1. Backbone .....	55
3.2.2. Neck.....	55
3.2.3. Head.....	56
3.3. Quá trình huấn luyện YOLOv8 [12][19] .....	56
3.3.1. Chuẩn Bị Dữ Liệu Huấn Luyện .....	56
3.3.2. Chuẩn bị dữ liệu (Tạo Dataset) .....	56
3.3.3. Quá trình huấn luyện .....	57
3.4. Đánh giá hiệu quả của mô hình YOLOv8 .....	58
3.4.1. Các chỉ số đánh giá.....	58
3.4.2. Kết quả đánh giá.....	59
3.5 Thư viện Opencv [21].....	59
3.6 Thư viện Pyzbar [22] .....	60
3.7 YOLOv8 trong máy tính toán tự động.....	61
3.7.1 Phương pháp nhận diện vật thể và mã vạch dựa vào yolov8 kết hợp opencv và pyzbar .....	61
3.7.2 Quá trình nhận diện vật thể .....	61
3.7.3 Quá trình nhận diện mã vạch .....	62

CHƯƠNG 4: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG ..... 63

4.1 Yêu cầu hệ thống .....	63
4.2 Sơ đồ khối hệ thống và chức năng từng khối .....	63
4.2.1 Sơ đồ khối hệ thống .....	63
4.2.2 Chức năng từng khối .....	63
4.3 Hoạt động của hệ thống.....	65
4.4 Thiết kế phần cứng.....	65
4.4.1. Khối cảm biến khói lượng .....	65
4.4.2. Khối hiển thị.....	67
4.4.3. Khối nút nhấn.....	68
4.4.4. Camera Pi Module.....	69
4.4.5. Khối cảm biến khoảng cách .....	71
4.4.6. Khối LED và BUZZER báo trạng thái.....	72
4.4.7. Khối xử lý trung tâm 1 .....	73
4.4.8. Khối xử lý trung tâm 2.....	74
4.4.9. Khối nguồn .....	75
4.5 Sơ đồ nguyên lý toàn hệ thống .....	76
4.6 Thiết kế phần mềm.....	77
4.6.1 Lưu đồ giải thuật chương trình chính.....	77
4.6.2 Lưu đồ giải thuật chương trình con gửi địa chỉ API .....	78
4.6.3 Lưu đồ giải thuật chương trình ESP32 .....	79
4.6.4 Lưu đồ giải thuật chương trình con từ địa chỉ API gửi tới server .....	80

4.6.5 Lưu đồ giải thuật hoạt động của giao diện Web.....	81
<b>CHƯƠNG 5: KẾT QUẢ THỰC HIỆN HỆ THỐNG.....</b>	<b>82</b>
5.1 Mô hình hệ thống .....	82
5.2 Kết quả nhận diện trái cây .....	84
5.2 Kết quả nhận diện mã vạch.....	90
5.3 Kết quả hoạt động của hệ thống.....	90
5.4 Giao diện người dùng .....	96
5.4 Giao diện cho ADMIN .....	98
5.5 Terminal địa chỉ IP gửi server.....	100
5.6 Nhận xét, đánh giá hệ thống .....	101
<b>CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>102</b>
6.1 Kết luận .....	102
6.2 Hạn chế và hướng phát triển.....	103
6.2.1 Hạn chế .....	103
6.2.2 Hướng phát triển.....	104
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>105</b>
Sách tham khảo: .....	105
Datasheet tham khảo: .....	105
Bài viết tham khảo:.....	105
<b>PHỤ LỤC 1 .....</b>	<b>107</b>
<b>PHỤ LỤC 2 .....</b>	<b>108</b>

## **DANH MỤC HÌNH ẢNH**

Hình 2. 1. Các lĩnh vực nghiên cứu chính trong xử lý ảnh số .....	19
Hình 2. 2. Tính toán IoU .....	22
Hình 2. 3. Lớp tích chập .....	23
Hình 2. 4. Cấu trúc YOLO .....	24
Hình 2. 5. Dòng thời gian phát triển YOLO .....	25
Hình 2. 6. So sánh tốc độ YOLOv4 với YOLOv3 .....	26
Hình 2. 7. UNA (Kiến trúc mạng hợp nhất) .....	27
Hình 2. 8. Cấu trúc của YOLOv5 .....	28
Hình 2. 9. Khung YOLOv6 .....	29
Hình 2. 10. So sánh YOLOv8 với các mẫu YOLO mới nhất .....	30
Hình 2. 11. Hình ảnh mô phỏng và thực tế của Raspberry Pi 4 .....	31
Hình 2. 12. Hình ảnh mô phỏng và thực tế của ESP32 .....	34
Hình 2. 13. Hình ảnh mô phỏng và thực tế của module ADC HX711 .....	35
Hình 2. 14. Hình ảnh mô phỏng và thực tế của loadcell 5kg .....	36
Hình 2. 15. Hình ảnh mô phỏng và thực tế của màn hình LCD .....	37
Hình 2. 16. Hình ảnh mô phỏng và thực tế của module I2C .....	38
Hình 2. 17. Hình ảnh mô phỏng và thực tế của camera Pi module .....	40
Hình 2. 18. Cảm biến khoảng cách HC-SR04 .....	41
Hình 2. 19. Buzzer 5V .....	42
Hình 2. 20. Led siêu sáng 10mm .....	42
Hình 2. 21. Giao tiếp UART .....	43
Hình 2. 22. Gói dữ liệu của chuẩn giao tiếp UART .....	44
Hình 2. 23. Gói dữ liệu của chuẩn giao tiếp I2C .....	45
Hình 2. 24. Giao tiếp I2C giữa một Master và nhiều Slave .....	46
Hình 2. 25. Mô hình MVT (Model – View – Template) .....	49
Hình 2. 26. Giao diện màn hình chính của phần mềm Proteus .....	50
Hình 2. 27. Giao diện chính của phần mềm Arduino IDE .....	51
Hình 2. 28. Giao diện chính của phần mềm Visual Studio Code .....	52

Hình 3. 1. Tạo khung đối tượng .....	57
Hình 3. 2. Mô hình mạng lưới neural đơn giản.....	58
Hình 3. 3. Thư viện OpenCV .....	60
Hình 4. 1. Sơ đồ khói hệ thống.....	63
Hình 4. 2. Sơ đồ nguyên lý kết nối cảm biến khói lượng với ESP32 .....	67
Hình 4. 3. Sơ đồ nguyên lý kết nối màn hình LCD với ESP32 .....	68
Hình 4. 4. Sơ đồ nguyên lý kết nối nút nhấn với Raspberry Pi .....	69
Hình 4. 5. Sơ đồ nguyên lý kết nối Camera Pi với Raspberry Pi .....	70
Hình 4. 6. Miêu tả kết nối của khói cảm biến khoảng cách.....	72
Hình 4. 7. Miêu tả kết nối của khói Led và Buzzer báo trạng thái. ....	73
Hình 4. 8. Sơ đồ chân của ESP32 .....	74
Hình 4. 9. Sơ đồ chân của Raspberry Pi.....	75
Hình 4. 10. Sơ đồ nguyên lý nguồn 5V.....	76
Hình 4. 11. Lưu đồ giải thuật chương trình chính.....	77
Hình 4. 12. Lưu đồ giải thuật chương trình con gửi API.....	78
Hình 4. 13. Lưu đồ giải thuật chương trình ESP32.....	79
Hình 4. 14. Lưu đồ giải thuật từ địa chỉ API gửi đến server.....	80
Hình 4. 15. Lưu đồ giải thuật hoạt động WEB .....	81
Hình 5. 1. Mô hình chụp tổng quát từ phía trước.....	82
Hình 5. 2. Mô hình được chụp từ trên xuống.....	82
Hình 5. 3. Mô hình được chụp bên trong không gian nhận diện .....	83
Hình 5. 4. Kết quả nhận diện Cam vàng bằng camera qua cân 1 và cân 2 .....	84
Hình 5. 5. Kết quả hiển thị nhận diện Cam vàng trên terminal ở cân 1 và cân 2 .....	84
Hình 5. 6. Kết quả nhận diện Cam xanh bằng camera qua cân 1 và cân 2.....	85
Hình 5. 7. Kết quả hiển thị nhận diện Cam xanh trên terminal ở cân 1 và cân 2 .....	85
Hình 5. 8. Kết quả nhận diện Chuối bằng camera.....	86
Hình 5. 9. Kết quả hiển thị nhận diện Chuối trên terminal.....	86
Hình 5. 10. Kết quả nhận diện Mân cầu bằng camera.....	87
Hình 5. 11. Kết quả hiển thị nhận diện Mân cầu trên terminal .....	87

Hình 5. 12. Kết quả nhận diện Táo bằng camera .....	88
Hình 5. 13. Kết quả hiển thị nhận diện Táo trên terminal .....	88
Hình 5. 14. Kết quả nhận diện Xoài bằng camera.....	89
Hình 5. 15. Kết quả hiển thị nhận diện Xoài trên terminal.....	89
Hình 5. 16. Màn hình nhận diện mã vạch .....	90
Hình 5. 17. Terminal nhận diện mã mạch .....	90
Hình 5. 18. Đưa trái cây vào trong bàn cân .....	91
Hình 5. 19. Khối lượng của sản phẩm khi đặt lên bàn cân.....	91
Hình 5. 20. Nhấn nút nhấn 1 .....	92
Hình 5. 21. Nhấn nút nhấn 3 .....	92
Hình 5. 22. Hiển thị thông tin sản phẩm.....	93
Hình 5. 23. Nhấn nút nhấn 2 .....	93
Hình 5. 24. Đưa vật phẩm và quay đầu mã vạch về hướng camera .....	94
Hình 5. 25. Hiển thị thông tin sản phẩm.....	95
Hình 5. 26. Mã QR thanh toán .....	95
Hình 5. 27. Giao diện hiển thị đã thanh toán.....	96
Hình 5. 28. Giao diện người dùng .....	96
Hình 5. 29. Giao diện hiển thị thông tin sản phẩm.....	97
Hình 5. 30. Giao diện web cho người dùng thanh toán.....	97
Hình 5. 31. Giao diện web khi người dùng thanh toán thành công. ....	98
Hình 5. 32. Giao diện ADMIN .....	99
Hình 5. 33. Giao diện quản lý sản phẩm.....	99
Hình 5. 34. Giao diện thông tin sản phẩm .....	100
Hình 5. 35. Terminal hiển thị thông tin sản phẩm.....	101

## **DANH MỤC BẢNG**

Bảng 2. 1. Thông số kỹ thuật của Raspberry Pi 4 .....	31
Bảng 2. 2. Thông số các chân của Raspberry Pi 4 được sử dụng.....	33
Bảng 2. 3. Thông số kỹ thuật của ESP32.....	34
Bảng 2. 4. Thông số các chân của module ADC HX711 .....	35
Bảng 2. 5. Thông số kỹ thuật của Loadcell 5kg.....	37
Bảng 2. 6. Thông số kỹ thuật của màn hình LCD .....	38
Bảng 2. 7. Thông số các chân của màn hình LCD.....	38
Bảng 2. 8. Thông số kỹ thuật của Camera Pi .....	40
Bảng 2. 9. Thông số kỹ thuật cảm biến khoảng cách HC-SR04.....	41
Bảng 2. 10. Chức năng các chân cảm biến HC-SR04 .....	41
Bảng 2. 11. Thông số kỹ thuật của buzzer 5V .....	42
Bảng 2. 12. Thông số kỹ thuật của led siêu sáng 10mm .....	43
Bảng 4. 1.Thông số về điện áp, dòng điện tiêu thụ của các linh kiện cảm biến .....	75

## DANH MỤC TỪ VIẾT TẮT

HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu siêu văn bản
CSS	Cascading Style Sheets	Bảng điều khiển kiểu trang
IoT	Internet of Things	Mạng lưới Vạn vật kết nối
Wi-Fi	Wireless Fidelity	Kết nối không dây
SPI	Serial Peripheral Interface	Giao diện Ngoại vi Nối tiếp
I2C	Inter-Integrated Circuit	Truyền thông nối tiếp, Bán bán song công
UART	Universal Asynchronous Receiver/Transmitter	Truyền/nhận không đồng bộ đa năng
GPIO	General Purpose Input/Output	Đầu vào/đầu ra tổng quát
ADC	Analog-to-Digital Converter	Bộ chuyển đổi từ tín hiệu analog sang số
USB	Universal Serial Bus	Giao tiếp Nối dây Chuẩn
RAM	Random Access Memory	Bộ Nhớ Truy Cập Ngẫu Nhiên
LED	Light Emitting Diode	Điốt Phát Quang
PCB	Printed Circuit Board	Bảng Mạch In
IDE	Integrated Development Environment	Môi trường Phát triển Tích hợp
ID	Identification	Nhận dạng
LCD	Liquid Crystal Display	Công nghệ màn hình tinh thể lỏng
IC	Integrated Circuit	Vì Mạch Tích Hợp
TX	Transmitter	Cổng dùng để gửi dữ liệu
RX	Receiver	Cổng dùng để nhận dữ liệu
QR	Quick Response	Mã vạch phản hồi nhanh
YOLO	You Only Look Once	Thuật toán nhận diện
CNNs	Convolutional Neural Network	Mạng nơ-ron tích chập

IoU	Intersection over Union	Phép đo trong xử lý ảnh
DRF	Django Rest Framework	Công cụ hỗ trợ đắc lực trong việc xây dựng WebAPI
ORM	Object-Relational Mapping	Kỹ thuật lập trình tiên tiến giúp ánh xạ cấu trúc của cơ sở dữ liệu
PAN	Path Aggregation Network	Một phiên bản cải tiến của FPN

## CHƯƠNG 1: TỔNG QUAN

### 1.1 Lý do chọn đề tài

Chúng ta đang ở trong thời đại mà nền kinh tế đang phát triển nhanh chóng, chính vì thế sự đơn giản hóa và tiện lợi là nhu cầu của người dân. Việc mua sắm là hoạt động diễn ra hàng ngày không thể thiếu, thúc đẩy sự phát triển của các siêu thị và việc ứng dụng máy tính tiền trong hoạt động kinh doanh trở nên phổ biến.

Mặc dù như thế, nhưng việc sử dụng máy tính tiền cũng gặp phải một số trở ngại, việc tốn thời gian của khách hàng trong quá trình thanh toán. Ngay cả khi không có nhiều khách hàng, thời gian hoàn tất thanh toán vẫn có thể kéo dài. Đó là do việc các doanh nghiệp giảm bớt nhân viên thu ngân để cắt giảm chi phí.

Cảm thấy vấn đề này cấp thiết, chính vì vậy nhóm chúng tôi đã đưa ra một giải pháp cũng như là chọn đề tài "**Ứng dụng xử lý ảnh và Yolo trong hệ thống thanh toán tự động**" cho đồ án tốt nghiệp của mình.

### 1.2 Tính cấp thiết của đề tài

Hầu hết trong các siêu thị ở Việt Nam vẫn dựa vào nhân viên thu ngân để tính hàng hóa mua cho khách hàng và thường là sử dụng mã vạch để quét và phân biệt sản phẩm. Tuy nhiên bên cạnh việc sử dụng mã vạch đôi khi cũng gặp sự cố và nó phù hợp với các sản phẩm tính theo đơn vị số lượng hơn, các sản phẩm về khối lượng bị hạn chế hơn. Điều này đã thúc đẩy chúng tôi tạo ra một thiết bị mà vừa có thể thanh toán bằng mã vạch vừa có thể thanh toán bằng cách nhận diện AI, giúp khách hàng có thể tự thanh toán nhanh chóng với 2 cân nhanh chóng, tiết kiệm thời gian.

Từ đó chúng tôi có thể đưa ra những lợi ích của hệ thống này như sau:

- Giảm thời gian chờ đợi cho khách hàng tại quầy thanh toán khi có hai cân hoạt động.
- Hệ thống thao tác đơn giản mang lại sự thuận tiện và tạo ra sự thoải mái tối đa trong quá trình thanh toán.
- Hạn chế rủi ro sai sót và gian lận trong các giao dịch.

- Nâng cao độ chính xác trong quản lý hàng hóa và ghi nhận giao dịch.
- Giúp doanh nghiệp tiết kiệm chi phí về nhân sự mà vẫn đáp ứng tốt nhu cầu của khách hàng.

### **1.3 Mục tiêu nghiên cứu**

Hệ thống thanh toán tự động mà nhóm chúng tôi đang phát triển sẽ mang lại nhiều lợi ích thiết thực, từ việc cắt giảm chi phí nhân sự cho cửa hàng và siêu thị đến các tính năng tự động nhận diện sản phẩm qua camera và mã vạch, đo khối lượng bằng việc sử dụng cân khối lượng có tích hợp thêm bộ chuyển đổi từ tín hiệu tương tự sang tín hiệu số. Hệ thống này xử lý thông tin từ camera và loadcell để tính toán giá trị sản phẩm và hiển thị lên màn hình cho người dùng. Ngoài ra, hệ thống hỗ trợ thanh toán bằng mã QR và mã vạch, giúp quá trình thanh toán trở nên nhanh chóng và tiện lợi. Với việc trang bị hai cân để tăng cường hiệu quả, hệ thống của chúng tôi hứa hẹn mang đến trải nghiệm mua sắm tối ưu.

### **1.4 Nhiệm vụ nghiên cứu**

- Đồ án được hoàn thành đúng hạn.
- Các tiêu chí đặt ra của hệ thống hoạt động ổn định và đúng yêu cầu.
- Giao diện được hiển thị trên màn hình là các thông tin mà người sử dụng dễ nhìn, dễ sử dụng.
- Thiết kế mô hình gọn, sạch, đẹp, tối ưu.

### **1.5 Đối tượng và phạm vi nghiên cứu**

#### **1.5.1 Đối tượng nghiên cứu**

Nhóm đang tập trung nghiên cứu các linh kiện như Esp32, Raspberry Pi4, Camera Pi Module, Loadcell, LCD 16x2 – I2C và các chuẩn giao tiếp sử dụng. Trong khi đó, về phần mềm, chọn các ngôn ngữ lập trình như Python, HTML, CSS, Javascript và sử dụng các công cụ như Proteus, Arduino IDE và Visual Studio Code để hỗ trợ việc thiết kế và lập trình.

Mô hình mà nhóm đang nghiên cứu bao gồm cách hoạt động của các máy tính tiền phổ

bien hiện nay, cũng như việc thiết kế hệ thống và bố trí linh kiện sao cho hợp lý, gọn gàng và thẩm mỹ.

### **1.5.2 Phạm vi nghiên cứu**

Giới hạn nghiên cứu của đề tài ở việc xây dựng hệ thống tính tiền tự động có khả năng nhận diện 6 loại quả khác nhau (có thể thêm nhiều loại quả khác) và nhận diện mã vạch, cho phép người mua hàng tự thực hiện thanh toán và trả phí qua mã QR. Cụ thể, việc nghiên cứu sẽ chú trọng vào:

- Kết nối và lập trình cho Raspberry Pi4 và Esp32.
- Lý thuyết về kết nối Loadcell Module và Camera Pi Module.
- Lý thuyết cơ bản lập trình C, Python, HTML, CSS và Javascript
- Thuật toán nhận diện hình ảnh, cấu trúc dữ liệu và tạo mã QR kết nối với trang Web.

### **1.6 Phương pháp nghiên cứu**

Nghiên cứu về cơ sở lý thuyết:

- Tham khảo tài liệu và thông tin mô tả về các linh kiện trong hệ thống để hiểu rõ tính năng và cơ chế hoạt động của chúng.
- Sử dụng nền tảng kiến thức đã học cùng với thông tin thu thập được để tiến hành mô phỏng hệ thống, kiểm tra và xác định tính khả thi của nó.
- Nghiên cứu các sản phẩm tương tự trên thị trường, đánh giá những ưu và nhược điểm của chúng nhằm đề xuất các cải tiến cho hệ thống.

Thực hiện nghiên cứu mô phỏng:

- Mô phỏng từng khối của hệ thống để kiểm tra và điều chỉnh các thông số sao cho đáp ứng được các yêu cầu đã đề ra.
- Các khối được kết hợp lại với nhau để tạo thành một mô hình thống nhất hoàn

chỉnh và tiến hành kiểm tra và điều chỉnh nếu cần.

## 1.7 Bố cục đồ án

**Chương 1: Tổng quan:** Đưa ra tình hình hiện tại và nêu bật tầm quan trọng và sự cấp thiết của đề tài. Giải thích nguyên do chọn đề tài và đề tài này được ứng dụng như thế nào trong thực tiễn.

**Chương 2: Cơ sở lý thuyết:** Trình bày giới thiệu các khái niệm và lý thuyết cơ bản được áp dụng trong đề tài.

**Chương 3: Yolov8 và thư viện OpenCV, Pyzbar:** Trình bày tổng quan về khái niệm, đặc điểm, tính năng và cách vận hành của Yolov8, cũng như giới thiệu chung về thư viện OpenCV và Pyzbar và cách kết hợp với nhau.

**Chương 4: Thiết kế hệ thống:** Phác thảo sơ đồ khối của hệ thống, chi tiết thiết kế từng khối, lựa chọn linh kiện, tính toán cần thiết, xây dựng lưu đồ giải thuật và đề xuất phương pháp vận hành hệ thống từ những tìm hiểu lý thuyết trước đó.

**Chương 5: Kết quả:** Trình bày kết quả thu được, triển khai phần cứng thực tế, đánh giá tính ổn định của hệ thống và chỉ ra những hạn chế còn tồn tại.

**Chương 6: Kết luận và hướng phát triển:** Trình bày các kết quả thu được và đưa ra dự kiến phát triển đề tài thời gian tới.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Các công nghệ liên quan

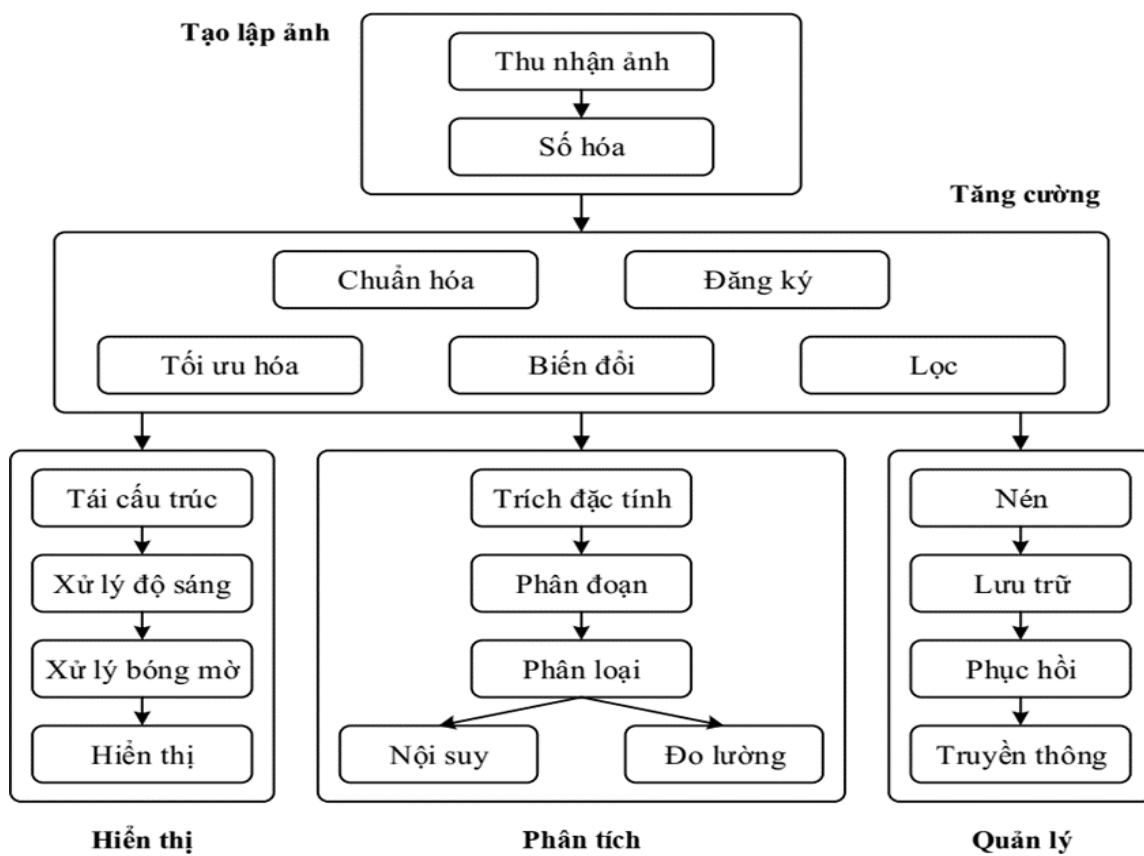
#### 2.1.1. Nhận dạng hình ảnh

Nhận dạng hình ảnh là một lĩnh vực của trí tuệ nhân tạo (AI) cho phép máy tính "nhìn" và hiểu được nội dung của hình ảnh. Các thuật toán nhận dạng hình ảnh có thể phân tích hình ảnh và xác định các đối tượng, cảnh vật, khuôn mặt, v.v. trong hình ảnh.

#### 2.1.2. Xử lý ảnh

##### a. Khái niệm [1]

Xử lý ảnh là việc áp dụng các kỹ thuật và phương pháp để phân tích, chuyển đổi và cải thiện chất lượng hình ảnh. Quá trình này thường nhằm trích xuất thông tin quan trọng từ hình ảnh hoặc nâng cao hình ảnh để đáp ứng các mục đích cụ thể. Lĩnh vực xử lý ảnh bao gồm nhiều khía cạnh khác nhau, nhưng có thể được chia thành bốn lĩnh vực chính như hình 2.1: tạo lập ảnh, biểu diễn, phân tích và quản lý



Hình 2. 1. Các lĩnh vực nghiên cứu chính trong xử lý ảnh số

- **Tạo lập ảnh:** bao gồm quá trình từ thu nhận hình ảnh đến tạo ra một biểu diễn số của ảnh dưới dạng ma trận.
- **Hiển thị:** đòi hỏi các phương pháp xử lý trên ma trận sao cho hiệu suất là tối ưu.
- **Phân tích:** sử dụng các kỹ thuật và công cụ để trích xuất thông tin từ hình ảnh. Quá trình này bao gồm nhận diện các đặc điểm, đối tượng, hoặc mẫu trong ảnh và hiểu nội dung của chúng.
- **Quản lý:** bao gồm các kỹ thuật lưu trữ, truyền thông, phục hồi và truy nhập dữ liệu hiệu quả. Trong lĩnh vực y học, việc lấy thông tin hình ảnh từ xa cũng được coi là một phần của quản lý hình ảnh.

### b. Các vấn đề trong xử lý ảnh [1] [2]

**Điểm ảnh:** Đơn vị cơ bản nhất của ảnh số là pixel, mỗi pixel chứa thông tin về màu sắc và độ sáng tại một vị trí cụ thể trên ảnh, được xác định bằng tọa độ (x, y). Độ phân giải của ảnh tăng khi có nhiều pixel hơn, từ đó chứa nhiều thông tin hơn.

**Ảnh số:** tập hợp từ vô số điểm ảnh, còn gọi là pixel, mỗi pixel biểu diễn một màu sắc hoặc độ sáng tại một vị trí cụ thể trên ảnh. Các pixel có thể được xem như các chấm nhỏ trên một tấm ảnh màu. Ảnh số được biểu diễn dưới dạng ma trận hai chiều, trong đó giá trị của mỗi pixel đại diện cho độ sáng (trong ảnh đen trắng) hoặc màu sắc (trong ảnh màu).

**Phân loại ảnh:** có hai dạng chính của ảnh số: ảnh màu và ảnh đen trắng (hay ảnh xám). Trong ảnh màu, mỗi pixel có thể mang các giá trị màu sắc khác nhau, trong khi ảnh đen trắng được tạo thành từ các pixel có giá trị mức xám khác nhau.

**Biểu diễn ảnh:** là cách thức mà một hình ảnh được lưu trữ và xử lý trong máy tính. Có nhiều cách khác nhau để biểu diễn ảnh, mỗi cách có ưu nhược điểm riêng. Dưới đây là một số cách biểu diễn ảnh phổ biến:

- **Biểu diễn ảnh pixel:** Đây là cách biểu diễn ảnh đơn giản nhất. Mỗi pixel trong ảnh được biểu diễn bằng một giá trị số, thường là từ 0 đến 255. Giá trị này thể hiện độ sáng của pixel, với 0 là màu đen và 255 là màu trắng.

- Biểu diễn ảnh vector: Biểu diễn ảnh vector sử dụng các đường thẳng và đường cong để biểu diễn các đối tượng trong ảnh. Mỗi đường được xác định bởi một tập hợp các điểm và một màu sắc.
- Biểu diễn ảnh bitmap: Biểu diễn ảnh bitmap là sự kết hợp giữa biểu diễn ảnh pixel và biểu diễn ảnh vector. Mỗi pixel trong ảnh được biểu diễn bằng một giá trị số, nhưng giá trị này có thể tham chiếu đến một bảng màu để xác định màu sắc của pixel.

**Phân vùng ảnh:** là một phần quan trọng trong quá trình xử lý ảnh, nó chia ảnh đầu vào thành các vùng có đặc tính tương tự dựa trên biên hoặc các vùng liên thông. Quá trình này hỗ trợ việc biểu diễn, phân tích và nhận dạng các đối tượng trong ảnh.

## 2.2. YOLO

### 2.2.1. Khái niệm [5]

YOLO là một thuật toán nhận dạng đối tượng được phát triển bởi Joseph Redmon, Santosh Divvala, Ross Girshick và Ali Farhadi vào năm 2015. YOLO là một thuật toán nhận dạng đối tượng một giai đoạn, có nghĩa là nó chỉ cần một lần quét qua hình ảnh để dự đoán các đối tượng và vị trí của chúng. Điều này khiến YOLO trở thành một thuật toán nhận dạng đối tượng nhanh và hiệu quả.

Kiến trúc của YOLO bao gồm một mạng nơ-ron tích chập (CNN) được chia thành hai phần:

- Phần trích xuất đặc trưng: Phần này sử dụng mạng nơ-ron VGG-16 để trích xuất các đặc trưng từ hình ảnh.
- Phần dự đoán: Phần này sử dụng các lớp fully-connected để dự đoán các đối tượng và vị trí của chúng.

### 2.2.2. Đặc điểm

- Độ chính xác cao và tốc độ nhanh: YOLO có khả năng đưa ra các dự đoán đồng thời về các đối tượng trong hình ảnh chỉ trong một lần chạy mô hình. Điều này giúp YOLO vượt trội về tốc độ so với các phương pháp truyền thống.

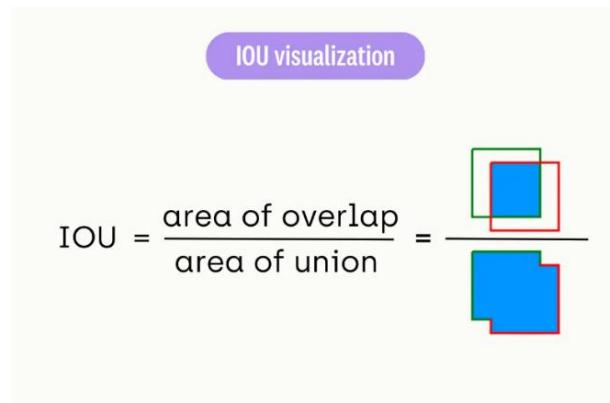
- Sử dụng mạng nơ-ron tích chập (CNNs): YOLO sử dụng mạng nơ-ron tích chập để học và trích xuất đặc trưng từ hình ảnh, giúp cho việc phát hiện đối tượng trở nên hiệu quả và chính xác hơn.
- Phân cấp dự đoán: YOLO chia hình ảnh thành một lưới các ô (grid), và mỗi ô sẽ dự đoán các bounding box và xác suất của các đối tượng trong ô đó. Quá trình này giúp giảm thiểu số lượng dự đoán sai lệch.
- Học từ dữ liệu lớn: Mô hình YOLO được huấn luyện trên các bộ dữ liệu lớn, từ đó học được nhiều đặc điểm của các đối tượng và có khả năng tổng quát hóa tốt trên các tình huống thực tế.
- Hỗ trợ đa lớp: YOLO có thể phát hiện nhiều lớp đối tượng khác nhau trong cùng một hình ảnh, từ người đi bộ đến các phương tiện giao thông và vật dụng.

### 2.2.3. Đánh giá hiệu suất

#### a. Union (IoU)

IoU là một chỉ số phổ biến được sử dụng để đánh giá hiệu suất của các thuật toán phát hiện đối tượng. Nó đo đạc mức độ trùng lắp giữa hộp giới hạn dự đoán (P) và hộp giới hạn thực tế (G). Giá trị IoU được tính bằng tỉ lệ diện tích của phần giao giữa P và G chia cho diện tích của phần hợp của chúng.

IoU có giá trị dao động từ 0 đến 1, trong đó 0 biểu thị không có sự trùng lắp nào và 1 biểu thị sự khớp hoàn hảo giữa hai hộp giới hạn. Giá trị IoU càng cao, cho thấy khả năng phát hiện đối tượng càng chính xác.



Hình 2. 2. Tính toán IoU.

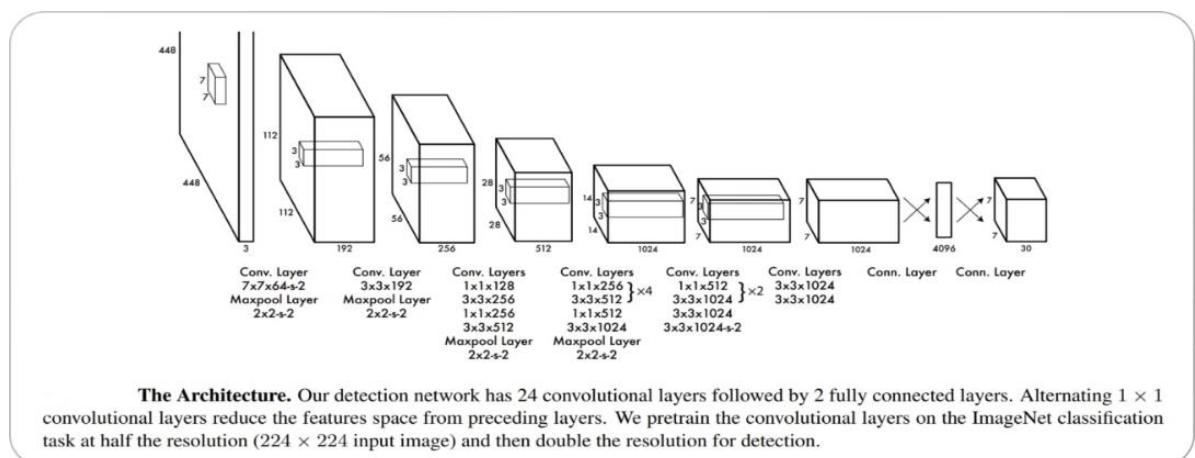
## b. Độ chính xác trung bình (AP)

Độ chính xác trung bình (AP) là một chỉ số quan trọng trong đánh giá các thuật toán phát hiện đối tượng, được tính bằng cách tổng hợp đường cong Precision-Recall tại các ngưỡng phát hiện khác nhau. Precision đo lường tỷ lệ dự đoán đúng so với tổng số dự đoán dương tính, trong khi Recall đo tỷ lệ đối tượng thực tế được dự đoán đúng so với tổng số đối tượng thực tế có trong hình ảnh.

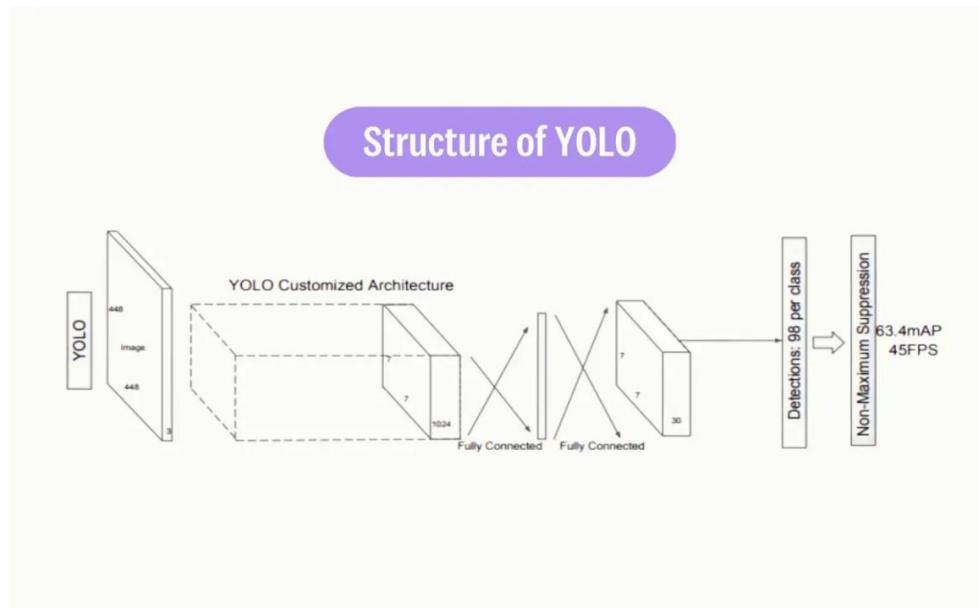
AP tính trung bình các giá trị precision trên các mức recall từ 0 đến 1. Điểm AP dao động từ 0 đến 1, với giá trị càng cao cho thấy hiệu suất càng tốt. Trên thực tế, độ chính xác trung bình trung bình mAP thường được tính bằng cách lấy trung bình của các giá trị AP của các lớp đối tượng khác nhau.

Hiểu rõ các chỉ số này giúp phân tích hiệu suất của các mô hình như YOLO và đưa ra những quyết định thông minh về việc áp dụng chúng trong các bối cảnh thực tế. YOLO đã đổi mới trong lĩnh vực phát hiện đối tượng bằng cách thực hiện cả hai nhiệm vụ phát hiện và phân loại trong một lần duy nhất, mang lại tốc độ và hiệu quả ấn tượng so với các phương pháp truyền thống.

### 2.2.4. Kiến trúc YOLO



Hình 2. 3. Lớp tích chập



Hình 2. 4. Cấu trúc YOLO

Thuật toán YOLO sử dụng một mạng nơ-ron tích chập duy nhất để chia hình ảnh thành một lưới. Mỗi ô trong lưới dự đoán một số hộp giới hạn. Đồng thời với mỗi hộp giới hạn, YOLO cũng dự đoán xác suất của lớp đối tượng, biểu thị khả năng có sự hiện diện của một đối tượng cụ thể trong hộp đó.

### 2.2.5. Quy trình nhận dạng hộp giới hạn

Quá trình nhận dạng khung giới hạn trong YOLO gồm các bước sau:

- Tạo lưới: Hình ảnh được chia thành lưới kích thước SxS. Mỗi ô trong lưới có nhiệm vụ dự đoán một đối tượng được nằm trong tâm ô đó.
- Dự đoán hộp bao: Mỗi ô trong lưới tạo ra B hộp bao và một giá trị tin cậy cho từng hộp. Giá trị tin cậy này biểu thị mức độ tin tưởng của mô hình rằng hộp bao chứa một đối tượng và độ chính xác của dự đoán cho hộp đó.
- Dự đoán xác suất lớp: Ngoài ra, mỗi ô lưới cũng dự đoán xác suất của các lớp đối tượng có điều kiện C (mỗi lớp một xác suất cho các đối tượng có thể). Các xác suất này phụ thuộc vào việc có một đối tượng nằm trong hộp.

## 2.3 Sự phát triển của YOLO



Hình 2. 5. Dòng thời gian phát triển YOLO

**YOLOv1:** Thuật toán phát hiện đối tượng thời gian thực đầu tiên

Mô hình YOLO ban đầu áp dụng phương pháp hồi quy để giải quyết vấn đề phát hiện đối tượng, điều này đánh dấu sự khác biệt lớn so với phương pháp phân loại truyền thống. Nó sử dụng mạng nơ-ron tích chập (CNN) để phát hiện các đối tượng trong hình ảnh bằng cách chia hình ảnh thành lưới ô vuông, mỗi ô vuông trên lưới đưa ra nhiều dự đoán. Sau đó, YOLO loại bỏ các dự đoán có độ tin cậy thấp và giải quyết vấn đề các hộp chồng chéo để tạo ra đầu ra cuối cùng.

**YOLOv2 [YOLO9000]:**

YOLOv2 mang đến nhiều cải tiến so với phiên bản YOLO ban đầu. Một trong những cải tiến quan trọng là việc sử dụng chuẩn hóa hàng loạt (batch normalization) trong tất cả các lớp tích chập của mình. Điều này giúp giảm hiện tượng quá khớp và cải thiện tính ổn định và hiệu suất của mô hình. Đặc biệt, YOLOv2 có khả năng xử lý các hình ảnh có độ phân giải cao hơn, từ đó cải thiện khả năng phát hiện các vật thể nhỏ.

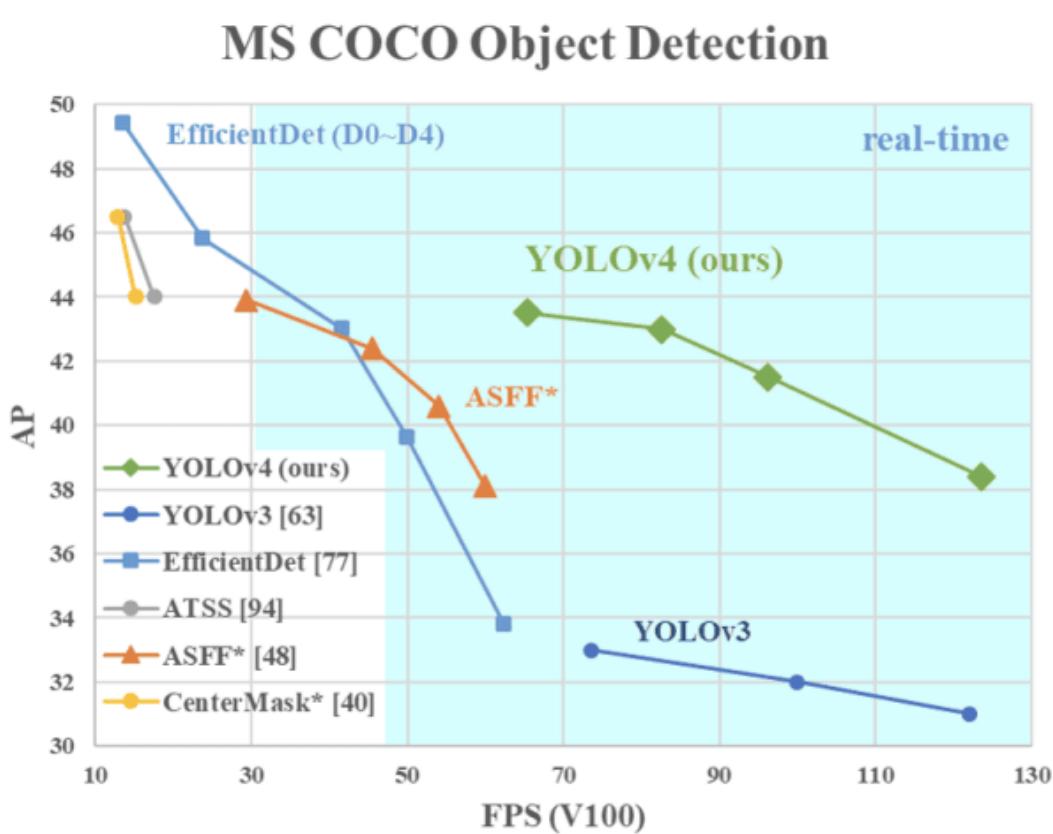
Ngoài ra, YOLOv2 áp dụng các hộp neo (anchors), một khái niệm được mượn từ Faster R-CNN. Các anchors này giúp thuật toán dự đoán hình dạng và kích thước của các vật thể chính xác hơn, cải thiện khả năng phát hiện và định vị đối tượng trong hình ảnh.

Điều này làm cho YOLOv2 trở thành một phiên bản cải tiến, hiệu quả hơn và phù hợp với nhiều ứng dụng thực tế trong lĩnh vực phân loại và định vị đối tượng.

### YOLOv3:

YOLOv3 giới thiệu mạng thần kinh mới Darknet-53, tận dụng các kết nối còn lại để cải thiện hiệu suất. Mô hình cũng đã được điều chỉnh thiết kế để cải thiện độ chính xác mà vẫn duy trì tốc độ. Với độ phân giải 320x320, YOLOv3 có thể chạy trong 22 ms với độ chính xác 28,2 mAP, tương đương với SSD nhưng nhanh hơn ba lần. Trên Titan X, YOLOv3 đạt 57,9 mAP@50 trong 51 ms, so với 57,5 mAP@50 trong 198 ms của RetinaNet, đảm bảo hiệu suất tương tự nhưng nhanh hơn gấp 3,8 lần.

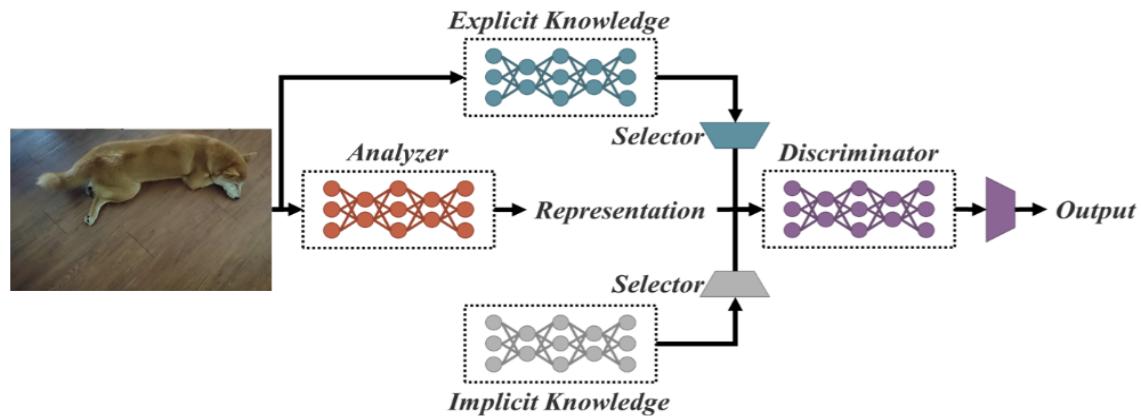
### YOLOv4:



Hình 2. 6. So sánh tốc độ YOLOv4 với YOLOv3

YOLOv4 đã đưa ra nhiều cải tiến mới để cải thiện cả độ chính xác và tốc độ. Mô hình sử dụng CSPDarknet làm xương sống và giới thiệu các kỹ thuật mới như chú ý không gian (Spatial Attention), hàm kích hoạt Mish và mất GIoU để tăng cường hiệu suất. YOLOv4 cải tiến cho thấy độ chính xác trung bình (AP) tăng thêm 0,5% so với phiên bản ban đầu, đồng thời giảm kích thước tệp khối lượng của mô hình xuống còn 45,3 MB.

### **YOLOR:**



Hình 2. 7. UNA (Kiến trúc mạng hợp nhất)

Không giống như các phiên bản YOLO trước đây, kiến trúc và cơ sở hạ tầng mô hình của YOLOR có sự khác biệt đáng kể. Cái tên “YOLOR” nhấn mạnh cách tiếp cận độc đáo của nó: nó kết hợp kiến thức rõ ràng và tiềm ẩn để tạo ra một mạng thống nhất có khả năng xử lý nhiều tác vụ với một đầu vào duy nhất. Bằng cách chỉ học một cách biểu diễn, YOLOR đạt được hiệu suất ấn tượng trong việc phát hiện đối tượng.

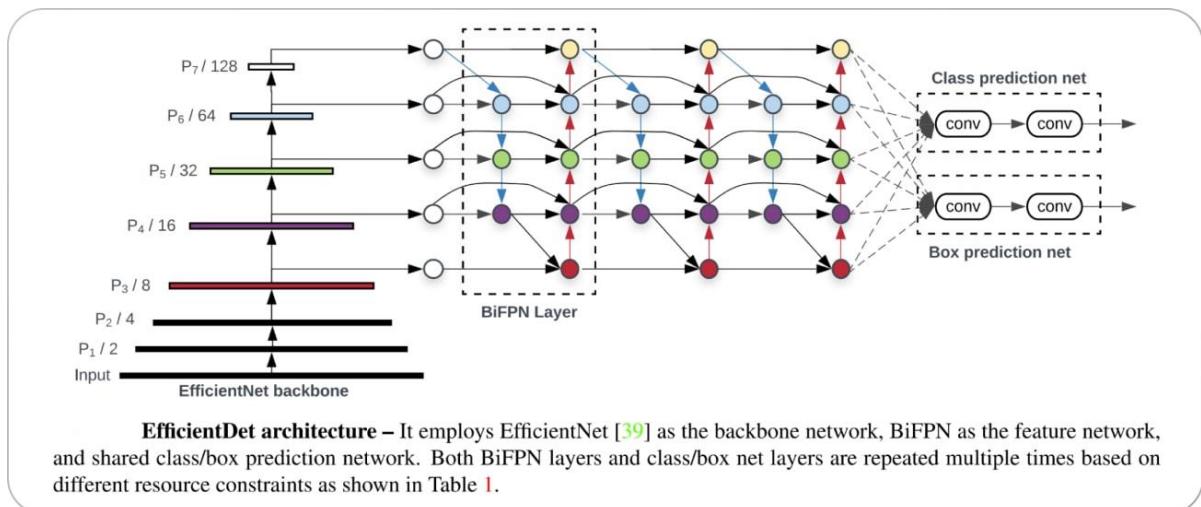
### **YOLOX:**

YOLOX là mô hình phát hiện đối tượng không có neo, được xây dựng trên nền tảng của YOLOv3 SPP với xương sống Darknet53, nhằm mục đích vượt qua hiệu suất của các phiên bản YOLO trước đó. Sự đổi mới quan trọng của YOLOX nằm ở phần đầu tách rời và cách tiếp cận SimOTA. Bằng cách loại bỏ các hộp neo, YOLOX đơn giản hóa thiết kế và đạt được độ chính xác cao hơn. Mô hình này thu hẹp khoảng cách giữa nghiên cứu và công nghiệp, cung cấp giải pháp mạnh mẽ để phát hiện đối tượng theo

thời gian thực. YOLOX có nhiều phiên bản kích cỡ khác nhau, từ YOLOX-Nano nhẹ đến YOLOX-x mạnh mẽ, mỗi phiên bản được thiết kế đặc biệt cho các trường hợp sử dụng khác nhau.

### YOLOv5:

YOLOv5 mang đến những cải tiến hơn nữa để tăng cả độ chính xác và hiệu quả. Nó sử dụng Scaled-YOLOv4 và kết hợp các chiến lược mới như mất CIOU và CSPDarknet53-PANet-SPP để tăng cường độ chính xác.



Hình 2. 8. Cấu trúc của YOLOv5

Thuật toán YOLOv5 đã được tinh chỉnh và chứng minh rằng độ chính xác trung bình trung bình (mAP) tăng thêm 0,7% so với YOLOv4, đồng thời giảm kích thước tệp khối lượng của mô hình xuống còn 53,7 MB. Những cải tiến này đã làm cho YOLOv5 trở thành một công cụ hiệu quả và hiệu quả hơn để phát hiện đối tượng theo thời gian thực.

### YOLOv6:

YOLOv6 đã sử dụng CSPDarknet-X và giới thiệu các phương pháp mới như phân đoạn toàn cảnh, chức năng kích hoạt Swish và mất DIoU để tăng độ chính xác.

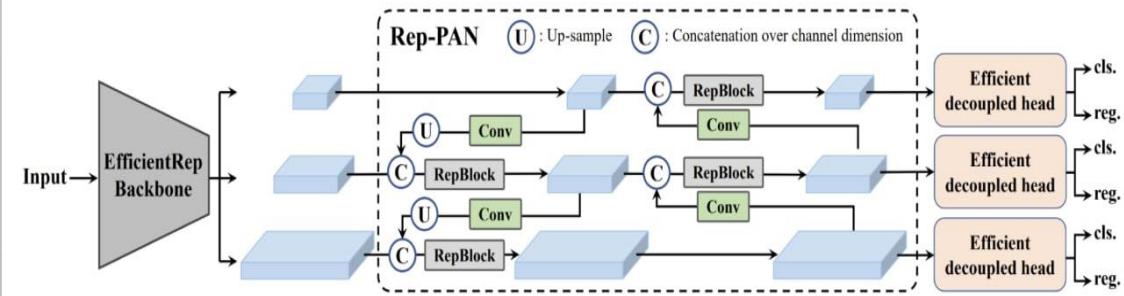


Figure 2: The YOLOv6 framework (N and S are shown). Note for M/L, RepBlocks is replaced with CSPStackRep.

Hình 2. 9. Khung YOLOv6

YOLOv6 đã được cải tiến với việc tăng độ chính xác trung bình (AP) lên 0,8% so với YOLOv5, đồng thời giảm kích thước tệp khói lượng của mô hình xuống còn 60,2 MB. Những cải tiến này làm cho YOLOv6 trở thành một công cụ mạnh mẽ hơn để phát hiện đối tượng trong thời gian thực.

### **YOLOv7:**

YOLOv7 sử dụng CSPDarknet-Z trong kiến trúc của mình. Thuật toán phát hiện đối tượng YOLOv7 đã được cải tiến bằng việc giới thiệu các kỹ thuật nâng cao như phân đoạn lấy đối tượng làm trung tâm, chức năng kích hoạt Leaky ReLU và măt TIoU để cải thiện độ chính xác.

Thuật toán YOLOv7 đã được cải tiến và chứng minh rằng độ chính xác trung bình (AP) tăng thêm 1,0% so với YOLOv6, đồng thời giảm kích thước tệp khói lượng của mô hình xuống còn 70,5 MB. Những cải tiến này đã làm cho YOLOv7 trở thành một công cụ mạnh mẽ hơn để phát hiện đối tượng trong thời gian thực.

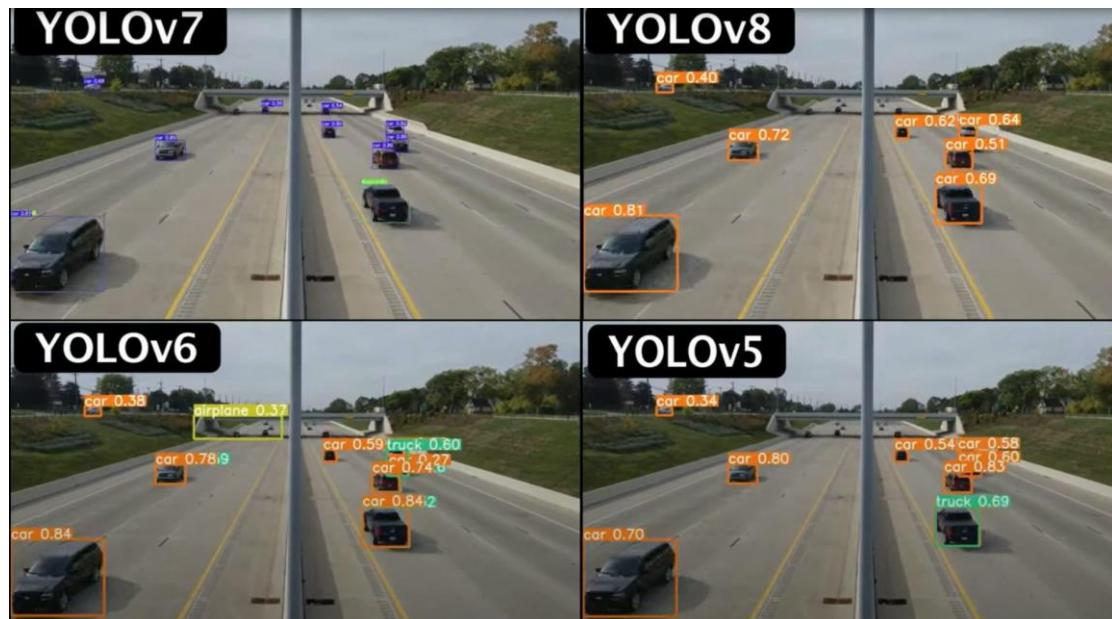
### **YOLOv8:**

YOLOv8 đã giới thiệu kiến trúc mới, CSPDarknet-AA, đây là phiên bản nâng cao của dòng CSPDarknet, hiệu quả và hiệu suất cao trong việc nhận diện đối tượng.

Một kỹ thuật quan trọng được giới thiệu trong YOLOv8 là phát hiện đối tượng đa quy mô. Kỹ thuật này cho phép mô hình phát hiện các đối tượng có kích thước khác nhau trong một hình ảnh. Một cải tiến đáng kể khác trong YOLOv8 là việc sử dụng chức năng kích hoạt ELU. ELU, hay Đơn vị tuyến tính hàm mũ, giúp tăng tốc độ học tập trong mạng lưới thần kinh sâu bằng cách giảm thiểu vấn đề độ dốc biến mất, dẫn đến sự hội tụ nhanh hơn.

YOLOv8 đã chấp nhận mất GIoU. GIoU, hay Giao lô tổng quát trên Liên minh, là phiên bản nâng cao hơn của số liệu IoU có tính đến hình dạng và kích thước của các hộp giới hạn, cải thiện độ chính xác của việc bản địa hóa đối tượng.

Thuật toán YOLOv8 cho thấy độ chính xác trung bình (AP) tăng 1,2% so với YOLOv7, đây là một cải tiến đáng kể. Nó đã đạt được điều này trong khi giảm kích thước tệp khối lượng của mô hình xuống 80,6 M, giúp mô hình hiệu quả hơn và dễ triển khai hơn trong môi trường hạn chế về tài nguyên.



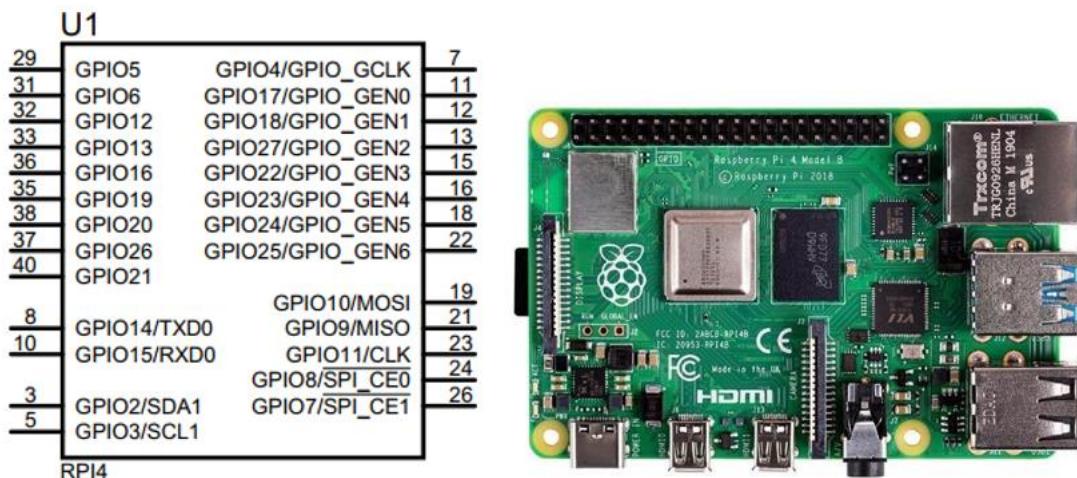
Hình 2. 10. So sánh YOLOv8 với các mẫu YOLO mới nhất

## 2.4 Các linh kiện sử dụng cho phần cứng

### 2.4.1 Raspberry Pi 4 [7]

Raspberry Pi là một dòng máy tính nhỏ gọn, giá cả phải chăng và tiêu thụ ít năng lượng. Mỗi chiếc Raspberry Pi có thể hoạt động như một máy tính đầy đủ, có thể kết nối với màn hình, bàn phím, chuột và các thiết bị khác qua cổng USB và HDMI. Được xây dựng trên nền tảng ARM và chạy hệ điều hành Linux, Raspberry Pi có nhiều phiên bản khác nhau.

Phiên bản Raspberry Pi 4 có nhiều cải tiến về hiệu suất và tính năng so với các phiên bản trước đó. Với kiến trúc lõi tứ và các cổng kết nối đa dạng, Raspberry Pi 4 đã trở thành một công cụ đa năng có thể được sử dụng trong nhiều lĩnh vực như xây dựng máy tính nhúng, IoT, phát triển web, robotics, hệ thống giám sát và quản lý, và nhiều ứng dụng khác.



Hình 2. 11. Hình ảnh mô phỏng và thực tế của Raspberry Pi 4

Bảng 2. 1. Thông số kỹ thuật của Raspberry Pi 4

<b>Phần cứng</b>	Bộ xử lý	ARM-Cortex A72 lõi tứ 64-bit
	Tốc độ xử lý	1.5 GHz

	LPDDR4 RAM	4GB
	Giải mã video	Giải mã phần cứng H.265 (HEVC) (lên tới 4Kp60)  Giải mã phần cứng H.264 (lên tới 1080p60)
	Đồ họa	3D VideoCore VI
	Màn hình hiển thị	Hỗ trợ màn hình HDMI kép lên tới 4Kp60
Giao diện	Wireless LAN	802.11 b/g/n/ac
	Bluetooth	Bluetooth 5.0 với BLE
	Lưu trữ	1 thẻ SD Card
	Hiển thị	2 Cổng micro-HDMI hỗ trợ màn hình kép có độ phân giải lên tới 4Kp60
	USB Ports	2 USB2 ports  2 USB3 ports
	Mạng	1 Cổng Gigabit Ethernet (hỗ trợ PoE với PoE HAT bổ sung)
	Camera Port	1 Cổng camera Raspberry Pi (MIPI CSI 2 làn)
	Display Port	1 Cổng hiển thị Raspberry Pi (MIPI DSI 2 làn)
	GPIO	28 User GPIO hỗ trợ nhiều tùy chọn giao diện:  - Lên đến 6 UART - Lên đến 6 I2C - Lên đến 5 SPI - 1 giao diện SDIO - 1 DPI (Màn hình RGB song song)

		<ul style="list-style-type: none"> <li>- 1 PCM</li> <li>- Lên đến 2 kênh PWM</li> <li>- Lên đến 3 đầu ra GPCLK</li> </ul>
<b>Phần mềm</b>	Tình trạng phát triển	<p>Tích cực phát triển và duy trì:</p> <ul style="list-style-type: none"> <li>- Hỗ trợ Linux kernel</li> <li>- Nhiều driver ngược dòng</li> <li>- Userland ổn định và được hỗ trợ tốt</li> <li>- Tính khả dụng của chức năng GPU thông qua các API tiêu chuẩn</li> </ul>

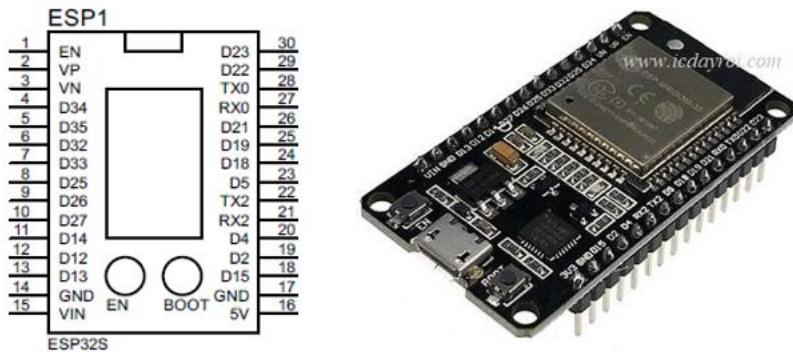
Bảng 2. 2. Thông số các chân của Raspberry Pi 4 được sử dụng

Chân	Tên	Mức điện áp hoạt động	Chức năng
<b>8</b>	GPIO14	THẤP	TXD0
<b>10</b>	GPIO15	THẤP	RXD0
<b>29</b>	GPIO5	CAO	GPCLK1
<b>31</b>	GPIO6	CAO	GPCLK2

#### 2.4.2 ESP32 [10]

ESP32 là một module phát triển bởi Espressif Systems, được coi là phiên bản cải tiến của ESP8266 SoC. Nó mang lại nhiều tính năng mới như hỗ trợ Wi-Fi 802.11 b/g/n, tích hợp Bluetooth 4.2 và BLE, và sử dụng bộ xử lý lõi kép Tensilica Xtensa LX6 với tốc độ lên đến 240 MHz, cải thiện khả năng xử lý đa nhiệm. Đồng thời, số lượng chân GPIO được mở rộng, tạo điều kiện thuận lợi cho kết nối và tương tác với nhiều thiết bị hơn. ESP32 được hỗ trợ bởi các phần mềm lập trình phổ biến như Arduino IDE, MicroPython. Với các tính năng tiên tiến này, ESP32 đã trở thành lựa chọn hàng đầu trong các dự án IoT.

Nhờ vào các tính năng vượt trội, module ESP32 đã được ưa chuộng và trở thành một trong những lựa chọn hàng đầu cho các dự án IoT.



Hình 2. 12. Hình ảnh mô phỏng và thực tế của ESP32

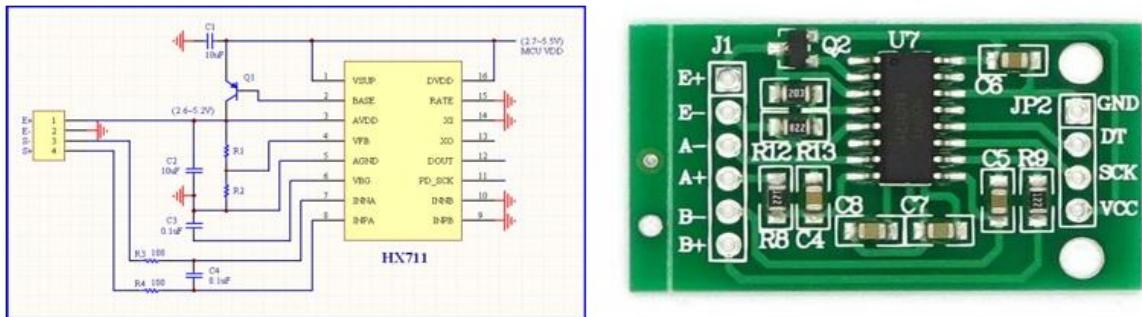
Bảng 2. 3. Thông số kỹ thuật của ESP32

	<b>Thông số</b>
<b>Nguồn cấp</b>	3.0V ~ 3.6V
<b>Chuẩn giao tiếp</b>	UART
<b>Kích thước</b>	28.33x51.45
<b>Số lượng cổng IO</b>	30

### 2.4.3 Các cảm biến và linh kiện sử dụng

#### a. Bộ chuyển đổi tín hiệu tương tự sang tín hiệu số Hx711 [9]

Module HX711 là một bộ chuyển đổi tín hiệu tương tự từ Loadcell sang tín hiệu kỹ thuật số, giúp việc đọc dữ liệu thông qua các chân số của vi điều khiển trở nên dễ dàng hơn. Khi Loadcell chịu lực, nó tạo ra một tín hiệu analog biến đổi. HX711 sau đó chuyển đổi tín hiệu này thành dạng số. Đặc biệt, HX711 cũng có khả năng lọc tín hiệu để giảm nhiễu và ổn định dữ liệu đó, làm cho kết quả từ Loadcell trở nên chính xác hơn. Module này được thiết kế để dễ dàng kết nối với các vi điều khiển thông qua các giao diện như SPI hoặc I2C, giúp tích hợp vào hệ thống điều khiển và xử lý dữ liệu một cách thuận tiện.



Hình 2. 13. Hình ảnh mô phỏng và thực tế của module ADC HX711

Bảng 2. 4. Thông số các chân của module ADC HX711

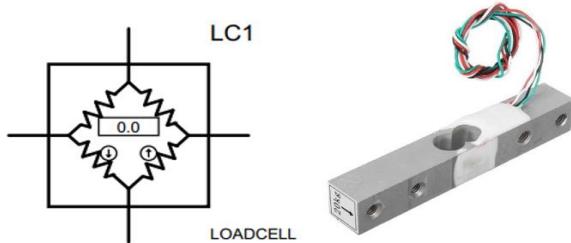
Chân	Tên	Chức năng	Mô tả
1	VSUP	Nguồn	2.7V ~ 5.5V
2	BASE	Đầu ra tín hiệu tương tự	NC khi không sử dụng
3	AVDD	Nguồn	Nguồn tín hiệu tương tự: 2.6V ~ 5.5V
4	VFB	Đầu vào tín hiệu tương tự	Nối với AGND khi không sử dụng
5	AGND	Nối đất	
6	VBG	Đầu ra tín hiệu tương tự	Đầu ra bô qua tham chiếu
7	INA-	Đầu vào tín hiệu tương tự	Kênh A đầu vào âm
8	INA+	Đầu vào tín hiệu tương tự	Kênh A đầu vào dương
9	INB-	Đầu vào tín hiệu tương tự	Kênh B đầu vào âm
10	INB+	Đầu vào tín hiệu tương tự	Kênh B đầu vào dương
11	PD_SCK	Đầu vào tín hiệu số	Điều khiển tắt nguồn (CAO) và tín hiệu Clock
12	DOUT	Đầu ra tín hiệu số	Đầu ra dữ liệu nối tiếp

13	XO	Đầu vào/đầu ra kỹ thuật số	Crystal I/O (NC khi không sử dụng)
14	XI	Đầu vào tín hiệu số	Crystal I/O hoặc tín hiệu Clock bên ngoài 0: sử dụng bộ tạo dao động trên chip
15	RATE	Đầu vào tín hiệu số	Kiểm soát tốc độ dữ liệu đầu ra 0: 10Hz ; 1: 80Hz
16	DVDD	Nguồn	Nguồn tín hiệu số: 2.6V ~ 5.5V

### b. Cảm biến khối lượng 5Kg [14]

Loadcell là một loại cảm biến được sử dụng để đo lực hoặc khối lượng, thường áp dụng trong các hệ thống cân và máy đo lực. Các loadcell có nhiều hình dạng khác nhau như đĩa, trụ, cầu, thanh, chữ S, tùy thuộc vào mục đích sử dụng.

Các loadcell sử dụng cảm biến điện trở gọi là strain gauge, thường được kết hợp trong một mạng Wheatstone Bridge. Khi có một lực tác động lên loadcell, sự biến đổi của điện trở trong mạng Wheatstone Bridge tạo ra một hiệu điện thế không đối xứng. Hiệu điện thế này thường được đo bằng một bộ chuyển đổi A/D để chuyển đổi thành giá trị số có thể đọc được.



Hình 2. 14. Hình ảnh mô phỏng và thực tế của loadcell 5kg

Bảng 2. 5. Thông số kỹ thuật của Loadcell 5kg

STT	Thông số	Giá trị
1	Tải trọng	5Kg
2	Điện áp hoạt động	5V
3	Nhiệt độ hoạt động	- 20°C ~ 65°C
4	Chất liệu cảm biến	Nhôm
5	Độ dài dây	180mm
6	Dây đỏ	Ngõ vào (+)
7	Dây đen	Ngõ vào (-)
8	Dây xanh lá	Ngõ ra (+)
9	Dây trắng	Ngõ ra (-)

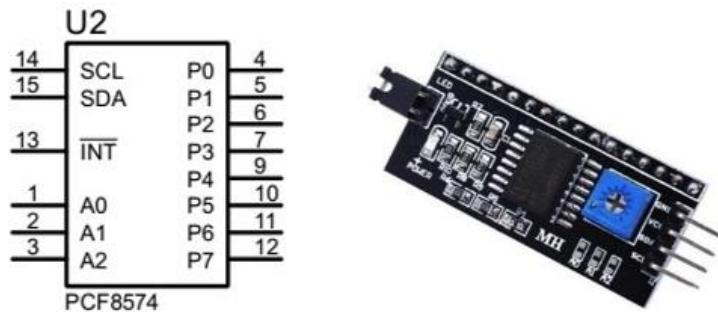
### c. Màn hình LCD – I2C 16x2 [11]

Màn hình LCD 16x2 thường được áp dụng để hiển thị dữ liệu trong các dự án nhúng và điện tử. Thiết bị này bao gồm hai dòng, mỗi dòng có khả năng hiển thị lên đến 16 ký tự.

Module I2C thường được tích hợp trực tiếp vào màn hình LCD nhằm giảm đáng kể số lượng chân kết nối cần thiết so với cách kết nối truyền thông qua các chân kỹ thuật số. Với module I2C, chỉ cần sử dụng hai chân SDA và SCL để kết nối với vi điều khiển, tạo điều kiện thuận lợi hơn cho việc tích hợp và sử dụng.



Hình 2. 15. Hình ảnh mô phỏng và thực tế của màn hình LCD



Hình 2. 16. Hình ảnh mô phỏng và thực tế của module I2C

Bảng 2. 6. Thông số kỹ thuật của màn hình LCD

STT	Ký hiệu	Thông số	Min (nhỏ nhất)	Nom (chuẩn)	Max (lớn nhất)	Đơn vị
1	VIH	Điện áp vào mức cao	2.2		VEE	V
2	VIL	Điện áp vào mức thấp			0.6	V
3	VOH	Điện áp ra mức cao	2.2			V
4	VOL	Điện áp ra mức thấp			0.4	V
5	IDD	Dòng điện ngõ vào		2.5	4.0	mA
6	VDD	Điện áp ngõ vào	4.7	5.0	5.3	V
			2.7	3.0	5.3	

Bảng 2. 7. Thông số các chân của màn hình LCD

Chân	Tên	Chức năng
1	V <sub>SS</sub>	Ground
2	V <sub>DD</sub>	Nguồn cung cấp đầu vào +3V hoặc +5V

<b>3</b>	V <sub>0</sub>	Điều chỉnh độ tương phản
<b>4</b>	RS	Tín hiệu chọn thanh ghi H/L
<b>5</b>	R/ $\bar{R}$	Tín hiệu đọc / ghi H/L
<b>6</b>	E	H → L cho phép tín hiệu
<b>7–14</b>	DB0–DB7	Đường truyền dữ liệu H/L
<b>15</b>	A	Nguồn cung cấp cho đèn LED (4.2V)
<b>16</b>	K	Nguồn cung cấp cho B/L (0V)
<b>17</b>	NC/V <sub>EE</sub>	NC hoặc đầu ra điện áp âm
<b>18</b>	NC	Kết nối NC

#### d. Camera Pi Module [8]

Camera Pi là một loại camera chuyên dụng cho việc xử lý trên Raspberry Pi, thường được ứng dụng trong các ứng dụng IoT. Camera này kết nối với Raspberry Pi thông qua giao diện CSI (Camera Serial Interface). Với khả năng chụp ảnh và quay video, người dùng có thể tinh chỉnh các thiết lập như độ phân giải, khẩu độ, cân bằng trắng và các tùy chọn khác thông qua phần mềm điều khiển. Camera Pi được thiết kế để tương thích với nhiều phiên bản Raspberry Pi.



Hình 2. 17. Hình ảnh mô phỏng và thực tế của camera Pi module

Bảng 2. 8. Thông số kỹ thuật của Camera Pi

	Thông số	Đơn vị
<b>Phiên bản</b>	Raspberry Pi Camera Module v2	
<b>Độ phân giải</b>	8	MP
<b>Độ phân giải ảnh tĩnh</b>	3280 x 2464	P
<b>Nhiệt độ hoạt động</b>	-20 – 60	°C

Chức năng điều khiển hình ảnh:

- Kiểm soát phơi sáng tự động
- Cân bằng trắng tự động
- Bộ lọc băng tần tự động
- Tự động phát hiện độ chói 50/60 Hz
- Tự động hiệu chỉnh mức độ đèn

### e. Cảm biến khoảng cách HC-SR04 [15]

Cảm biến HC-SR04 là loại cảm biến siêu âm dùng để đo khoảng cách. Nó hoạt động bằng cách phát ra sóng siêu âm và đo thời gian sóng dội lại sau khi chạm vào vật cản, từ đó tính toán khoảng cách với độ chính xác cao. Trong đề tài này nó được sử dụng để chuyển đổi giá trị hiển thị của 2 bàn cân.



Hình 2. 18. Cảm biến khoảng cách HC-SR04

Bảng 2. 9. Thông số kỹ thuật cảm biến khoảng cách HC-SR04

STT	Thông số	Chức năng, giá trị
1	Điện áp hoạt động	5VDC
2	Dòng tiêu thụ	10~40mA
3	Tần số phát sóng	40Khz
4	Góc quét	<15 độ
5	Khoảng cách đo được	2-450cm
6	Sai số	0.3cm

Bảng 2. 10. Chức năng các chân cảm biến HC-SR04

Tên	Chức năng
VCC	Điện áp hoạt động
GND	Dòng tiêu thụ
ECHO	Tần số phát sóng
TRIGGER	Góc quét

#### f. Buzzer [17]

Buzzer được sử dụng để phát ra tín hiệu âm thanh để báo hiệu khi dữ liệu được gửi trong đồ án.



Hình 2. 19. Buzzer 5V

Bảng 2. 11. Thông số kỹ thuật của buzzer 5V

STT	Thông số	Chức năng
1	Điện áp hoạt động	3.5V - 5.5V
2	Dòng điện tiêu thụ	<25mA
3	Tần số cộng hưởng	2300Hz ± 500Hz
4	Biên độ âm thanh	>80 dB

#### g. Led siêu sáng 10mm [16]

Dựa trên yêu cầu của hệ thống, nhóm chúng tôi lựa chọn led siêu sáng để đáp ứng được nhu cầu.



Hình 2. 20. Led siêu sáng 10mm

Bảng 2. 12. Thông số kỹ thuật của led siêu sáng 10mm

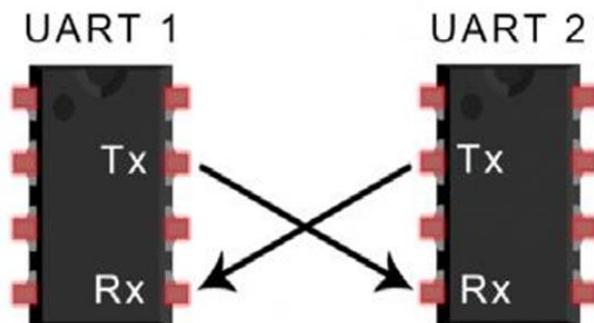
STT	Thông số	Chức năng
1	Điện áp hoạt động	3.2-3.4V
2	Dòng điện tiêu thụ	10mA
3	Độ sáng	20000-25000mcd
4	Phạm vi nhiệt độ màu	5000 K

## 2.5 Các chuẩn giao tiếp

### 2.5.1 Chuẩn giao tiếp UART [24]

UART là một chuẩn giao diện truyền thông nối tiếp, cho phép truyền và nhận dữ liệu giữa master và slave. Đây là giao diện không đồng bộ, không có đường xung nhịp để đồng bộ hóa tốc độ truyền dữ liệu.

Trong UART, cần đảm bảo rằng cả hai thiết bị truyền và nhận hoạt động ở cùng một tốc độ, được biểu diễn bằng bit trên giây (bps). Điều quan trọng là tốc độ truyền của cả hai bên (truyền và nhận) không được chênh lệch quá 10% để đảm bảo giao tiếp được thực hiện một cách chính xác.



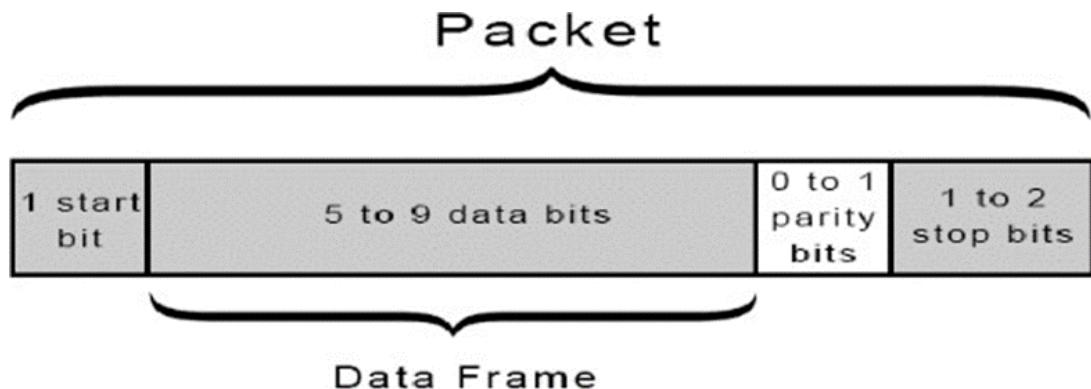
Hình 2. 21. Giao tiếp UART

UART có 3 chế độ truyền nối tiếp:

- **Full duplex:** Truyền dữ liệu đồng thời đến và đi từ cả master và slave
- **Half duplex:** Dữ liệu được truyền theo một hướng tại một thời điểm
- **Simplex:** Chỉ truyền dữ liệu một chiều

## Quá trình truyền dữ liệu của UART

Dữ liệu truyền được chuyển đổi thành các gói, mỗi gói chứa 1 bit bắt đầu, từ 5 đến 9 bit dữ liệu, 1 bit chẵn lẻ tùy chọn và từ 1 đến 2 bit dừng.



Hình 2. 22. Gói dữ liệu của chuẩn giao tiếp UART

Để khởi đầu việc truyền dữ liệu, UART truyền thường chuyển đổi đường truyền từ mức điện áp cao xuống mức điện áp thấp. Khi có sự thay đổi này trên đường truyền, UART bên nhận sẽ nhận biết và bắt đầu quá trình nhận dữ liệu.

Sau khi nhận được tất cả các bit dữ liệu, UART bên nhận thường đọc các parity bits để kiểm tra xem quá trình truyền có bị lỗi hay không. Cuối cùng, UART truyền sẽ đưa đường truyền dữ liệu từ mức điện áp thấp lên cao ít nhất 2 bit để kết thúc quá trình truyền.

Mặc dù UART là một giao thức cổ điển và chỉ hỗ trợ giao tiếp giữa một bên truyền và một bên nhận, nhưng nó vẫn được sử dụng phổ biến nhờ vào các ưu điểm của nó, như:

- Sử dụng số lượng dây ít (2 dây)
- Clock có thể không cần
- Có thể kiểm tra lỗi dựa trên bit chẵn lẻ
- Tài liệu sử dụng đa dạng

## 2.5.2 Chuẩn giao tiếp I2C [24]

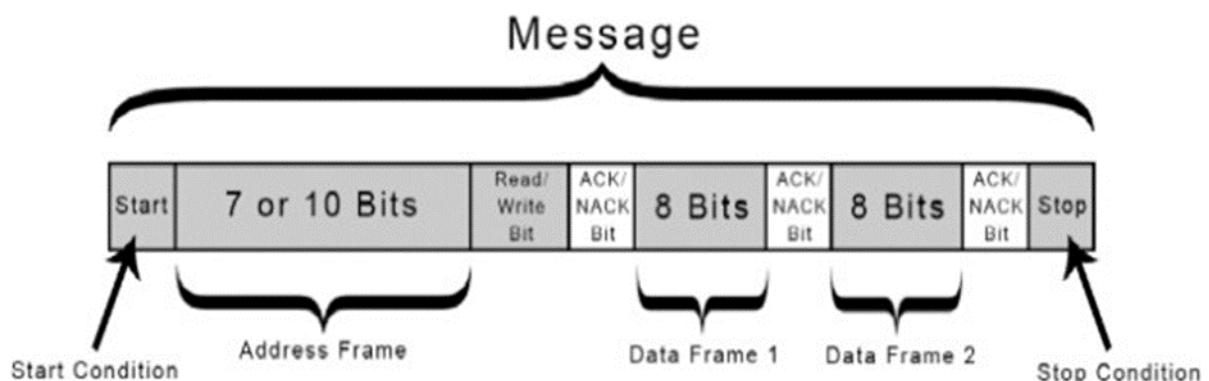
I2C là một giao thức truyền thông nối tiếp, thường áp dụng hiệu quả cho các cảm biến và module. Tuy nhiên I2C chưa hẳn luôn tốt đối với sự giao tiếp các linh kiện mà ta sử dụng trên bo mạch (PCB). Nó cho chuyển đổi 2 chiều truyền thông giữa các thiết bị kết nối chỉ bằng hai dây.

Với I2C, việc kết nối tối đa tới 128 thiết bị trên một bo mạch mà luôn có thể truyền tốt và rõ ràng giữa chúng. I2C thích hợp với việc dự án có nhiều thành phần hoạt động cùng nhau.

Tốc độ của I2C bị chi phối bởi tốc độ truyền dữ liệu, dây truyền thông tốt hay không, và mức độ nhiễu ngoại vi. Nó cũng có thể dùng cho việc kết nối thiết bị chậm như bộ chuyển đổi ADC, vi điều khiển, giao diện I/O, và nhiều thiết bị khác.

### I2C hoạt động theo nguyên lý

Dữ liệu được truyền đi dưới dạng các thông điệp, các thông điệp này được chia thành các khung bao gồm: một khung chứa địa chỉ nhị phân của thiết bị tớ, một hoặc nhiều khung dữ liệu, các bit đọc/ghi, các bit xác nhận ACK/NACK, cùng với điều kiện bắt đầu và điều kiện kết thúc.



Hình 2. 23. Gói dữ liệu của chuẩn giao tiếp I2C

I2C có hai dây: SCL và SDA, tương ứng là dây xung Clock và dây dữ liệu nối tiếp. Dây xung Clock (SCL) được sử dụng để truyền dữ liệu được đồng bộ hóa, và dữ liệu được gửi và nhận thông qua dây SDA.

Master tạo xung Clock và gửi địa chỉ của slave cần nhận hoặc truyền dữ liệu đến tất cả các slave được kết nối. Các thiết bị slave sẽ so sánh địa chỉ nhận được từ thiết bị master với địa chỉ của chính chúng, nếu khác nhau slave sẽ rời khỏi đường tín hiệu mức cao. Nếu giống nhau, bit ACK sẽ được trả về và bắt đầu truyền / nhận dữ liệu.

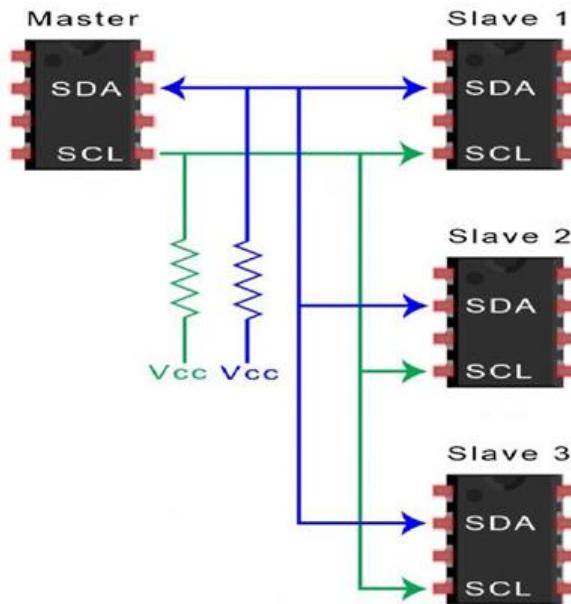
Để quá trình truyền dữ liệu chấm dứt, master gửi tín hiệu dừng đến slave.

**Ưu điểm :**

- Hỗ trợ nhiều thiết bị master khác nhau
- Cung cấp giao tiếp nhiều master nhiều slave
- Là một giao thức linh hoạt, phỏ biến

**Nhược điểm :**

- Tương đối chậm do cần sử dụng điện trở kéo lên
- Chiếm nhiều diện tích
- Số lượng thiết bị kết nối càng tăng khiến kiến trúc càng phức tạp



Hình 2. 24. Giao tiếp I2C giữa một Master và nhiều Slave

## **2.6 Giới thiệu về IoTs và các ứng dụng**

IoT đã trở thành một khái niệm quan trọng đối với sự tiến bộ liên tục trong thế giới công nghệ hiện nay. Hệ thống IoT bao gồm các thành phần như cảm biến, vi xử lý, vi điều khiển, và chúng giao tiếp với nhau qua các mạng kết nối như Internet, Wi-Fi, Bluetooth, Zigbee, LoRa. Dữ liệu từ các thiết bị này được chuyển đến trung tâm xử lý dữ liệu, từ đó người dùng có thể theo dõi và điều khiển các thiết bị qua các ứng dụng trên điện thoại hoặc máy tính.

Với sự phát triển mạnh mẽ của công nghệ, ngày nay chúng ta chứng kiến sự xuất hiện ngày càng nhiều thiết bị IoT được phát triển để đáp ứng các nhu cầu đa dạng của con người. Một số lĩnh vực phổ biến mà IoT được ứng dụng gồm:

- Nhà thông minh: Điều khiển các thiết bị như đèn, quạt, TV, camera an ninh,... từ xa thông qua Internet qua các thiết bị di động.
- Y tế: Sử dụng đồng hồ thông minh và các thiết bị theo dõi sức khỏe để gửi dữ liệu trực tuyến và theo dõi sức khỏe cá nhân.
- Công nghiệp: Thu thập dữ liệu từ các thiết bị và máy móc để tối ưu hóa quy trình sản xuất và cải thiện hiệu suất công việc.
- Nông nghiệp: Quản lý tưới tiêu, giám sát độ ẩm đất và điều kiện thời tiết để cải thiện năng suất nông nghiệp.
- Giao thông: Giám sát và quản lý giao thông đô thị, cung cấp thông tin về lưu lượng giao thông và đồ xe.

## **2.7 Nền tảng về Django Rest Framework**

### **2.7.1 Tổng quan về Django Rest Framework [3]**

Django Rest Framework là một framework phổ biến được dùng để phát triển các ứng dụng web, đặc biệt là các API RESTful trong ứng dụng Django. Nó mang lại nhiều lợi ích quan trọng như sau:

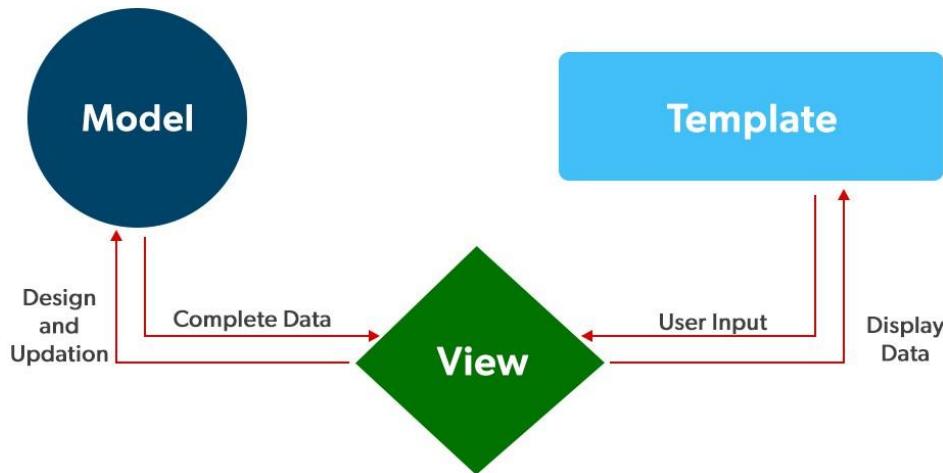
- Đa dạng các loại views như APIView, GenericAPIView, ModelViewSet, giúp linh hoạt trong việc xử lý yêu cầu và trả về phản hồi.
- Đơn giản hóa thao tác CRUD (Create, Retrieve, Update, Delete) thông qua các ViewSets, giảm sự lặp lại mã nguồn.

- Hỗ trợ nhiều định dạng dữ liệu như JSON, XML, YAML, HTML, thuận tiện cho việc tương tác với các loại dữ liệu khác nhau.
- Cung cấp nhiều phương pháp xác thực người dùng và kiểm soát quyền truy cập vào các API, giúp bảo mật thông tin và quản lý người dùng một cách linh hoạt.

### 2.7.2 Ứng dụng Django vào việc xử lý khi người dùng yêu cầu truy cập bên trong ứng dụng [3]

Django Rest Framework (DRF) làm cho việc phát triển API RESTful trở nên dễ dàng hơn bằng cách sử dụng các thành phần chính của Django. Quy trình xử lý yêu cầu trong ứng dụng thường bao gồm các bước sau:

- Định tuyến (Routing): Tương tự như khi xây dựng ứng dụng web, chúng ta cần định rõ các đường dẫn cho các endpoint API trong file 'urls.py'. Các endpoint này được liên kết với các view function, giống như khi xử lý yêu cầu từ trình duyệt.
- Mô hình (Model): Đây là bước tạo bảng hoặc trường cho Django, tạo cơ sở dữ liệu cho ứng dụng. Lớp này chứa mọi thứ về dữ liệu và có trách nhiệm duy trì mọi dữ liệu cho ứng dụng khi đang hoạt động. Nó có thể thêm, xóa, chỉnh sửa bất kỳ nội dung nào mà người dùng mong muốn.
- Mẫu (Template): Mẫu là đầu ra của HTML. Thông thường, chúng được xây dựng bằng HTML, CSS, và JavaScript trong các tệp có đuôi là .html. Mục đích của lớp này là trình bày và hiển thị các nội dung trên trang web để người dùng có thể xem và tương tác. Để cung cấp một giao diện người dùng, chúng ta cần một mẫu để hiển thị nội dung một cách thích hợp và phản ánh nhu cầu của người dùng.
- Xử lý View: Tạo các view function để xử lý logic khi người dùng gửi yêu cầu. Views được xem là cầu nối của mô hình và mẫu.



Hình 2. 25. Mô hình MVT (Model – View – Template)

Trong hình 2.11, sau khi người dùng đã thành công trong việc tạo cơ sở dữ liệu và thiết kế mẫu theo ý muốn, thông tin sẽ được gửi tới bộ xử lý view. Bộ xử lý view nhận yêu cầu từ hai hàm: một là mô hình và hai là mẫu. Tiếp theo, nó sẽ thực hiện cập nhật thông tin người dùng thông qua mô hình và hiển thị các thông tin đó lên màn hình dựa trên mẫu đã được thiết kế trước đó. Quá trình này giúp người dùng tương tác với dữ liệu một cách trực quan và thuận tiện.

### 2.7.3 Giao tiếp giữa máy chủ với cơ sở dữ liệu [4]

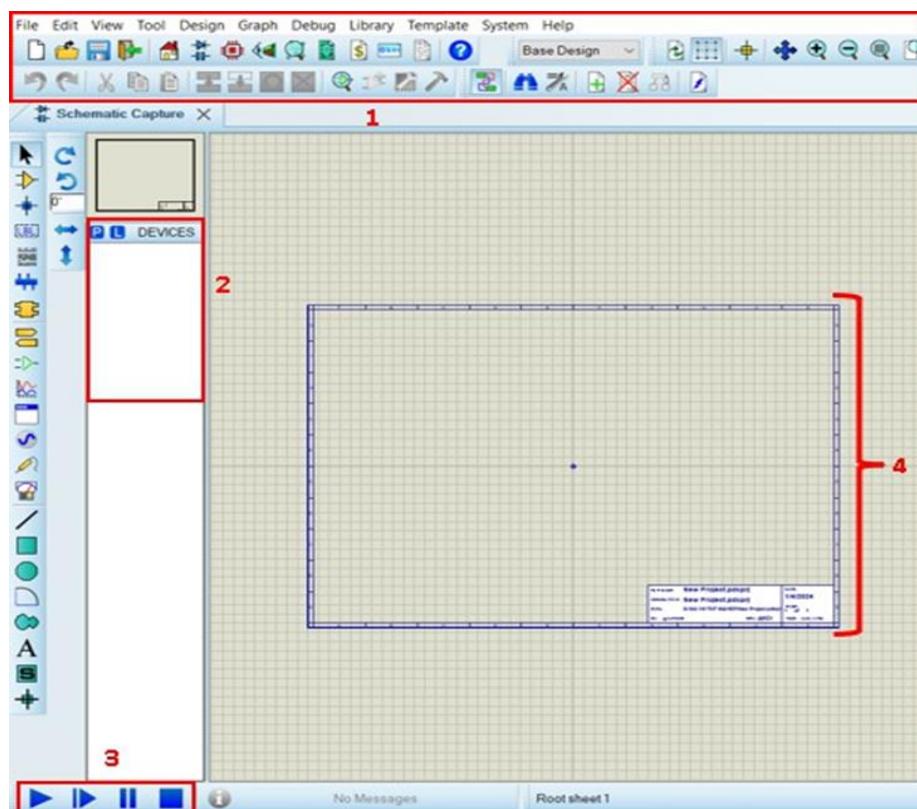
Một trong những phần quan trọng nhất của việc phát triển API là khả năng tương tác với cơ sở dữ liệu. Django Rest Framework tích hợp chặt chẽ với ORM (Object-Relational Mapping) của Django để thực hiện các hoạt động trên cơ sở dữ liệu.

Models đại diện cho cấu trúc dữ liệu của ứng dụng và được sử dụng để mô hình hóa dữ liệu. Serializer, một thành phần quan trọng của DRF, giúp chuyển đổi giữa các đối tượng Python và các định dạng dữ liệu có thể truyền qua mạng. Việc này không chỉ giúp quản lý dữ liệu một cách hiệu quả mà còn đảm bảo tính nhất quán giữa dữ liệu trên máy chủ và cơ sở dữ liệu. Serializer cung cấp một cầu nối linh hoạt giữa thế giới của các đối tượng Python và các định dạng dữ liệu di động như JSON, từ đó giúp xây dựng và duy trì các API RESTful một cách hiệu quả và dễ dàng bảo trì.

## 2.8 Tổng quan về các công cụ hỗ trợ xây dựng vào lập trình cho hệ thống

### 2.8.1 Proteus

Proteus là một phần mềm mô phỏng hệ thống mạch điện tử và thiết kế PCB, được sử dụng rộng rãi trong lĩnh vực thiết kế và phát triển các mạch điện tử. Được phát triển bởi Labcenter Electronics Ltd., Proteus giúp kỹ sư và nhà thiết kế kiểm tra, mô phỏng và thử nghiệm các mạch điện tử trước khi chúng được thi công.



Hình 2. 26. Giao diện màn hình chính của phần mềm Proteus

Các thành phần chính của giao diện Proteus:

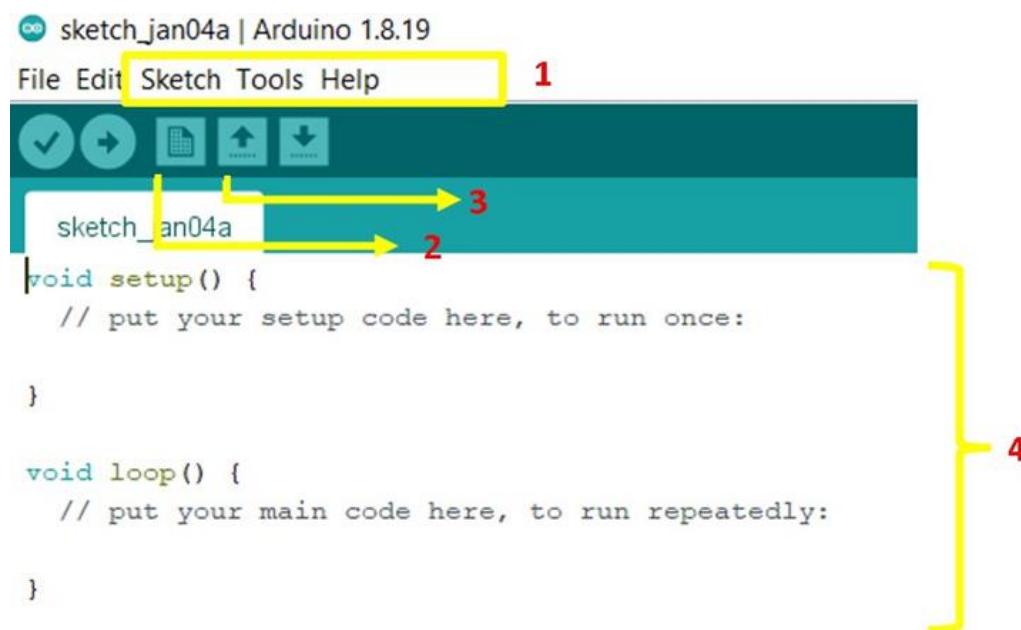
1. Thanh công cụ: cung cấp các tùy chọn phục vụ cho việc vẽ, chỉnh sửa, chú thích, tính toán thông số... cho các linh kiện
2. Bảng chứa danh sách linh kiện: hiển thị tên các linh kiện đã được sử dụng, giúp người dùng nắm được tên, số lượng các linh kiện có trong mạch và dễ dàng lấy ra sử dụng lại mà không mất nhiều thời gian tìm kiếm

3. Các nút chức năng: thực hiện các tác vụ như chạy chương trình, tạm dừng, ngắt, bỏ qua.
4. Không gian thiết kế: nơi người dùng đặt các linh kiện, nối dây để tạo thành một hệ thống hoàn chỉnh

### 2.8.2 Arduino IDE

Arduino IDE là một phần mềm được dùng để lập trình và biên dịch mã cho các bo mạch Arduino khác nhau như Arduino Uno, Arduino Mega, Arduino Micro,... Mỗi bo mạch này đi kèm với một vi điều khiển được lập trình sẵn, chấp nhận thông tin dưới dạng mã được tạo ra từ môi trường lập trình tích hợp (IDE). Mã nguồn, hay còn gọi là "sketch," được phát triển trên giao diện người dùng của IDE, sau đó được biên dịch thành một file Hex trước khi được tải lên bo mạch thông qua cổng nối USB hoặc các phương tiện truyền thông khác.

Môi trường IDE bao gồm hai thành phần chính: Trình chỉnh sửa và Trình biên dịch. Phần soạn thảo mã được sử dụng để viết mã nguồn theo yêu cầu, phần biên dịch chịu trách nhiệm biên dịch mã nguồn đó thành ngôn ngữ máy có thể thực thi trực tiếp trên bo mạch Arduino. Môi trường này hỗ trợ cả ngôn ngữ lập trình C và C++.



Hình 2. 27. Giao diện chính của phần mềm Arduino IDE

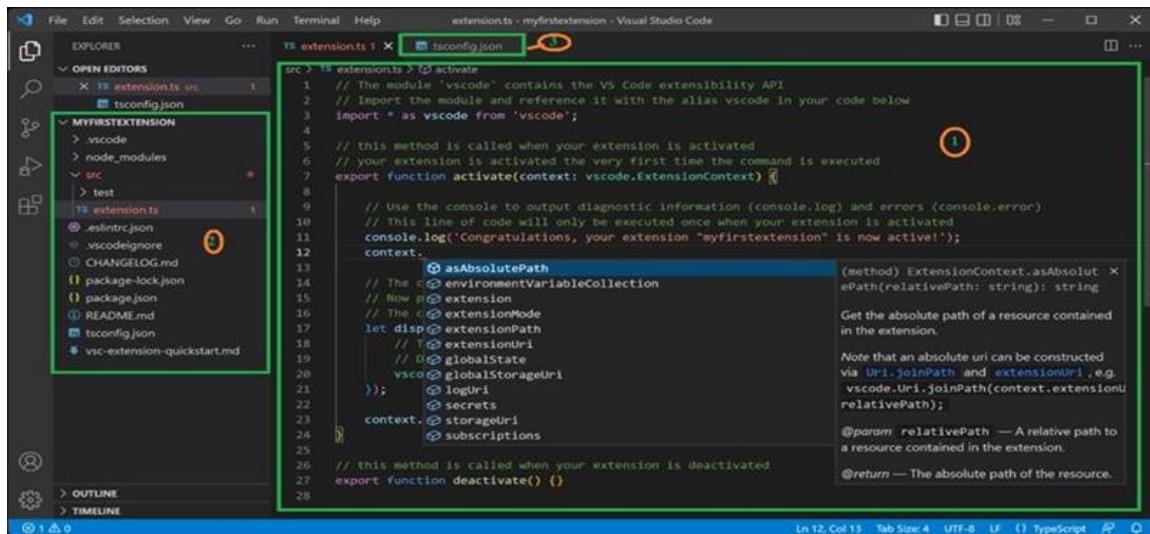
Khi làm việc trên Arduino IDE lưu ý một số phần:

1. Menu lệnh: Cung cấp các tùy chọn để tạo, mở, thay tên,... một dự án.
2. Nút kiểm tra lỗi (Verify): Dùng để xem xét việc có lỗi không đối với chương trình. Phát hiện lỗi, sẽ có thông báo về việc sai ở hàng nào và sai về nội dung gì.
3. Nút đưa chương trình vào Arduino (Upload): Dùng để tải chương trình đã viết vào Arduino. Trước khi nạp, lỗi sẽ được kiểm tra, và chỉ khi không có lỗi thì chương trình mới được tải xuống Arduino.
4. Vùng lập trình: Dùng để viết và chỉnh sửa các lệnh trong chương trình.

### 2.8.3 Visual Studio Code

Visual Studio Code là một ứng dụng hỗ trợ việc biên tập và soạn thảo mã nguồn, hỗ trợ quá trình xây dựng và thiết kế website một cách hiệu quả. Được phát triển và duy trì bởi Microsoft, VS Code có sẵn trên ba hệ điều hành phổ biến là Windows, macOS và Linux.

Ứng dụng này đặc biệt được tối ưu hóa để hỗ trợ các ngôn ngữ lập trình chủ yếu trong lĩnh vực phát triển web, bao gồm JavaScript, TypeScript và Node.js.



Hình 2. 28. Giao diện chính của phần mềm Visual Studio Code

- (1) Giao diện làm việc chính, nơi chứa code của chương trình.
- (2) Khu vực chứa các dự án đã hoặc đang thực hiện, ta có thể phân biệt tên

ngôn ngữ được dùng trong tệp tin đó thông qua ký tự viết tắt như: “.js”,  
“.html”,...

(3) Khu vực chứa các tab đang làm việc, ta có thể chuyển đổi các tab đang  
thực hiện qua lại tại đây

## CHƯƠNG 3: YOLOV8, THƯ VIỆN OPENCV VÀ PYZBAR

### 3.1. Giới thiệu về YOLOv8

#### 3.1.1. Lịch sử phát triển [5]

YOLOv8 là phiên bản mới nhất của thuật toán nhận dạng đối tượng YOLO (You Only Look Once), được phát triển bởi Ultralytics vào năm 2022. YOLOv8 là một thuật toán nhận dạng đối tượng nhanh chóng, chính xác và hiệu quả, được sử dụng trong nhiều ứng dụng khác nhau, bao gồm:

- Nhận dạng đối tượng trong thời gian thực
- Giám sát video
- Xe tự lái

YOLOv8 là sự kế thừa của các phiên bản YOLO trước đó, bao gồm YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6 và YOLOv7. Mỗi phiên bản YOLO đều có những cải tiến về tốc độ, độ chính xác và hiệu quả. YOLOv8 là phiên bản YOLO mới và tốt nhất, với nhiều cải tiến so với các phiên bản trước đó.

#### 3.1.2. Ưu điểm và nhược điểm

Ưu điểm:

- Nhanh chóng: YOLOv8 là một trong những thuật toán nhận dạng đối tượng nhanh nhất hiện nay.
- Chính xác: YOLOv8 là một trong những thuật toán nhận dạng đối tượng được xây dựng trên các tiền bộ về học sâu và thị giác máy tính, đảm bảo độ chính xác cao trong nhận diện đối tượng.
- Hiệu quả: YOLOv8 là một thuật toán nhận dạng đối tượng hiệu quả, có thể chạy trên các thiết bị có tài nguyên hạn chế.
- Dễ dàng triển khai: YOLOv8 là một thuật toán nhận dạng đối tượng dễ dàng triển khai, với nhiều tài liệu và mã nguồn mở có sẵn.

Nhược điểm:

- Ít chính xác hơn một số thuật toán nhận dạng đối tượng hai giai đoạn.
- Khó khăn trong việc phát hiện các đối tượng nhỏ.
- Yêu cầu các tài nguyên tính toán cao để đạt tốc độ xử lý nhanh và chính xác.

### **3.1.3. Kiến trúc mạng [5]**

Kiến trúc mạng của YOLOv8 dựa trên kiến trúc mạng của YOLOv7, với một số cải tiến. Kiến trúc mạng của YOLOv8 bao gồm:

- Phần trích xuất đặc trưng: Phần này sử dụng mạng nơ-ron tích chập (CNN) để trích xuất các đặc trưng từ hình ảnh.
- Phần dự đoán: Phần này sử dụng các lớp fully-connected để dự đoán các đối tượng và vị trí của chúng.

Kiến trúc mạng của YOLOv8 được thiết kế để tối ưu hóa tốc độ và độ chính xác. Phần trích xuất đặc trưng sử dụng mạng nơ-ron CNN hiệu quả để trích xuất các đặc trưng từ hình ảnh. Phần dự đoán sử dụng các lớp fully-connected để dự đoán các đối tượng và vị trí của chúng một cách chính xác.

### **3.1.4 Ứng dụng của YOLOv8**

YOLOv8 được sử dụng trong nhiều ứng dụng khác nhau, bao gồm:

- Nhận dạng đối tượng trong thời gian thực
- Giám sát video
- Xe tự lái
- Robot
- Y tế
- An ninh

## **3.2. Cấu trúc của YOLOv8**

YOLOv8 là một thuật toán nhận dạng đối tượng dựa trên mạng nơ-ron tích chập (CNN). Cấu trúc của YOLOv8 bao gồm ba phần chính: backbone, neck và head.

### **3.2.1. Backbone**

Backbone là phần trích xuất đặc trưng của YOLOv8. Backbone sử dụng mạng nơ-ron CNN để trích xuất các đặc trưng từ hình ảnh. YOLOv8 sử dụng mạng nơ-ron CNN EfficientDet-D7 làm backbone. EfficientDet-D7 là một mạng nơ-ron CNN hiệu quả, được thiết kế để tối ưu hóa tốc độ và độ chính xác.

### **3.2.2. Neck**

Neck là phần kết nối giữa backbone và head. Neck sử dụng các lớp convolutional để

kết hợp các đặc trưng từ các lớp khác nhau của backbone. YOLOv8 sử dụng mạng nơ-ron Path Aggregation Network (PAN) làm neck. PAN là một mạng nơ-ron hiệu quả, được thiết kế để kết hợp các đặc trưng từ các lớp khác nhau của backbone một cách hiệu quả.

### 3.2.3. Head

Head là phần dự đoán của YOLOv8. Head sử dụng các lớp fully-connected để dự đoán các đối tượng và vị trí của chúng. YOLOv8 sử dụng ba head để dự đoán các đối tượng ở ba kích thước khác nhau.

## 3.3. Quá trình huấn luyện YOLOv8 [12][19]

### 3.3.1. Chuẩn Bị Dữ Liệu Huấn Luyện

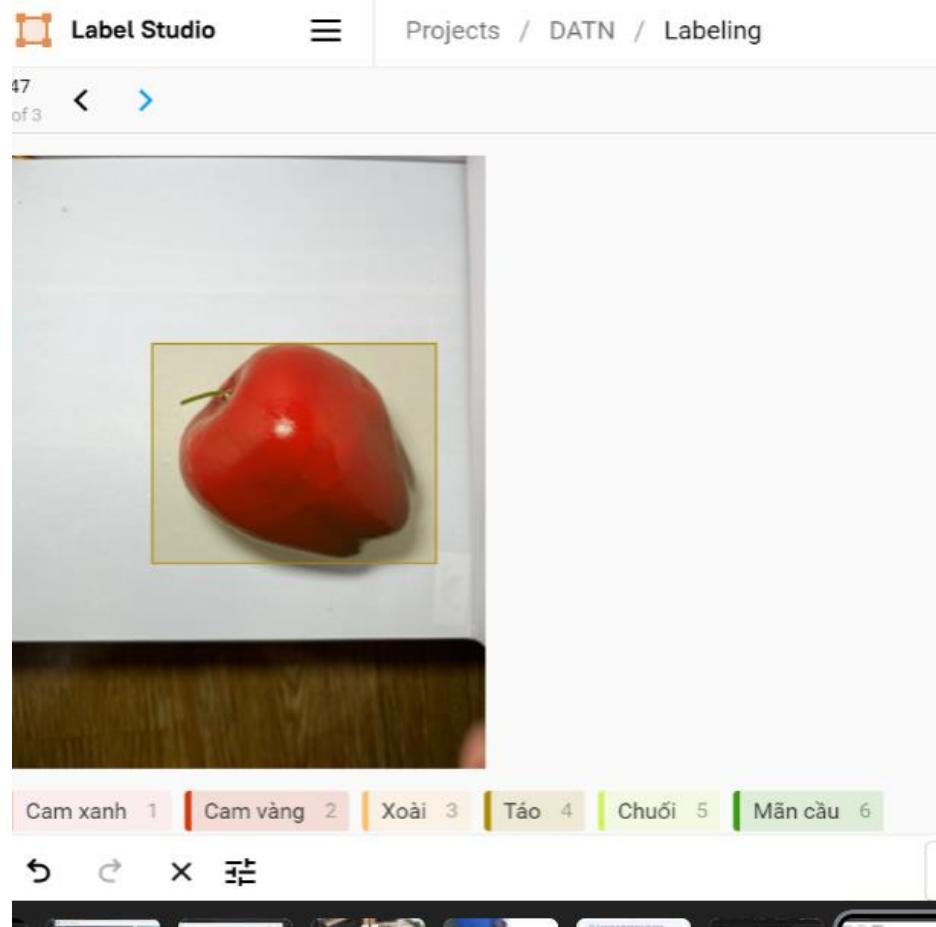
Thu thập dữ liệu là bước đầu tiên và quan trọng nhất trong quá trình huấn luyện YOLOv8. Dữ liệu huấn luyện nên đa dạng và phong phú, bao gồm nhiều đối tượng khác nhau trong nhiều bối cảnh khác nhau. Dữ liệu huấn luyện cũng nên được chú thích chính xác, với các bounding box bao quanh mỗi đối tượng.

### 3.3.2. Chuẩn bị dữ liệu (Tạo Dataset)

Chuẩn bị dữ liệu bao gồm chia dữ liệu thành các tập huấn luyện, xác thực và thử nghiệm, cũng như chuyển đổi dữ liệu sang định dạng phù hợp với YOLOv8.

Các bước chuẩn bị dữ liệu:

- Chia dữ liệu: Chia dữ liệu thành ba tập:
- Tập huấn luyện (train set): Dữ liệu được sử dụng để huấn luyện mô hình.
- Tập xác thực (validation set): Dữ liệu được sử dụng để đánh giá hiệu quả của mô hình trong quá trình huấn luyện.
- Tập thử nghiệm (test set): Dữ liệu được sử dụng để đánh giá hiệu quả của mô hình sau khi huấn luyện.
- Chuyển đổi dữ liệu: Chuyển đổi dữ liệu sang định dạng phù hợp với YOLOv8, chẳng hạn như định dạng YOLO Darknet.



Hình 3. 1. Tạo khung đối tượng

### 3.3.3. Quá trình huấn luyện

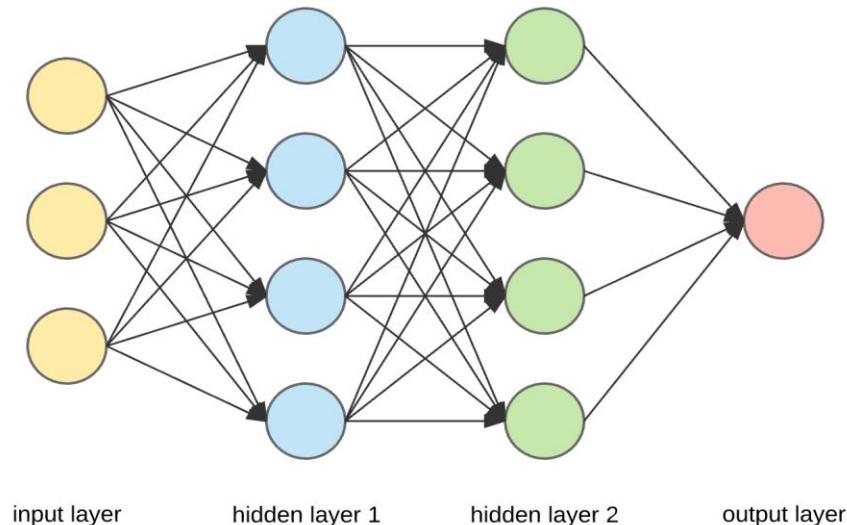
Quá trình huấn luyện là bước cuối cùng trong quá trình huấn luyện YOLOv8. Quá trình huấn luyện bao gồm việc sử dụng dữ liệu huấn luyện để cập nhật các tham số của mô hình. Quá trình huấn luyện có thể mất nhiều thời gian, tùy thuộc vào kích thước của dữ liệu huấn luyện và cấu hình của mô hình.

Thiết lập các tham số là bước quan trọng nhằm tối ưu hiệu suất và độ chính xác của mô hình. Các tham số này bao gồm:

- Batch size: Số lượng mẫu dữ liệu huấn luyện được truyền qua mạng trong mỗi lần lặp. Lựa chọn batch size phụ thuộc vào kích thước dữ liệu, phức tạp của mô hình và tài nguyên máy tính.
- Learning rate: Điều chỉnh kích thước của bước cập nhật trọng số trong quá

trình đào tạo. Learning rate lớn giúp học nhanh hơn nhưng có thể làm mô hình không ổn định, trong khi learning rate nhỏ làm chậm quá trình học nhưng giúp cải thiện độ ổn định và chính xác của mô hình.

- Epochs: Số lượt huấn luyện trên toàn bộ tập dữ liệu nhằm mục đích tối ưu hóa chính xác của mô hình bằng cách điều chỉnh các trọng số dựa trên lỗi dự đoán toàn phần.



Hình 3. 2. Mô hình mạng lưới neural đơn giản

### 3.4. Đánh giá hiệu quả của mô hình YOLOv8

#### 3.4.1. Các chỉ số đánh giá

Để đánh giá hiệu quả của mô hình YOLOv8, chúng ta cần sử dụng các chỉ số đánh giá phù hợp với nhiệm vụ. Một số chỉ số đánh giá phổ biến bao gồm:

- Accuracy (Độ chính xác): Đo lường tỷ lệ dự đoán chính xác của mô hình so với tổng số dự đoán. Tuy nhiên, độ chính xác không đủ để đánh giá khi mô hình phải đối mặt với dữ liệu mất ổn định.
- Precision (Độ chuẩn xác): Tỷ lệ các dự đoán tích cực thực sự trong số các dự đoán tích cực mà mô hình đưa ra. Đây là số liệu quan trọng trong các tình huống mà các kết quả dương tính giả có thể gây hậu quả nghiêm trọng.
- Recall (Khả năng thu hồi): Tỷ lệ các dự đoán tích cực thực sự trong số các điểm dữ liệu tích cực. Chỉ số này quan trọng trong các tình huống mà việc bỏ sót các trường hợp dương tính có thể có hậu quả nghiêm trọng.

- F1 Score: Là một chỉ số kết hợp giữa độ chính xác và khả năng thu hồi, cung cấp sự cân bằng giữa hai chỉ số trên và hữu ích khi cả độ chính xác và khả năng thu hồi đều quan trọng.

### **3.4.2. Kết quả đánh giá**

Kết quả đánh giá hiệu quả của mô hình YOLOv8 phụ thuộc vào nhiều yếu tố, bao gồm:

Bộ dữ liệu: Bộ dữ liệu được sử dụng để huấn luyện và đánh giá mô hình.

Cấu hình mô hình: Lựa chọn backbone, neck và head phù hợp với nhiệm vụ của bạn.

Kỹ thuật huấn luyện: Các kỹ thuật huấn luyện được sử dụng để cải thiện hiệu quả của mô hình.

Dưới đây là một số kết quả đánh giá hiệu quả của mô hình YOLOv8 trên các bộ dữ liệu phổ biến:

- Pascal VOC: mAP = 88.0%
- Microsoft COCO: mAP = 56.7%
- ImageNet: Top-1 accuracy = 78.1%

## **3.5 Thư viện OpenCV [21]**

OpenCV, một thư viện mã nguồn mở, được khởi đầu tại Intel bởi Gary Bradsky vào năm 1999, và phiên bản đầu tiên đã ra mắt vào năm 2000. Tính đến năm 2005, OpenCV đã được sử dụng trên chiếc xe Stanley, chiến thắng trong cuộc thi DARPA Grand Challenge. Sự phát triển tích cực của nó tiếp tục được thúc đẩy bởi Willow Garage, với sự lãnh đạo của Gary Bradsky và Vadim Pisarevsky.

OpenCV hỗ trợ rất nhiều thuật toán liên quan đến thị giác máy tính và học máy, và đang tiếp tục mở rộng tính năng của mình. Thư viện này hiện có sẵn trên nhiều nền tảng như Windows, Linux, OS X, Android và iOS, và hỗ trợ các ngôn ngữ lập trình như C++, Python, Java, v.v. Ngoài ra, OpenCV đang phát triển các giao diện dựa trên CUDA và OpenCL để tối ưu hiệu suất cho các hoạt động GPU.

Các tính năng chính của OpenCV bao gồm xử lý ảnh và video với nhiều định dạng khác nhau, điều chỉnh màu sắc và phát hiện biên và đối tượng. Nó cũng cung cấp các

công cụ phân đoạn ảnh và trích xuất đặc trưng để hỗ trợ việc huấn luyện mô hình máy học. Đồng thời, OpenCV cũng có khả năng phân tích dữ liệu video để phát hiện và theo dõi sự chuyển động của vật thể trong thời gian thực. Cuối cùng, nó còn được áp dụng trong lĩnh vực y tế để phân tích hình ảnh và hỗ trợ cho việc chẩn đoán.



Hình 3. 3. Thư viện OpenCV

### 3.6 Thư viện Pyzbar [22]

Pyzbar là một thư viện mã nguồn mở được sử dụng cho việc giải mã mã vạch từ hình ảnh hoặc video. Thư viện này được phát triển bởi Alex Bain và được phát hành lần đầu vào năm 2017. Pyzbar hỗ trợ các mã vạch phổ biến như mã vạch 1D (EAN-13, UPC-A, ISBN, vv.) và mã vạch 2D (QR code, Data Matrix, Aztec, vv.).

Pyzbar được viết bằng ngôn ngữ Python và sử dụng thư viện ZBar làm backend để thực hiện quá trình giải mã mã vạch. Nó hỗ trợ nhiều phiên bản Python từ 2.7 đến 3.8 và có thể được cài đặt thông qua pip, trên cả hệ điều hành Windows, Linux và macOS.

Các tính năng chính của Pyzbar bao gồm khả năng đọc và giải mã mã vạch từ hình ảnh hoặc video, đồng thời cung cấp thông tin về loại mã vạch được nhận diện và dữ liệu mã vạch tương ứng. Pyzbar cũng hỗ trợ việc xử lý ảnh trước khi giải mã, cho phép người dùng điều chỉnh các thông số như độ sáng, tương phản và tỉ lệ khung hình để cải thiện độ chính xác của quá trình giải mã.

### **3.7 YOLOv8 trong máy tính toán tự động**

#### **3.7.1 Phương pháp nhận diện vật thể và mã vạch dựa vào yolov8 kết hợp opencv và pyzbar**

Phương pháp nhận diện vật thể và mã vạch bằng phương pháp kết hợp YOLOv8, OpenCV và Pyzbar là một quá trình hiệu quả và linh hoạt trong xử lý hình ảnh và video. YOLOv8 đại diện cho một công nghệ nhận diện vật thể tiên tiến, đem lại khả năng nhận diện chính xác và nhanh chóng đối với các vật thể trong ảnh hoặc video. OpenCV là một thư viện phổ biến cho xử lý ảnh và video, cung cấp các công cụ cần thiết để đọc, hiển thị và xử lý video từ nhiều nguồn khác nhau. Pyzbar, một thư viện mã nguồn mở, chủ yếu được sử dụng để giải mã mã vạch từ hình ảnh hoặc video.

Quá trình này bắt đầu bằng việc đọc khung hình video từ nguồn như camera bằng OpenCV. Sau đó, các khung hình này được truyền vào mô hình YOLOv8 để thực hiện nhận diện vật thể. Kết quả của quá trình này bao gồm các hộp giới hạn quanh vật thể và nhãn tương ứng. Đồng thời, Pyzbar được áp dụng để giải mã mã vạch từ hình ảnh hoặc video, đưa ra thông tin về loại mã vạch và dữ liệu tương ứng.

Cuối cùng, kết quả từ cả hai quá trình nhận diện sẽ được hiển thị trên video hoặc ảnh gốc, tạo ra một trải nghiệm trực quan và dễ hiểu cho người dùng.

#### **3.7.2 Quá trình nhận diện vật thể**

Với sự hỗ trợ từ thư viện OpenCV, quá trình nhận diện vật thể dựa vào yolov8 sử dụng mô hình học sâu đã được huấn luyện trước để nhận diện các đối tượng trong ảnh hoặc video.

Quá trình nhận diện trái cây bằng YOLOv8 diễn ra như sau:

- Tải mô hình: Đầu tiên, cần tải mô hình YOLOv8 đã được huấn luyện trước cho nhiệm vụ nhận diện trái cây. Mô hình này có thể được tải từ các nguồn như Ultralytics, hoặc bạn có thể huấn luyện mô hình riêng của mình bằng cách sử dụng tập dữ liệu chứa ảnh trái cây đã được gán nhãn.

- Đọc ảnh hoặc video: Tiếp theo, đọc ảnh hoặc video chứa trái cây cần nhận diện bằng cách sử dụng thư viện OpenCV hoặc các thư viện khác.
- Thực hiện nhận diện: Sau khi đọc ảnh hoặc video, truyền nó vào mô hình YOLOv8 đã tải. Mô hình sẽ thực hiện quá trình nhận diện và trả về kết quả, bao gồm các hộp giới hạn (bounding boxes) quanh các đối tượng được nhận diện, tên đối tượng và độ tin cậy.
- Hiển thị kết quả: Cuối cùng, bạn có thể hiển thị kết quả nhận diện trên ảnh hoặc video, bao gồm các hộp giới hạn và tên đối tượng tương ứng.

### **3.7.3 Quá trình nhận diện mã vạch**

Quá trình nhận diện mã vạch trong mã nguồn Pyzbar bắt đầu bằng việc đọc các khung hình từ camera sử dụng OpenCV. Trong mỗi khung hình, thư viện Pyzbar được sử dụng để nhận diện và giải mã các mã vạch có thể xuất hiện. Mỗi khi một mã vạch được nhận diện, thông tin về loại mã vạch và dữ liệu tương ứng được trích xuất và hiển thị. Đồng thời, một hộp giới hạn được vẽ quanh mã vạch để thể hiện vị trí của nó trên khung hình. Quá trình này tiếp tục lặp lại liên tục, cho phép hệ thống nhận diện và xử lý mã vạch trong thời gian thực từ các khung hình mới nhất từ camera.

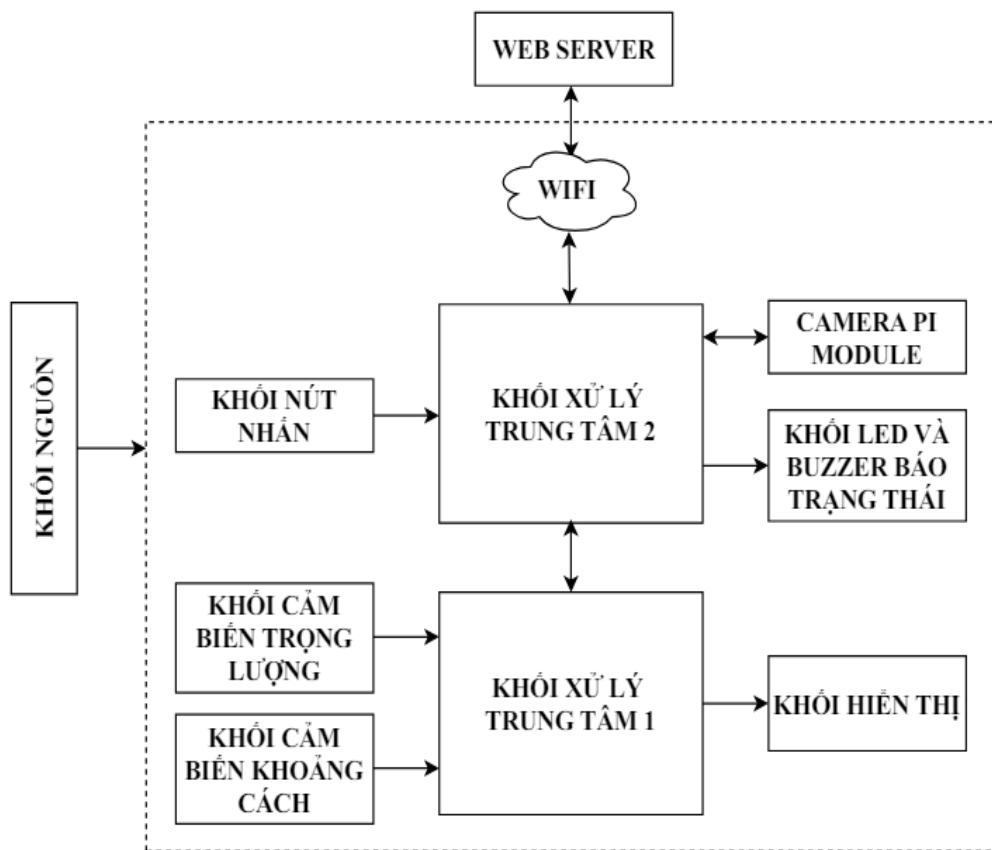
## CHƯƠNG 4: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

### 4.1 Yêu cầu hệ thống

Hệ thống thanh toán tự động này mang đến trải nghiệm mua sắm tiện lợi và hiệu quả cho khách hàng. Bằng cách sử dụng công nghệ nhận diện sản phẩm qua camera, khách hàng có thể tự thanh toán tại các cửa hàng mà không cần phải nhờ đến sự hỗ trợ từ nhân viên. Giao diện thanh toán được thiết kế thân thiện và dễ sử dụng, cho phép người dùng quét mã QR code để việc thanh toán trở nên nhanh chóng và thuận tiện.

### 4.2 Sơ đồ khái niệm hệ thống và chức năng từng khái niệm

#### 4.2.1 Sơ đồ khái niệm hệ thống



Hình 4. 1. Sơ đồ khái niệm hệ thống

#### 4.2.2 Chức năng từng khái

- Khối xử lý trung tâm 1:** Thu thập dữ liệu từ cân khối lượng. Nhận dữ liệu từ cảm biến khoảng cách HC-SR04 để phân biệt giữa cân 1 và cân 2. Gửi dữ liệu ra

màn hình LCD 16x2 có tích hợp I2C.

- **Khối xử lý trung tâm 2:** Là khối nhận công việc xử lý hình ảnh được gửi về được từ CAM Pi sau đó phân loại trái cây và nhận diện mã vạch dựa trên dữ liệu đã được chuẩn bị trước. Sau đó xử lí dữ liệu và gửi lên giao diện WEB thông qua giao tiếp API. Nhận tín hiệu từ nút nhấn để phân biệt các công việc như nhận diện trái cây, nhận diện mã vạch và gửi dữ liệu đi.
- **Khối cảm biến khối lượng:** Sẽ bao gồm hai cân được phân biệt dựa vào khoảng cách được đo từ bộ cảm biến khoảng cách HC-SR04.
- **Camera Pi module:** Từ hình ảnh thu được qua máy ảnh sẽ gửi ảnh về bộ xử lí trung tâm 2 từ bộ xử lí trung tâm 2 sẽ đưa ra dữ liệu nhận diện dựa vào dữ liệu đã được chuẩn bị trước.
- **Khối nút nhấn:** Sẽ bao gồm ba nút nhấn là nút nhấn gửi, nút nhấn nhận diện trái cây và nút nhấn nhận diện mã vạch. Nút nhấn nhận diện trái cây khi được nhấn sẽ thực hiện nhiệm vụ nhận diện trái cây thông qua hình ảnh được gửi về từ CAM PI. Nút nhấn nhận diện sẽ là loại nút nhấn giữ để bật tắt trạng thái nhận diện mã vạch sau đó sẽ tự gửi thông tin lên giao diện WEB nếu mã vạch hợp lệ. Nút nhấn gửi dùng để gửi dữ liệu nhận diện trái cây bao gồm mã của loại trái cây và cân nặng.
- **Web Server:** Lưu trữ tất cả dữ liệu của sản phẩm từ hình ảnh đại diện, giá, của loại trái cây. Đồng thời cũng là nơi lưu trữ dữ liệu mã vạch của các sản phẩm có mã vạch cần nhận diện.
- **Khối hiển thị:** Hiển thị cân nặng của loại trái cây thông qua khói xử lí trung tâm 1. Ở khói này sử dụng màn hình LCD 16x2 có tích hợp I2C.
- **Khối cảm biến khoảng cách:** sử dụng cảm biến đo khoảng cách HC-SR04 để tính toán khoảng cách từ cảm biến đến cân 1 hoặc cân 2. Từ đó có thể lựa chọn để lấy giá trị cân nặng của trái cây từ cân 1 hoặc cân 2.
- **Khối LED và BUZZER báo trạng thái:** sử dụng 2 led và 1 loa buzzer để báo trạng thái cho người dùng biết hệ thống đang ở trạng thái nhận diện trái cây hay mã vạch. Loa buzzer báo cho người dùng biết đã nhận diện được mã vạch hay chưa.

- **Khối nguồn:** Cung cấp nguồn cho toàn bộ hệ thống bao gồm khói xử lí trung tâm 1 và 2, cân, đèn led cảnh báo, cảm biến đo khoảng cách HC-SR04, CAM PI, nút nhấn và màn hình LCD.

### 4.3 Hoạt động của hệ thống

Ban đầu khi cấp nguồn cho hệ thống, hệ thống sẽ chạy những khởi tạo ban đầu bao gồm hiển thị thông tin lên màn hình LCD về cân nặng, khởi tạo giá trị của 2 cân về 0 KG. Sau đó nhóm sẽ thực hiện thao tác mở các chương trình như giao diện WEB, chương trình nhận diện trái cây và mã vạch. Chương trình sẽ hoạt động đúng chức năng chỉ khi ta sử dụng nút nhấn. Nếu ta nhấn nút nhấn nhận diện trái cây thì hệ thống sẽ bắt đầu nhận diện trái cây cũng như lấy giá trị từ cân ngay thời điểm đó từ khói xử lí trung tâm 1. Sau đó khi ta nhấn nút nhấn gửi thì dữ liệu về mã trái cây và cân nặng của trái cây sẽ được gửi lên giao diện WEB từ đó trong giỏ hàng trên giao diện Web sẽ xuất hiện mặt hàng trái cây ta vừa nhận diện cùng với cân nặng của nó.

Khi ta nhấn nút nhấn nhận diện mã vạch thì hệ thống sẽ bắt đầu nhận diện mã vạch của các sản phẩm có sẵn mã vạch. Sau khi nhận diện thành công hệ thống sẽ tự động gửi thông tin của sản phẩm lên giao diện WEB.

Để cho người dùng có thể biết về trạng thái hoạt động của hệ thống thì nhóm có tích hợp thêm đèn led để thông báo cho người dùng biết hệ thống đang ở trạng thái nào. Khi nhận diện trái cây ta nhấn nút đèn led cũng sẽ sáng cùng báo hiệu cho người dùng biết hệ thống đang hoạt động chức năng nhận diện trái cây. Khi ta nhấn nút nhận diện mã vạch sẽ sáng cho đến khi người dùng tắt chức năng đi bằng cách nhấn lại 1 lần nữa vào phím nhấn đó. Đồng thời khi nhận diện mã vạch thành công cũng sẽ có 1 loa Buzzer kêu lên để báo hiệu cho người dùng biết là đã nhận diện được sản phẩm thành công.

### 4.4 Thiết kế phần cứng

#### 4.4.1. Khối cảm biến khối lượng

Cảm biến đo khối lượng là một thiết bị có khả năng chuyển đổi khối lượng thành tín hiệu điện tương ứng. Nguyên lý hoạt động của cảm biến này dựa trên sự thay đổi điện trở trong cầu điện trở, điều này dựa trên nguyên lý áp lực - trở kháng. Khi một lực tác

động lên cảm biến, điện trở của nó thay đổi, điều này dẫn đến sự biến thiên trong tín hiệu điện đầu ra của cảm biến. Trong hệ thống này, nhóm đã quyết định sử dụng hai cân song song để tăng tính linh hoạt của hệ thống và cũng như để đáp ứng nhu cầu thực tế, cho phép giảm bớt thời gian xử lý trong quá trình thanh toán.

Trong quá trình thiết kế nhóm đã có hai phương án để hệ thống có thể tăng hiệu năng cũng như giảm thời gian cho khách hàng trong quá trình thanh toán.

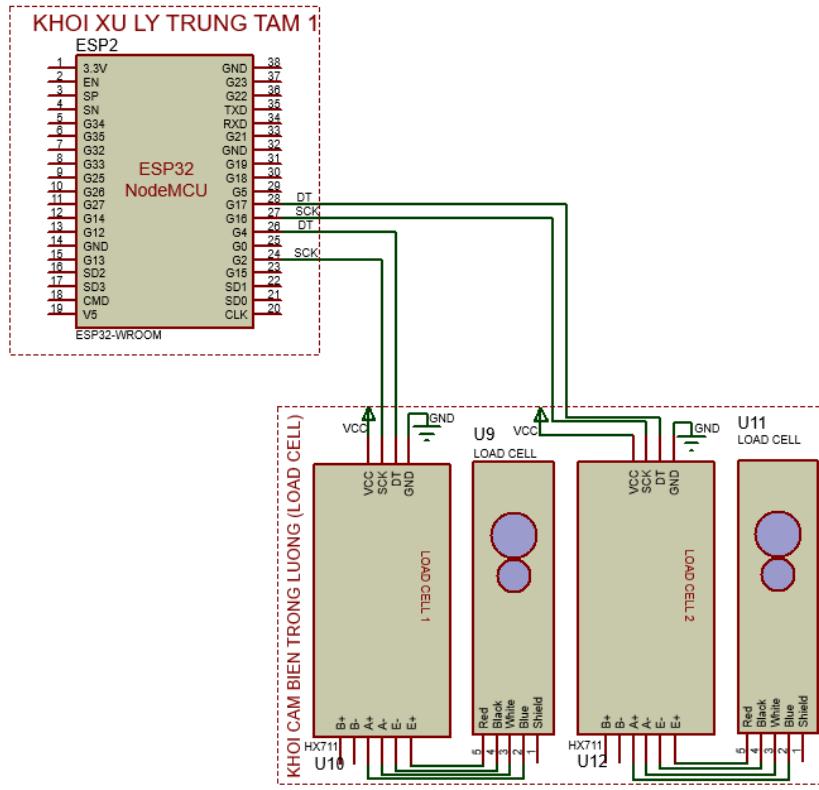
Ở phương án đầu tiên khi nhóm nghĩ tới sẽ là sử dụng hai cân đi kèm đó cũng sẽ là hai khối xử lí trung tâm 2 Raspberry Pi4 và hai Cam Pi. Khi đưa ra thực tế thì hệ thống sẽ hoạt động tối ưu hơn vì có thể xử lí cùng lúc cho hai khách hàng. Nhưng vì do kinh phí còn hạn hẹp nên nhóm chưa thể thực hiện mô hình hệ thống bao gồm có hai khối xử lí trung tâm 2 Raspberry Pi4 và hai Cam Pi nên nhóm đã nghĩ đến một phương án khác.

Ở phương án thứ 2 nhóm nghĩ tới việc sẽ chỉ sử dụng một khối xử lí trung tâm 2 Raspberry Pi4 và một Cam Pi nhưng việc này cũng sẽ đưa ra một vấn đề là nhóm phải làm sao để hệ thống có thể phân biệt được giữa hai cân. Khi đó nhóm đã quyết định sử dụng thêm một cảm biến khoảng cách HC-SR04.

Mô tả kết nối chân của khối cảm biến khoảng cách sẽ được miêu tả rõ hơn trong mục Khối cảm biến khoảng cách

Mô tả kết nối cho khối cảm biến khối lượng bao gồm:

- Chân IO4 sẽ kết nối với chân DT của cảm biến khối lượng 1 và chân IO2 sẽ kết nối với chân SCK của cảm biến khối lượng 1.
- Chân IO17 sẽ kết nối với chân DT của cảm biến khối lượng 2 và chân IO16 sẽ kết nối với chân SCK của cảm biến khối lượng 2.



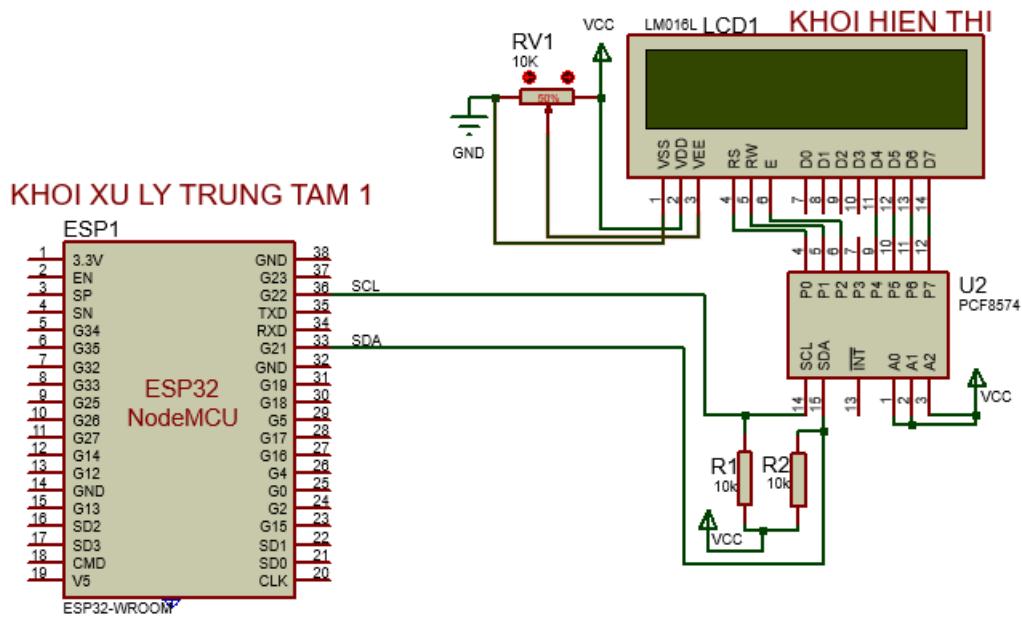
Hình 4. 2. Sơ đồ nguyên lý kết nối cảm biến khối lượng với ESP32

#### 4.4.2. Khối hiển thị

Trong khối hiển thị bằng màn hình LCD thì nhóm đã sử dụng màn hình LCD 16x2 có tích hợp module chuyển đổi I2C. Việc kết hợp giữa màn hình LCD và module chuyển đổi I2C giúp tiết kiệm chân trong việc hiển thị với LCD. Và cũng giúp cho mạch được gọn gàng hơn và mang tính thẩm mỹ hơn. Ở khối hiển thị dùng màn hình LCD nhóm đã hiển thị thông tin về cân nặng của loại trái cây.

Về kết nối phần cứng. LCD tích hợp I2C được gói gọn lại chỉ còn bốn pin. Về chân VCC sẽ nối với nguồn 5V, chân GND sẽ nối ở mức 0V. Còn hai chân tín hiệu còn lại là SCL và SDA thì SCL sẽ được nối với IO22 và SDA sẽ được nối với IO21. Hai chân SCL và SDA là hai chân của chuẩn giao tiếp I2C trên ESP32, trong đó chân SDA sẽ là chân truyền dữ liệu đi và chân SCL sẽ là chân xung Clock. Chân SCL có tác dụng đồng bộ hóa việc truyền dữ liệu đi giữa các thiết bị. Việc tạo ra xung clock đó sẽ dựa vào thiết bị chủ hay còn được gọi là Master.

Mô tả cụ thể về cách kết nối được mô tả cụ thể trong hình 4.5.



Hình 4. 3. Sơ đồ nguyên lý kết nối màn hình LCD với ESP32

Ở hình 4.5, khi đã kết nối chân nguồn và chân dữ liệu của màn hình LCD với ESP32, có thể sử dụng thư viện LiquidCrystal\_I2C có sẵn để điều khiển màn hình. Thư viện này giúp hiển thị các dữ liệu và thông tin theo mong muốn của người dùng lên màn hình LCD một cách thuận tiện.

Và để sử dụng được LCD 16x2 có tích hợp I2C trong phần mềm lập trình Arduino sẽ phải cần thêm thư viện : Wire.h và LiquidCrystal\_I2C.h .

#### 4.4.3. Khối nút nhấn

Nút nhấn được sử dụng để người dùng có thể dễ dàng kiểm soát hệ thống và sản phẩm khi thực hiện thanh toán. Với kích thước nhỏ gọn và khả năng hoạt động lâu dài mà không gặp sự cố, nút nhấn có thể tương thích với đa số mạch điện và dễ dàng tích hợp vào hệ thống. Trong quá trình thiết kế nhóm có sử dụng thêm điện trở 10k và để giải thích cho việc này thì để xác định rõ ràng một mức logic chúng ta sẽ cần một điện áp tham chiếu 5V khi không nhấn nút và khi ta nhấn nút sẽ là 0V. Nhưng khi không gắn thêm điện trở sẽ có một vấn đề đó là khi không nhấn nút điện áp đầu vào sẽ là 5V nhưng khi ta nhấn nút thì sẽ xảy ra hiện tượng ngắn mạch. Và để giải quyết vấn đề này ta cần phải đặt một điện trở giữa dây trung tính, nút bấm và nguồn. Để tính toán

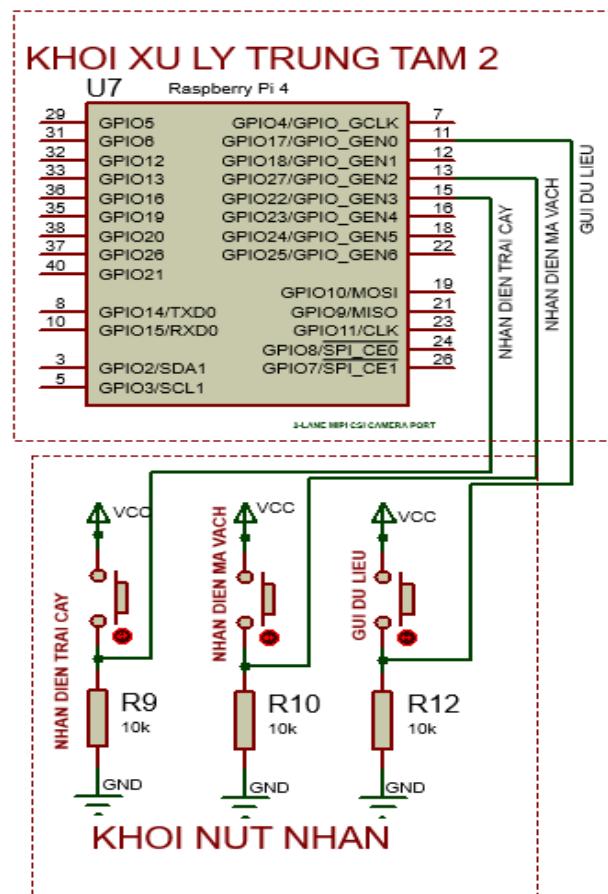
giá trị điện trở thì ta sẽ cần dòng cho phép trên mỗi trân I/O của một vi điều khiển. Và thông số này thường chỉ nằm trong khoảng 10-20mA. Để tính toán cần sử dụng công thức  $I=V/R$

$$I=5V/10000\text{ Ohms}=0.5(\text{mA})$$

Do vậy nên điện trở kéo xuống trong mạch sẽ là 10k.

Hình 4.6 dưới đây miêu tả các kết nối của các nút nhấn.

- Chân IO17 sẽ nối với nút nhấn có chức năng gửi dữ liệu lên giao diện WEB.
- Chân IO27 sẽ nối với nút nhấn có chức năng nhận mã vạch.
- Chân IO22 sẽ nối với nút nhấn có chức năng nhận điện trái cây.

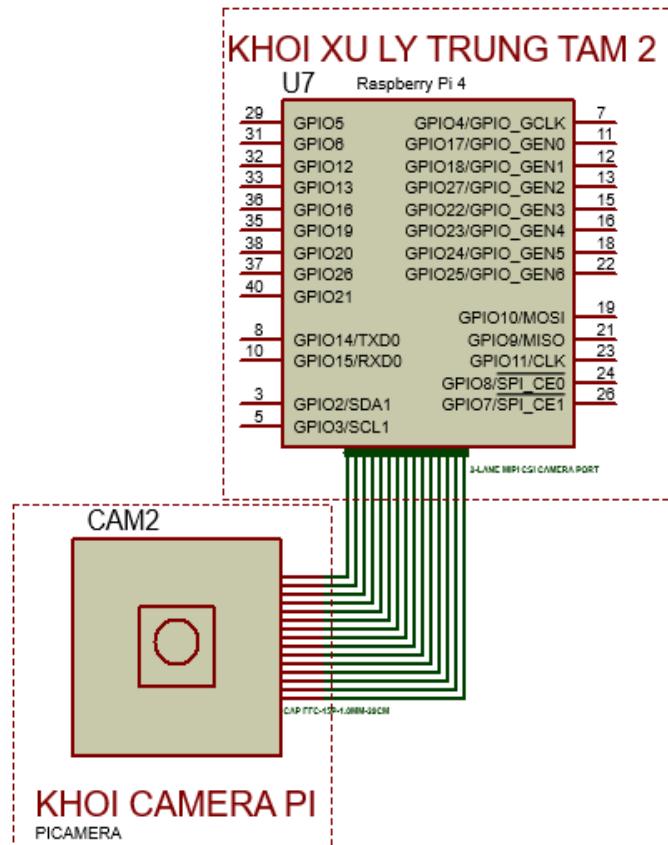


Hình 4.4. Sơ đồ nguyên lý kết nối nút nhấn với Raspberry Pi

#### 4.4.4. Camera Pi Module

Có nhiều phương thức để kết nối camera với Raspberry Pi thông qua các loại giao tiếp

khác nhau, nhưng Raspberry Pi có sẵn cổng Camera Module Port để kết nối camera một cách thuận tiện. Sử dụng cổng này giúp đảm bảo sự ổn định hơn so với việc sử dụng cổng USB hoặc các phương thức khác để kết nối camera. Camera Pi Module mang lại sự tiện lợi, ổn định và hiệu quả khi chụp ảnh, quay video và thu thập dữ liệu hình ảnh trên Raspberry Pi. Để kết nối giữa Raspberry Pi4 và Cam Pi thì ta sẽ cắm dây cáp của Cam Pi vào cổng Camera Module Port của Raspberry Pi 4. Cụ thể hơn thì trên Raspberry Pi 4 có hỗ trợ một cổng máy ảnh MIPI CSI 2-lane được sử dụng để kết nối trực tiếp với máy ảnh mà không cần các giao tiếp thứ ba nào can thiệp. Nhóm sẽ sử dụng cáp FFC để kết nối giữa Raspberry Pi4 và Cam Pi. Cáp FFC là một loại cáp mỏng và rất linh hoạt có thể truyền dữ liệu và nguồn điện trong một không gian chật hẹp. Điều này sẽ cho phép các thiết bị điện tử nhỏ gọn và hiệu quả hơn. Ngoài ra một ưu điểm khác của cáp FFC là về khả năng truyền dữ liệu ở tốc độ cao. Điều này giúp cho việc nhận diện hình ảnh sẽ tăng hiệu năng.



Hình 4. 5. Sơ đồ nguyên lý kết nối Camera Pi với Raspberry Pi

#### **4.4.5. Khối cảm biến khoảng cách**

Ở khói này để cho hệ thống phân biệt được giá trị cân nặng giữa cân 1 và cân 2 thì nhóm quyết định sử dụng một cảm biến HC-SR04 để đo khoảng cách từ CAM PI đến cân 1 và từ CAM PI đến cân 2. Cảm biến HC-SR04 được sử dụng khá rộng rãi một phần vì giá thành rẻ và mang lại độ chính xác. Cảm biến sẽ sử dụng sóng siêu âm để có thể đo khoảng cách từ khoảng 2 đến 300 cm. Nhóm đã thực hiện gắn cảm biến khoảng cách HC-SR04 vào vị trí trên cao ngay kế bên vị trí của Cam Pi để thuận tiện cho việc phân biệt. Hệ thống sẽ phân biệt giữa hai cân dựa vào vị trí của cảm biến đến cân 1 và từ vị trí của cảm biến đến cân 2. Khoảng cách từ cảm biến đến vị trí của cân 1 sẽ là 30cm và từ cảm biến đến vị trí cân 2 sẽ là 10cm.

Việc tính toán vị trí của cảm biến đến cân sẽ được thực hiện như sau:

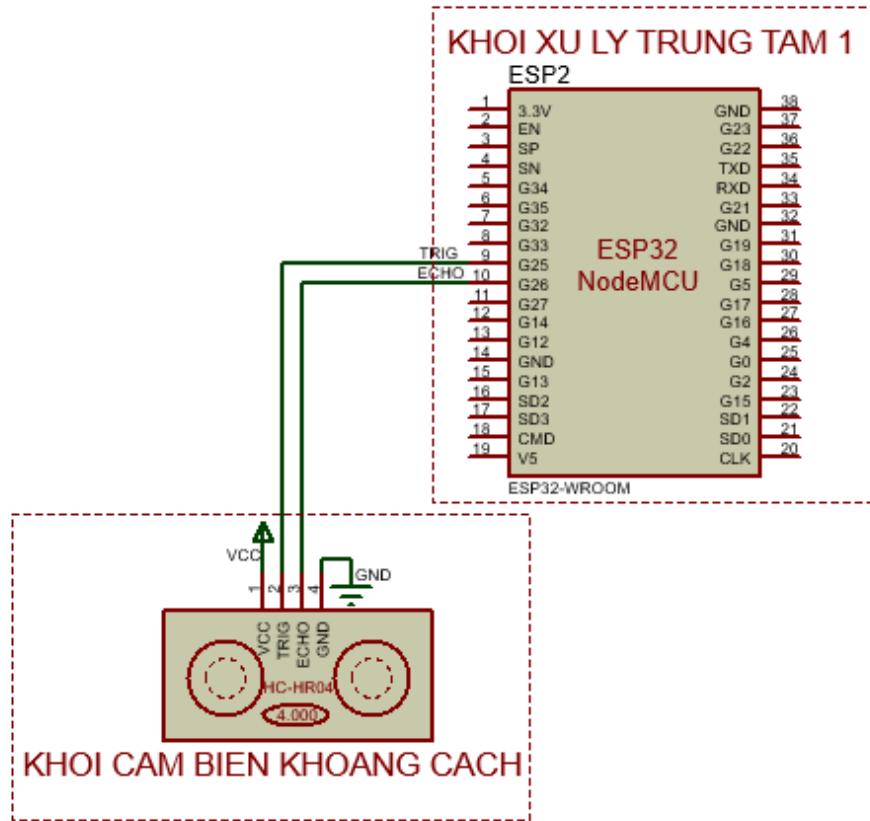
Nhóm sẽ sử dụng hàm “pulseIn()” đã được xây dựng sẵn trong Arduino để đo độ rộng của xung. Và kết quả trả về sẽ bằng độ dài xung ở mức cao ở chân echo được tính theo micro giây. Cụ thể thì hàm này sẽ đọc một xung tín hiệu số ở mức cao hoặc thấp tùy thuộc vào việc lập trình và sau đó sẽ trả về một chu kỳ của xung tín hiệu. Ở đây nhóm đã quyết định sẽ sử dụng độ dài xung ở mức cao. Sau đó thời gian truyền sóng đến vật sẽ là kết quả được trả về từ hàm “pulseIn()” chia cho 2 rồi sau đó sẽ chia tiếp cho 29.412 để tính toán khoảng cách.

$$\text{Distance} = \text{int}(pulseIn()) / 29.412$$

Việc sử dụng hai cân theo cách này sẽ mang lại hiệu quả cao trong việc mô phỏng và giảm đi khá nhiều trong chi phí. Tăng hiệu năng trong việc thanh toán cho khách hàng trong thực tế.

Kết nối giữa ESP32 và cảm biến khoảng cách HC-SR04 sẽ được miêu tả trong hình 4.8.

- Chân IO25 của ESP32 sẽ nối với chân TRIG của cảm biến HC-SR04.
- Chân IO26 của ESP32 sẽ nối với chân ECHO của cảm biến HC-SR04.



Hình 4. 6. Miêu tả kết nối của khối cảm biến khoảng cách.

#### 4.4.6. Khối LED và BUZZER báo trạng thái

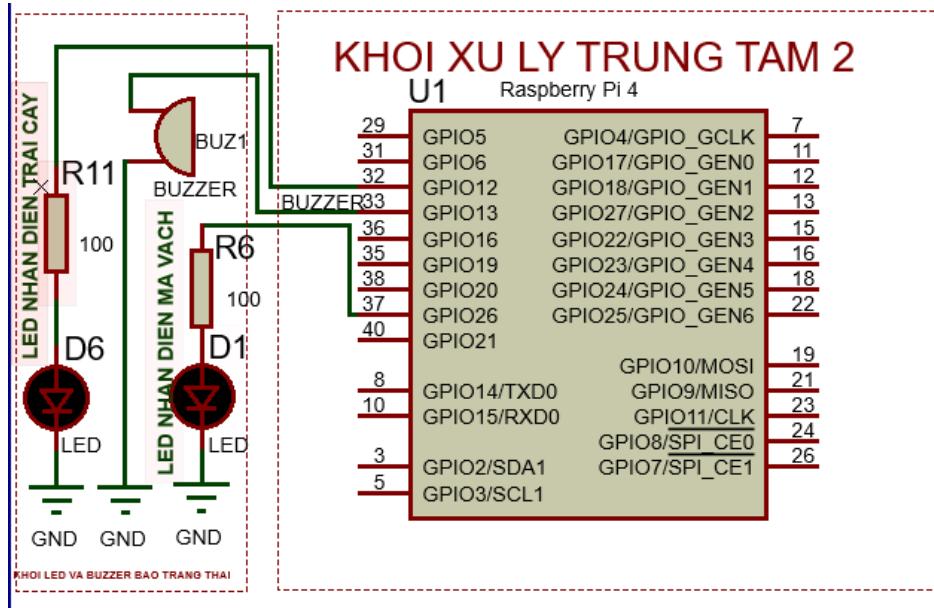
Để cho người dùng biết được hệ thống có đang hoạt động đúng chức năng không thì nhóm quyết định sử dụng thêm hai led đơn và một buzzer. Hai led đơn sẽ được sử dụng để báo cho người dùng biết hệ thống đang hoạt động ở trạng thái nhận diện trái cây hay mã vạch. Một buzzer được sử dụng để báo cho người dùng biết hệ thống đã nhận diện được mã vạch hay chưa. Nếu hệ thống đã nhận diện được mã vạch thì loa buzzer sẽ kêu lên một tiếng.

Trong quá trình thi công nhóm đã sử dụng thêm điện trở có giá trị 100 Ohm. Để giải thích cho điều này thì thông thường Led sẽ hoạt động ở mức từ 1.8 đến 3V với dòng 10 đến 20mA. Và nếu ta lấy Led cắm vào nguồn 5V thì sẽ dễ làm Led hỏng vì dòng chạy qua Led cao hơn dòng định mức cho phép của Led. Vì thế để bảo vệ Led ta nên sử dụng

thêm điện trở. Và giá trị của điện trở sẽ được xác định bằng công thức  $(5-3V)/20mA = 0.1k$  ( $\Omega$ ) = 100  $\Omega$ m.

- Chân IO12 của Raspberry Pi 4 sẽ được nối với led nhận diện trái cây.
- Chân IO26 của Raspberry Pi4 sẽ được nối với led nhận diện mã vạch.
- Chân IO13 của Raspberry Pi4 sẽ được nối với buzzer.

Hình 4.8 dưới đây sẽ miêu tả kết nối.



Hình 4.7. Miêu tả kết nối của khói Led và Buzzer báo trạng thái.

#### 4.4.7. Khối xử lý trung tâm 1

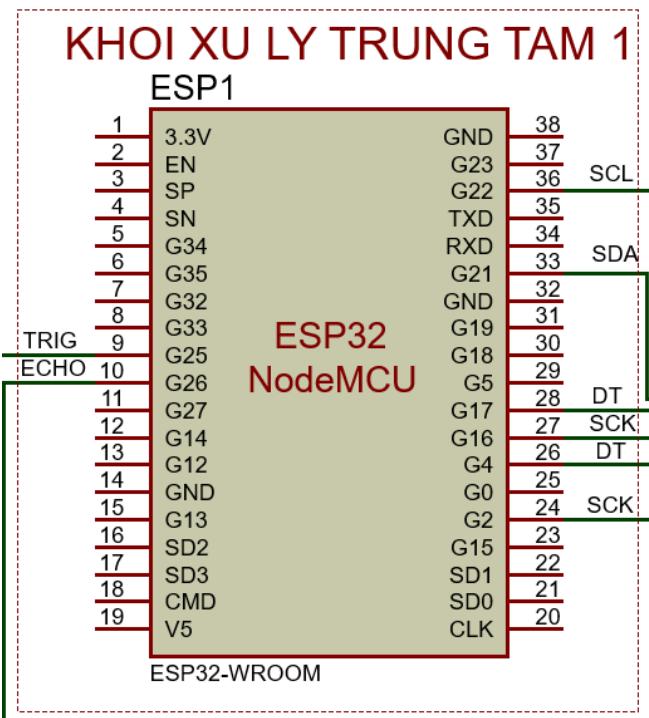
Bởi vì khói xử lý trung tâm sẽ đảm nhiệm chức năng chính trong toàn hệ thống và có khả năng kết nối wifi nên nhóm đã quyết định sử dụng ESP32 để làm khói xử lý trung tâm. Và ESP32 cũng đảm bảo số chân I/O và hoạt động ổn định.

Mô tả kết nối phần cứng giữa các ngoại vi và ESP32

- Chân IO22 sẽ kết nối với chân SCL và chân IO21 sẽ kết nối với chân SDA của màn hình LCD 16x2 tích hợp module chuyển đổi I2C.
- Chân IO4 sẽ kết nối với chân DT của cân khói lượng 1 và chân IO2 sẽ kết nối với chân SCK của cân khói lượng 1.

- Chân IO17 sẽ kết nối với chân DT của cân khối lượng 2 và chân IO16 sẽ kết nối với chân SCK của cân khối lượng 2.
- Chân IO25 sẽ kết nối với chân TRIG và chân IO26 sẽ kết nối với chân ECHO của cảm biến đo khoảng cách HC-SR04.

Hình 4.2 dưới đây sẽ miêu tả cụ thể các kết nối chân với khối xử lý trung tâm.



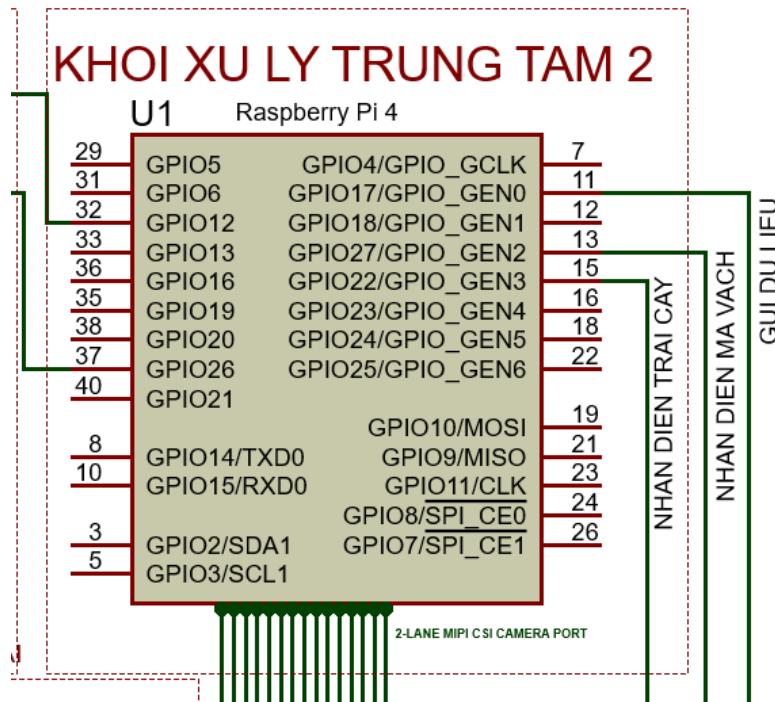
Hình 4. 8. Sơ đồ chân của ESP32

#### 4.4.8. Khối xử lý trung tâm 2

Raspberry Pi đã phổ biến cho việc lập từ xa các thiết bị. Nhờ kích thước nhỏ gọn, nó có thể được sử dụng để điều khiển máy in không dây, hệ thống nhà thông minh, và nhiều ứng dụng khác. Ngoài ra, Raspberry Pi cũng có thể được sử dụng như một VPN cá nhân hoặc được biến thành nhiều loại thiết bị khác nhau, từ phần cứng đến phần mềm. Sơ đồ chân của Raspberry Pi 4 được minh họa trong hình 4.3. Mô tả kết nối cho Raspberry Pi bao gồm:

- Chân IO22, IO27 và IO17 lần lượt kết nối với các chân của nút nhấn.
- Chân IO12, IO26 và IO13 sẽ lần lượt được nối với led và buzzer thông báo trạng thái bao gồm hai trạng thái là khi đang nhận diện trái cây hoặc đang nhận diện mã vạch.

- CAM PI sẽ được kết nối với Raspberry Pi thông qua cổng camera trên board.



Hình 4. 9. Sơ đồ chân của Raspberry Pi

#### 4.4.9. Khối nguồn

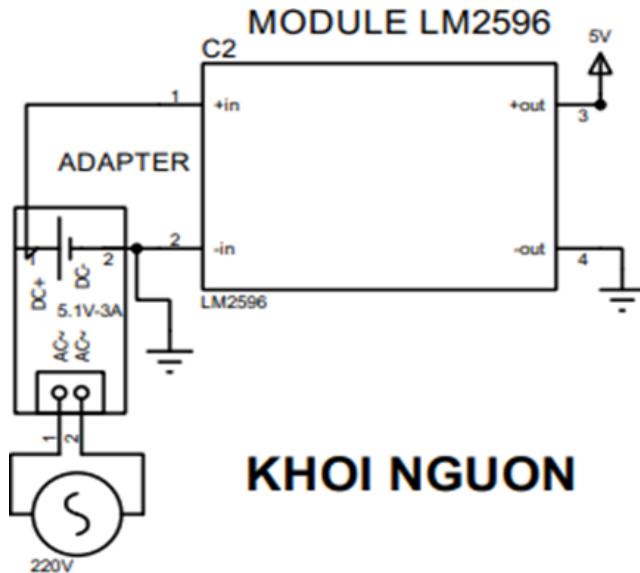
Sau khi đã hoàn thành các khối chức năng được miêu tả trong sơ đồ khái niệm hệ thống. Nhóm tiến hành tính toán để chọn nguồn cung cấp cho toàn bộ hệ thống. Việc tính toán này sẽ mang lại tính ổn định cho toàn bộ hệ thống. Và đảm bảo cho hệ thống được vận hành mượt mà nhất. Để thực hiện việc thiết kế khái niệm, nhóm đã liệt kê ra toàn bộ linh kiện sẽ sử dụng trong hệ thống. Từ đó dựa vào dòng tiêu thụ và điện áp sử dụng của từng linh kiện nhóm quyết định chọn adapter làm nguồn để cung cấp điện cho hoạt động của hệ thống.

Bảng 4. 1.Thông số về điện áp, dòng điện tiêu thụ của các linh kiện cảm biến

Tên linh kiện	Điện áp hoạt động	Dòng điện tiêu thụ
ESP32	5V	0.008A
Raspberry Pi4	5V	3.0A
LCD – I2C 16x2	5V	0.0025A

Loadcell Module	5V	0.001A
Cảm biến HC-SR04	5V	0.04A
Tổng		3.0515A

Dựa vào tính toán nhóm chọn sử dụng Adapter nguồn 5V-3A cấp nguồn cho hệ thống.



Hình 4. 10. Sơ đồ nguyên lý nguồn 5V

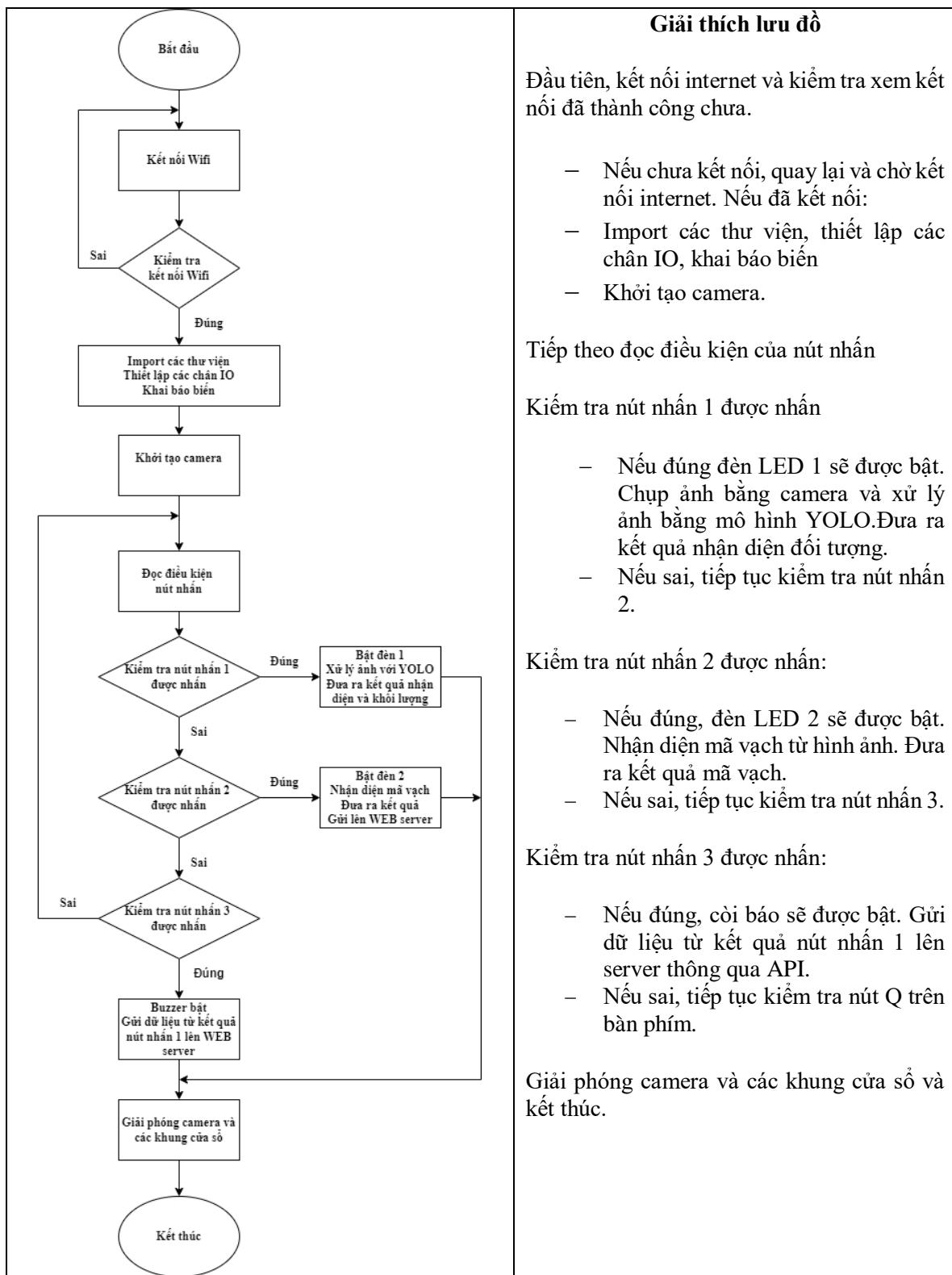
#### 4.5 Sơ đồ nguyên lý toàn hệ thống

Sau quá trình tính toán, thiết kế và chọn linh kiện phù hợp nhóm thực hiện đã thu được sơ đồ nguyên lý như sau:

(Sơ đồ nguyên lý toàn hệ thống được đính kèm theo khổ A4)

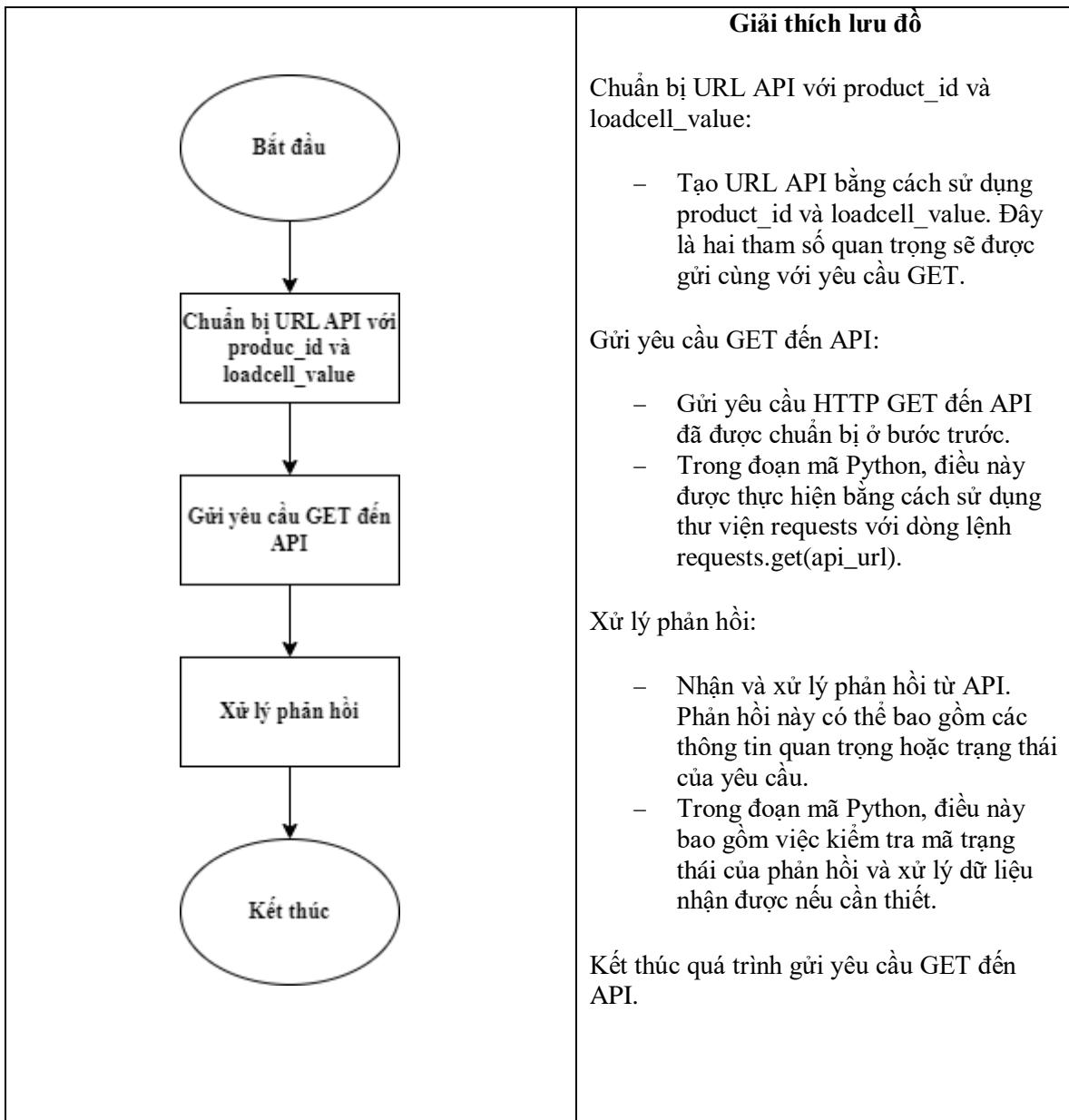
## 4.6 Thiết kế phần mềm

### 4.6.1 Lưu đồ giải thuật chương trình chính



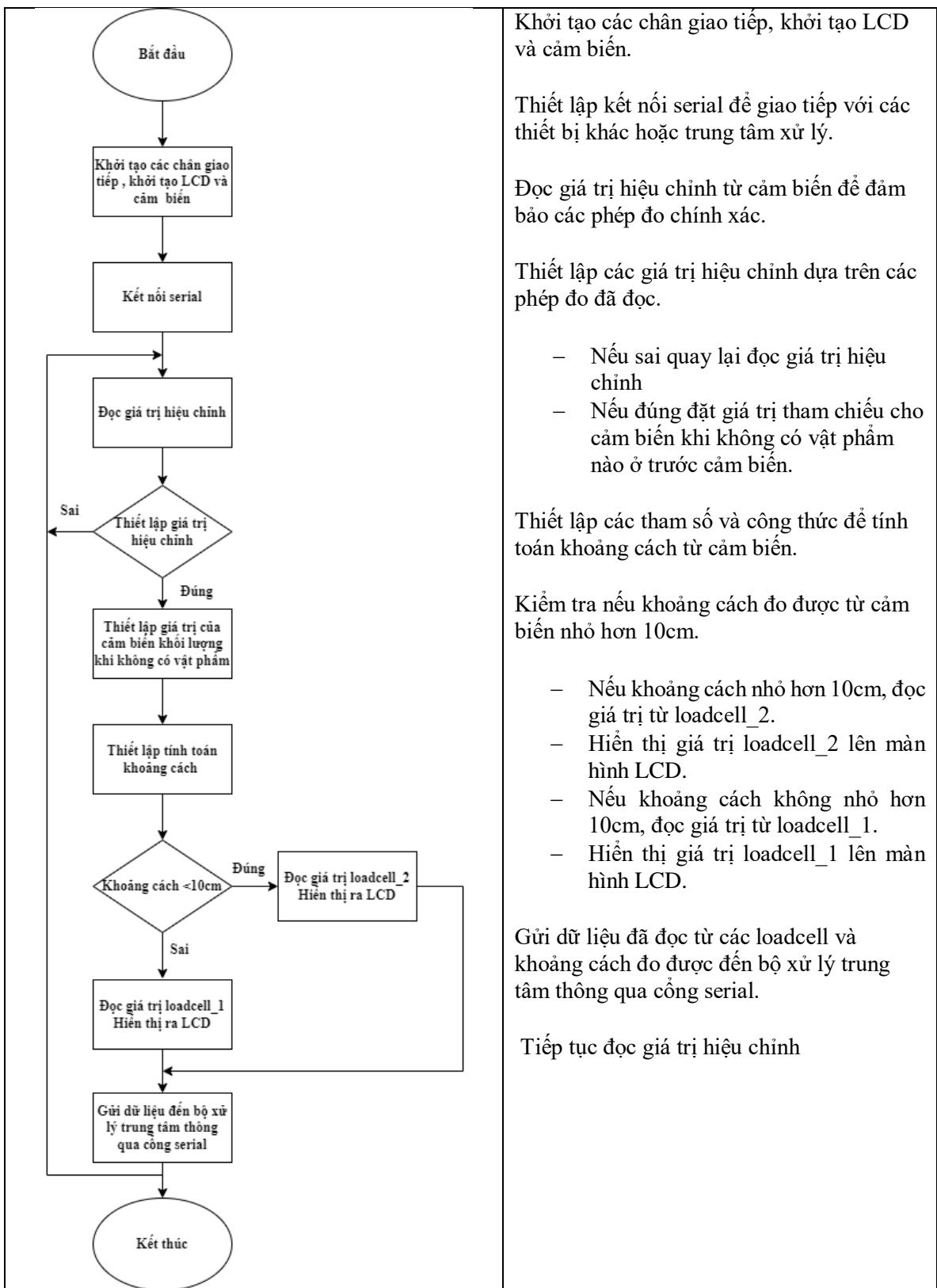
Hình 4. 11. Lưu đồ giải thuật chương trình chính

#### 4.6.2 Lưu đồ giải thuật chương trình con gửi địa chỉ API



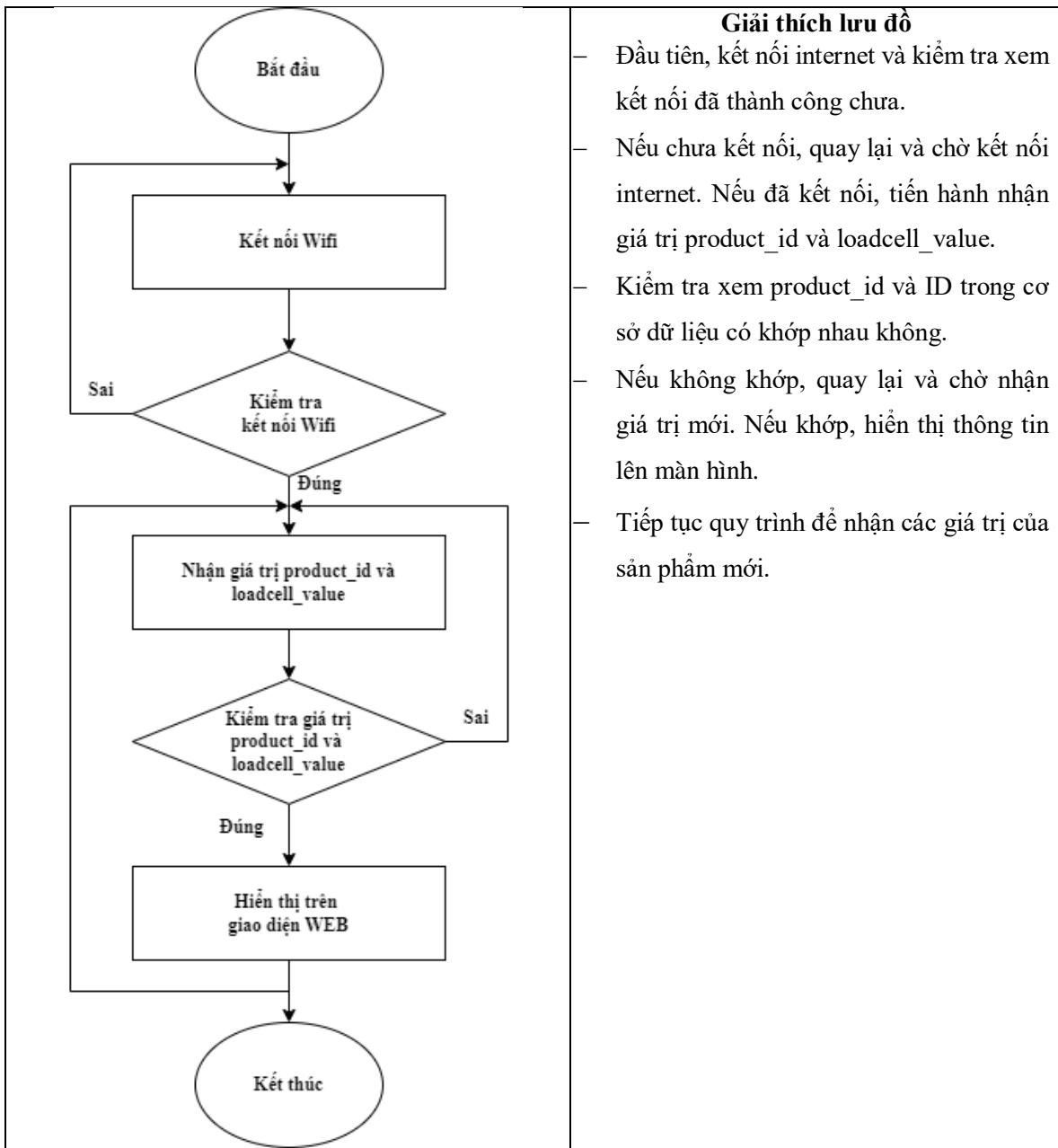
Hình 4. 12. Lưu đồ giải thuật chương trình con gửi API

### 4.6.3 Lưu đồ giải thuật chương trình ESP32



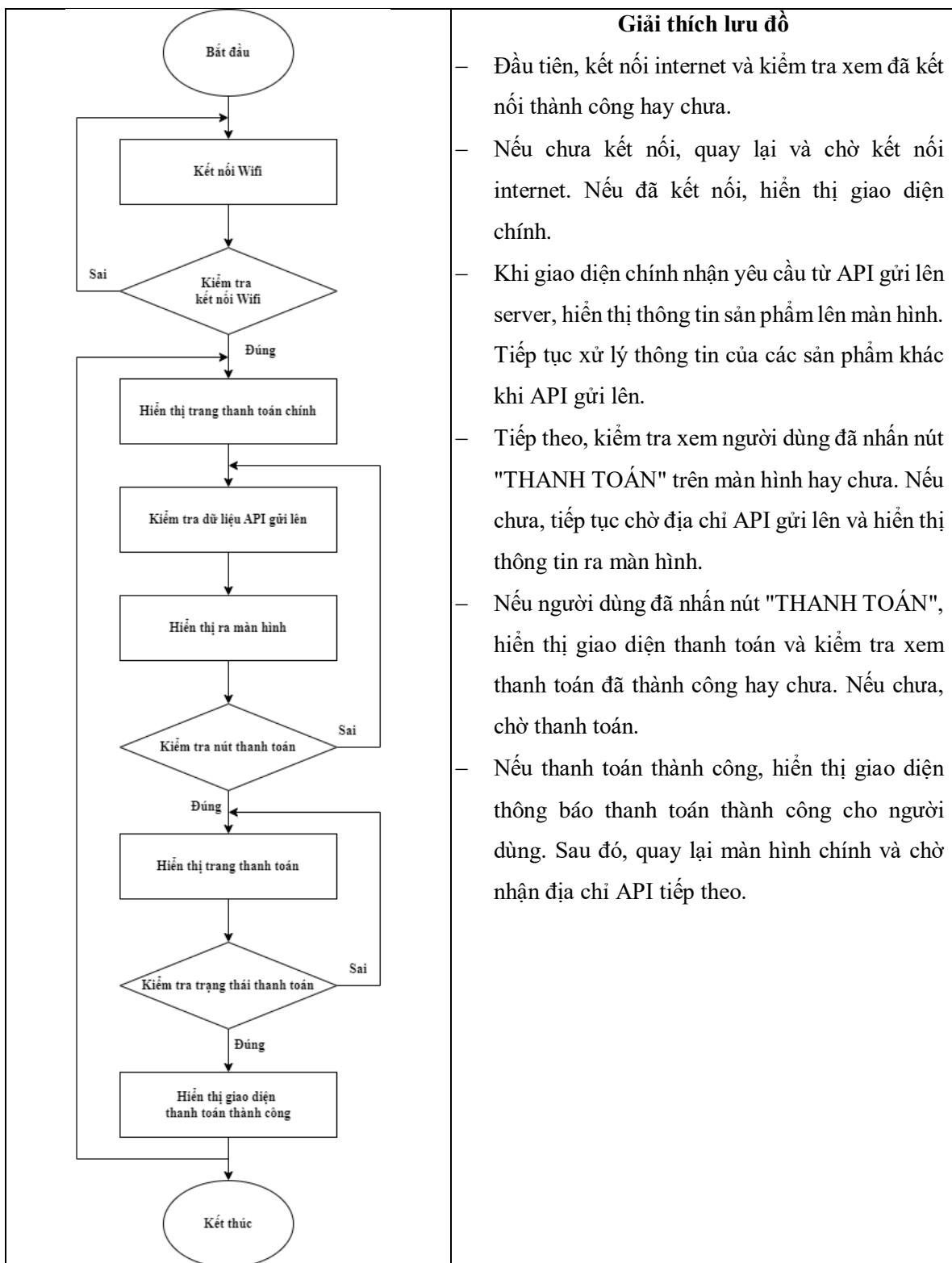
Hình 4. 13. Lưu đồ giải thuật chương trình ESP32

#### 4.6.4 Lưu đồ giải thuật chương trình con từ địa chỉ API gửi tới server



Hình 4. 14. Lưu đồ giải thuật từ địa chỉ API gửi đến server

#### 4.6.5 Lưu đồ giải thuật hoạt động của giao diện Web

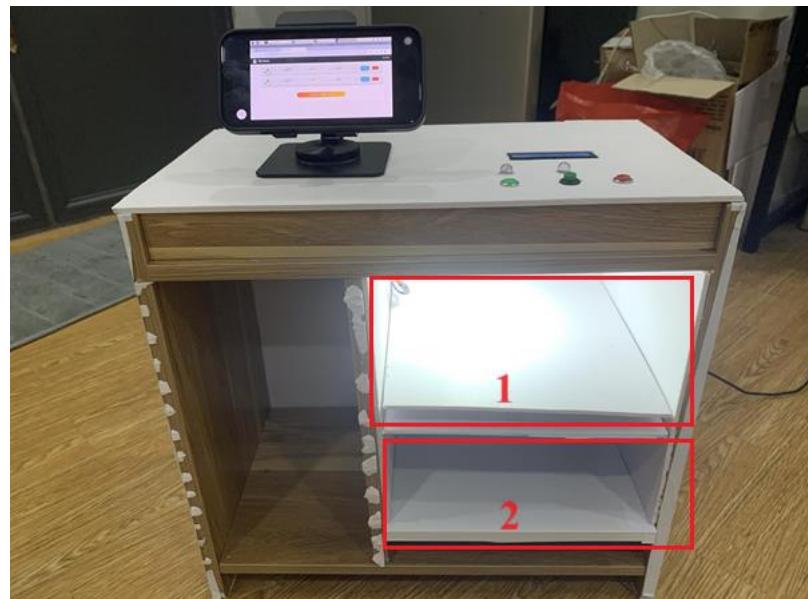


Hình 4. 15. Lưu đồ giải thuật hoạt động WEB

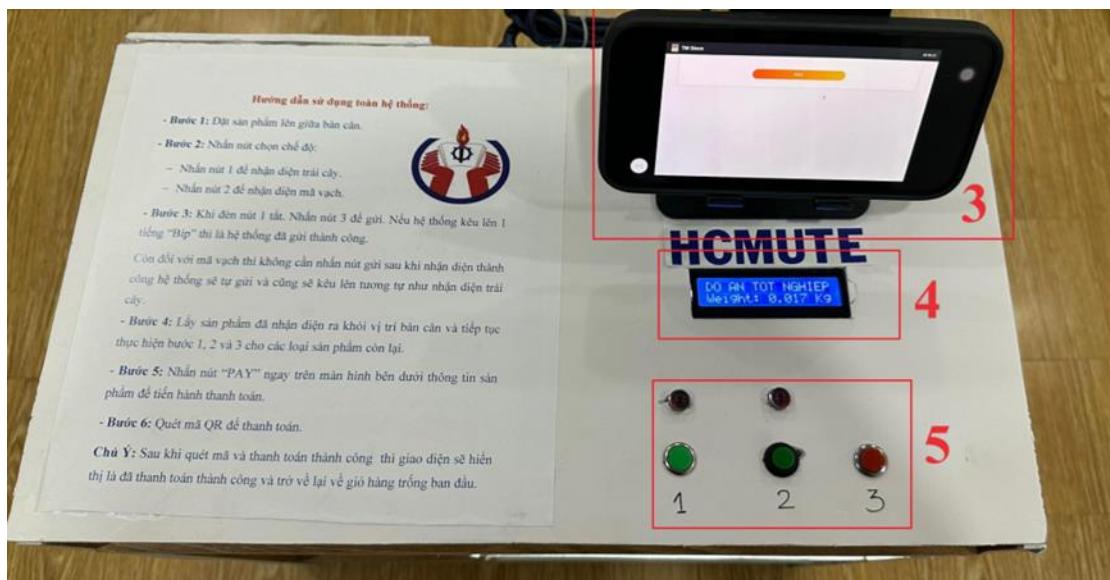
## CHƯƠNG 5: KẾT QUẢ THỰC HIỆN HỆ THỐNG

### 5.1 Mô hình hệ thống

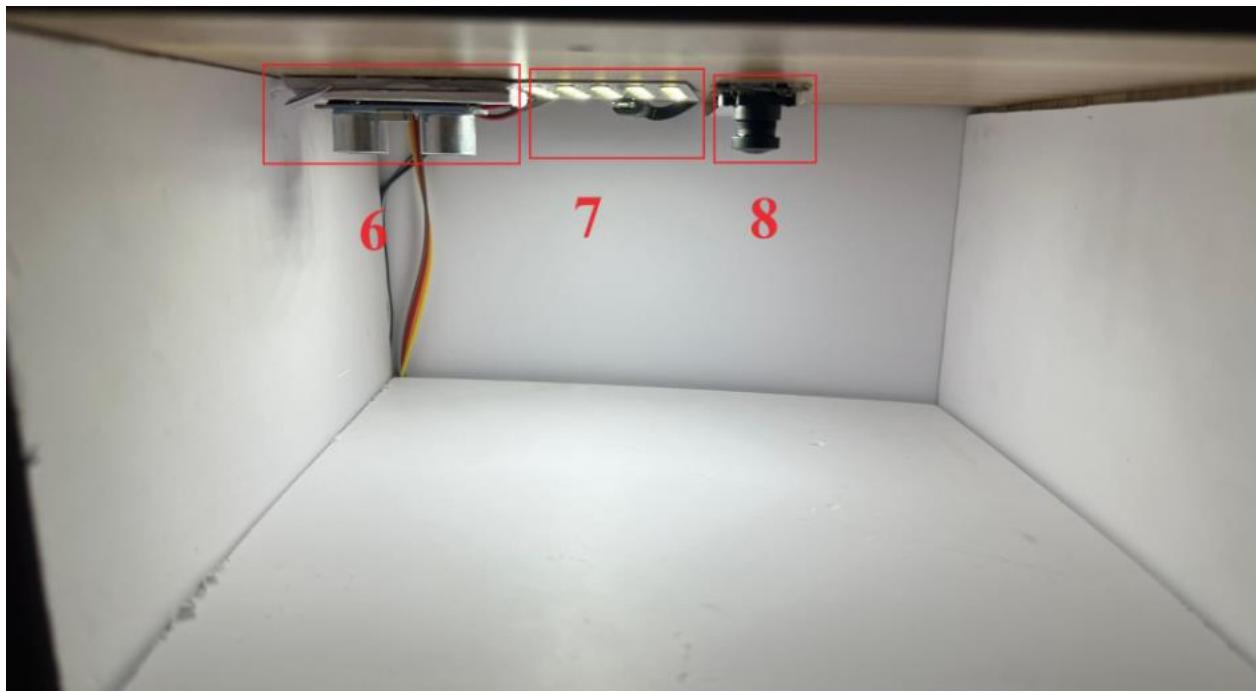
Hệ thống được hoàn thiện với độ chi tiết và chính xác cao. Hệ thống đáp ứng được yêu cầu đặt ra về tính năng lẫn hình dạng. Dưới đây là hình ảnh chi tiết về mô hình đã được hoàn thiện dưới nhiều góc độ chụp khác nhau.



Hình 5. 1. Mô hình chụp tổng quát từ phía trước



Hình 5. 2. Mô hình được chụp từ trên xuống



Hình 5. 3. Mô hình được chụp bên trong không gian nhận diện

Các hình trên mô tả chi tiết về mô hình gồm các thiết bị và cảm biến sau:

- (1) Bàn cân chứa loadcell 1.
- (2) Bàn cân chứa loadcell 2.
- (3) Màn hình hiển thị.
- (4) Màn hình LCD hiển thị cân nặng sản phẩm.
- (5) Nút nhấn và led nút nhấn
- (6) Cảm biến khoảng cách.
- (7) Đèn trợ sáng
- (8) Camera pi

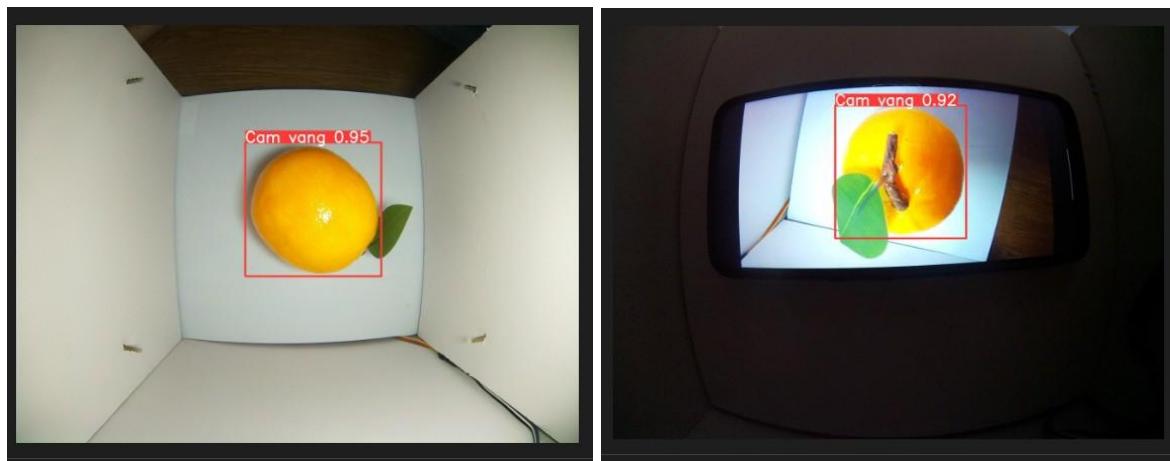
Theo sự mô tả đã được nêu ở trên thì màn hình điện thoại sẽ hiển thị các thông tin của sản phẩm khi camera nhận diện được cho người sử dụng biết. Ba nút nhấn lần lượt là nút nhấn một dùng để chọn chế độ nhận diện trái cây, nút nhấn hai dùng để chọn chế độ nhận diện mã vạch, nút nhấn thứ ba là dùng để gửi sau khi nhận diện ra kết quả từ

nút nhấn một. Camera pi dùng để nhận diện sản phẩm. Bàn cân 1 gồm cảm biến khối lượng 1 và bàn cân 2 gồm cảm biến khối lượng 2 dùng để tính khối lượng vật phẩm.

## 5.2 Kết quả nhận diện trái cây

Sau quá trình thực hiện các bước nhận diện vật thể bằng phương pháp đào tạo hình ảnh, ta có được kết quả nhận diện của từng loại quả như sau:

Ở hình 5.4 và 5.5, đối với cam vàng, thì chương trình có thể nhận diện khá tốt, xác định được gần như toàn bộ diện tích bề mặt quả.



Hình 5. 4. Kết quả nhận diện Cam vàng bằng camera qua cân 1 và cân 2

```
0: 480x640 1 Cam vang, 955.7ms  
Speed: 17.6ms preprocess, 955.7ms inference  
Results saved to runs/detect/predict6  
Label: 0, Confidence: 0.95  
Loadcell value: 0.175
```

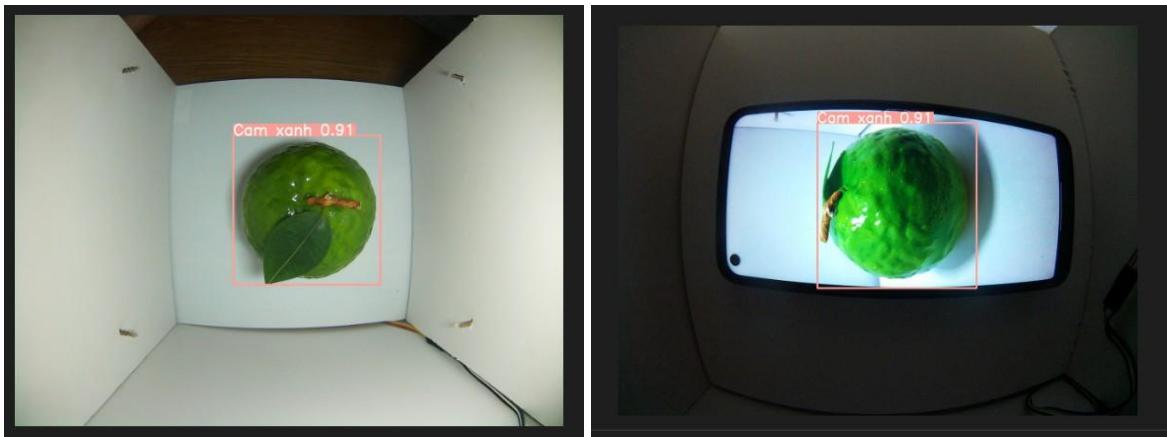
```
0: 480x640 1 Cam vang, 792.4ms  
Speed: 17.0ms preprocess, 792.4ms inference  
Results saved to runs/detect/predict6  
Label: 0, Confidence: 0.92  
Loadcell value: 0.261
```

Hình 5. 5. Kết quả hiển thị nhận diện Cam vàng trên terminal ở cân 1 và cân 2  
Cân 1 là ảnh bên trái, cân 2 là ảnh bên phải, cân 1 là dùng trái cây nhựa, cân 2 dùng ảnh chụp sẵn và mở bằng điện thoại.

Kết quả nhận diện được ở cân 1 là “Cam vang” tốc độ nhận diện dưới 1s, ở lớp thứ 0 và độ chính xác 95%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.175 Kg.

Kết quả nhận diện được ở cân 2 là “Cam vang” tốc độ nhận diện dưới 1s, ở lớp thứ 0 và độ chính xác 92%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.261 Kg.

Ở hình 5.6 và 5.7, đối với cam xanh, thì chương trình có thể nhận diện khá tốt, xác định được gần như toàn bộ diện tích bề mặt quả.



Hình 5. 6. Kết quả nhận diện Cam xanh bằng camera qua cân 1 và cân 2

```
0: 480x640 1 Cam xanh, 936.7ms
Speed: 28.1ms preprocess, 936.7ms inference
Results saved to runs/detect/predict6
Label: 1, Confidence: 0.91
Loadcell value: .177
```

```
0: 480x640 1 Cam xanh, 906.8ms
Speed: 18.7ms preprocess, 906.8ms inference
Results saved to runs/detect/predict6
Label: 1, Confidence: 0.91
Loadcell value: 0.252
```

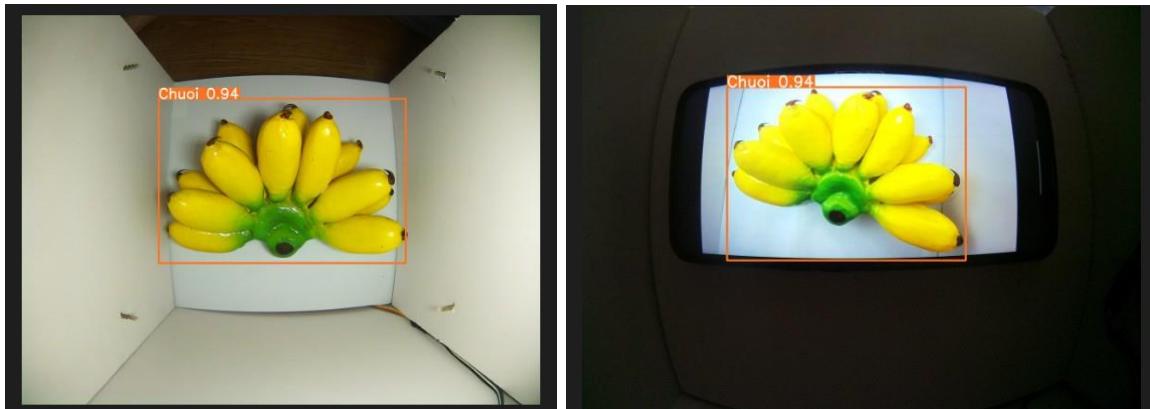
Hình 5. 7. Kết quả hiển thị nhận diện Cam xanh trên terminal ở cân 1 và cân 2

Cân 1 là ảnh bên trái, cân 2 là ảnh bên phải, cân 1 là dùng trái cây nhựa, cân 2 dùng ảnh chụp sẵn và mở bằng điện thoại.

Kết quả nhận diện được ở cân 1 là “Cam xanh” tốc độ nhận diện dưới 1s, ở lớp thứ 1 và độ chính xác 91%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.177 Kg.

Kết quả nhận diện được ở cân 2 là “Cam xanh” tốc độ nhận diện dưới 1s, ở lớp thứ 1 và độ chính xác 91%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.252 Kg.

Ở hình 5.8 và 5.9, đối với chuối, thì chương trình có thể nhận diện khá tốt, xác định được gần như toàn bộ diện tích bề mặt.



Hình 5. 8. Kết quả nhận diện Chuối bằng camera

0: 480x640 1 Chuoi, 1315.8ms Speed: 16.9ms preprocess, 1315.8ms inference Results saved to runs/detect/predict6 Label: 2, Confidence: 0.94 Loadcell value: 0.243	0: 480x640 1 Chuoi, 1185.3ms Speed: 17.7ms preprocess, 1185.3ms inference Results saved to runs/detect/predict6 Label: 2, Confidence: 0.94 Loadcell value: 0.262
--	--

Hình 5. 9. Kết quả hiển thị nhận diện Chuối trên terminal

Cân 1 là ảnh bên trái, cân 2 là ảnh bên phải, cân 1 là dùng trái cây nhựa, cân 2 dùng ảnh chụp sẵn và mở bằng điện thoại.

Kết quả nhận diện được ở cân 1 là “Chuoi” tốc độ nhận diện 1,3s, ở lớp thứ 2 và độ chính xác 94%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.243 Kg.

Kết quả nhận diện được ở cân 2 là “Chuoi” tốc độ nhận diện 1,2s, ở lớp thứ 2 và độ chính xác 94%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.262 Kg.

Ở hình 5.10 và 5.11, đối với mãn cầu, thì chương trình có thể nhận diện khá tốt, xác định được gần như toàn bộ diện tích bề mặt.



Hình 5. 10. Kết quả nhận diện Mãn cầu bằng camera

0: 480x640 1 Man cau, 816.3ms Speed: 16.4ms preprocess, 816.3ms inference Results saved to runs/detect/predict6 Label: 3, Confidence: 0.97 Loadcell value: 0.194	0: 480x640 1 Man cau, 925.7ms Speed: 18.2ms preprocess, 925.7ms inference Results saved to runs/detect/predict6 Label: 3, Confidence: 0.96 Loadcell value: 56
--	---

Hình 5. 11. Kết quả hiển thị nhận diện Mãn cầu trên terminal

Cân 1 là ảnh bên trái, cân 2 là ảnh bên phải, cân 1 là dùng trái cây nhựa, cân 2 dùng ảnh chụp sẵn và mở bằng điện thoại.

Kết quả nhận diện được ở cân 1 là “Man cau” tốc độ nhận diện dưới 1s, ở lớp thứ 3 và độ chính xác 97%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.194 Kg.

Kết quả nhận diện được ở cân 2 là “Man cau” tốc độ nhận diện dưới 1s, ở lớp thứ 3 và độ chính xác 96%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.256 Kg.

Ở hình 5.12 và 5.13, đối với mận cầu, thì chương trình có thể nhận diện khá tốt, xác định được gần như toàn bộ diện tích bề mặt.



Hình 5. 12. Kết quả nhận diện Táo bằng camera

0: 480x640 1 Tao, 891.0ms Speed: 16.2ms preprocess, 891.0ms inference Results saved to runs/detect/predict6 Label: 4, Confidence: 0.94 Loadcell value: 0.181	0: 480x640 1 Tao, 937.6ms Speed: 17.8ms preprocess, 937.6ms inference Results saved to runs/detect/predict6 Label: 4, Confidence: 0.98 Loadcell value: 0.254
--	--

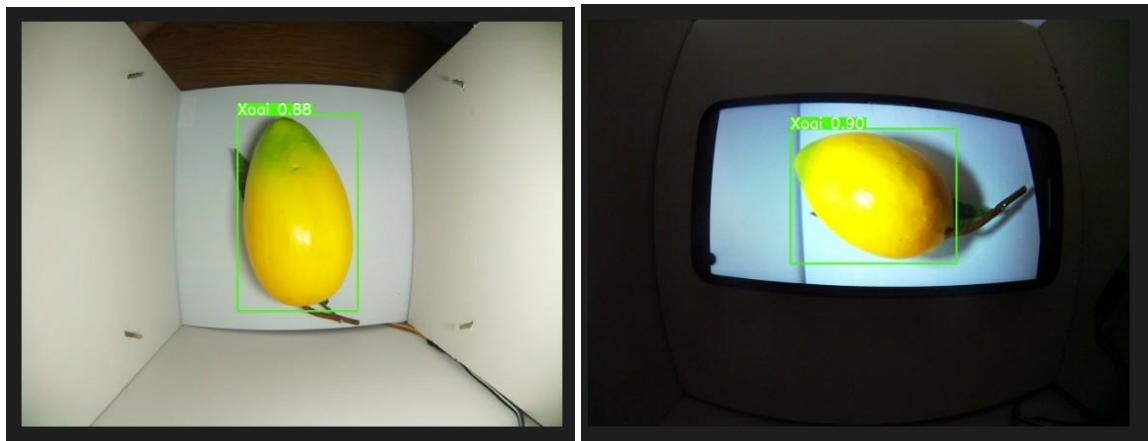
Hình 5. 13. Kết quả hiển thị nhận diện Táo trên terminal

Cân 1 là ảnh bên trái, cân 2 là ảnh bên phải, cân 1 là dùng trái cây nhựa, cân 2 dùng ảnh chụp sẵn và mở bằng điện thoại.

Kết quả nhận diện được ở cân 1 là “Tao” tốc độ nhận diện dưới 1s, ở lớp thứ 4 và độ chính xác 94%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.181 Kg.

Kết quả nhận diện được ở cân 2 là “Tao” tốc độ nhận diện dưới 1s, ở lớp thứ 4 và độ chính xác 98%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.254 Kg.

Ở hình 5.14 và 5.15, đối với xoài, thì chương trình có thể nhận diện khá tốt, xác định được gần như toàn bộ diện tích bề mặt.



Hình 5. 14. Kết quả nhận diện Xoài bằng camera

0: 480x640 1 Xoài, 810.4ms Speed: 17.9ms preprocess, 810.4ms inference Results saved to runs/detect/predict6 Label: 5, Confidence: 0.88 Loadcell value: 0.185	0: 480x640 1 Xoài, 781.0ms Speed: 19.2ms preprocess, 781.0ms inference Results saved to runs/detect/predict6 Label: 5, Confidence: 0.90 Loadcell value: 0.249
---	---

Hình 5. 15. Kết quả hiển thị nhận diện Xoài trên terminal

Cân 1 là ảnh bên trái, cân 2 là ảnh bên phải, cân 1 là dùng trái cây nhựa, cân 2 dùng ảnh chụp sẵn và mở bằng điện thoại.

Kết quả nhận diện được ở cân 1 là “Xoài” tốc độ nhận diện dưới 1s, ở lớp thứ 5 và độ chính xác 88%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.185 Kg.

Kết quả nhận diện được ở cân 2 là “Xoài” tốc độ nhận diện dưới 1s, ở lớp thứ 5 và độ chính xác 90%. Đồng thời ta cũng thấy loadcell hiển thị khối lượng là 0.249 Kg.

Đánh giá độ chính xác khi nhận diện trái cây: Với điều kiện ánh sáng tốt (sử dụng ánh sáng trắng) chiếu thẳng vào sản phẩm, nhìn chung kết quả nhận diện của hệ thống đạt hiệu quả tốt, tỉ lệ chính xác cao.

## 5.2 Kết quả nhận diện mã vạch

Sau đây là kết quả nhận diện mã vạch.



Hình 5. 16. Màn hình nhận diện mã vạch

```
LED 2 ON  
8935006530935  
Button 2 released.
```

Hình 5. 17. Terminal nhận diện mã mạch

Khi chọn chế độ nhận diện mã vạch, camera sẽ nhận diện theo thời gian thực, và chuyển chế độ sang trắng đen để có thể nhận diện tốt, ta có thể mở video camera lên để xem hoặc tắt đi nếu không cần thiết. Hình ảnh mã vạch trên vật phẩm sẽ nhận diện tốt nếu nó được chiếu thẳng vào camera. Khi xác định được mã vạch thì sẽ vẽ khung bao quanh mã vạch đó và hiển thị id của mã vạch. Đồng thời trên terminal sẽ hiển thị được id của vật phẩm sau khi được quét.

## 5.3 Kết quả hoạt động của hệ thống

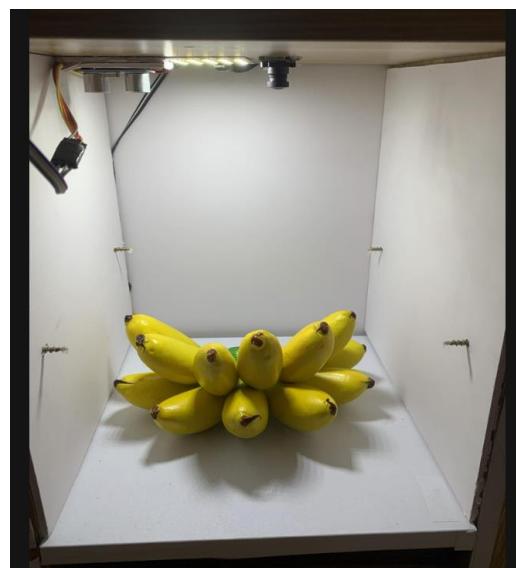
Hệ thống đã được hoàn thiện và đưa ra hoạt động ổn định, khi cấp nguồn hệ thống sẽ được hoạt động.

Khi đưa vật phẩm lên bàn cân 1 hoặc 2 và chọn chế độ nhận diện thì camera sẽ tiến hành quét và nhận diện sản phẩm. Hệ thống hoạt động dựa vào các thuật toán xử lý ảnh, thuật

toán YOLO và các thư viện hỗ trợ để nhận biết sản phẩm. Sau khi có kết quả thì sẽ được gửi đi tới địa chỉ API và theo đó là khối lượng của sản phẩm được ghi nhận khi đặt trên bàn cân.

Đối với chế độ nhận diện trái cây thì sau khi đã nhận diện ta cần nhấn nút gửi để có thể gửi kết quả lên cơ sở dữ liệu của server. Sau khi so sánh dữ liệu với kết quả nhận diện, nếu đồng khớp thì sẽ tiến hành hiển thị thông tin bên trong cơ sở dữ liệu ra màn hình để người dùng nhận biết thông tin sản phẩm, thông tin bao gồm: tên, khối lượng, giá tiền sản phẩm 1kg và giá tiền tương ứng của sản phẩm người dùng.

### Mô tả quá trình nhận diện trái cây



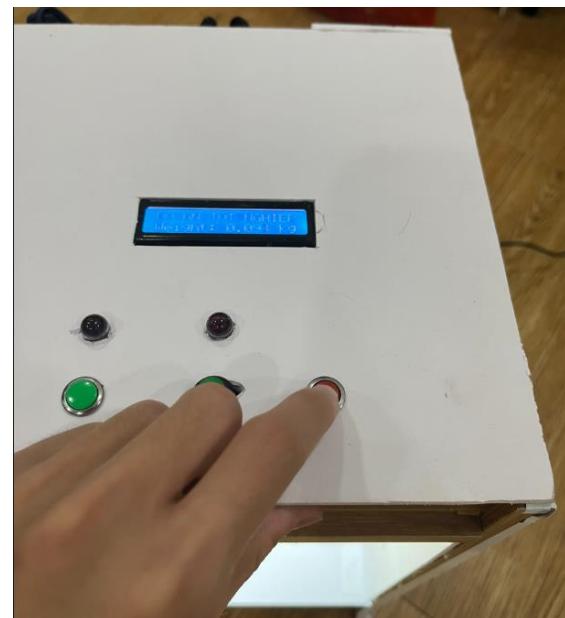
Hình 5. 18. Đưa trái cây vào trong bàn cân



Hình 5. 19. Khối lượng của sản phẩm khi đặt lên bàn cân

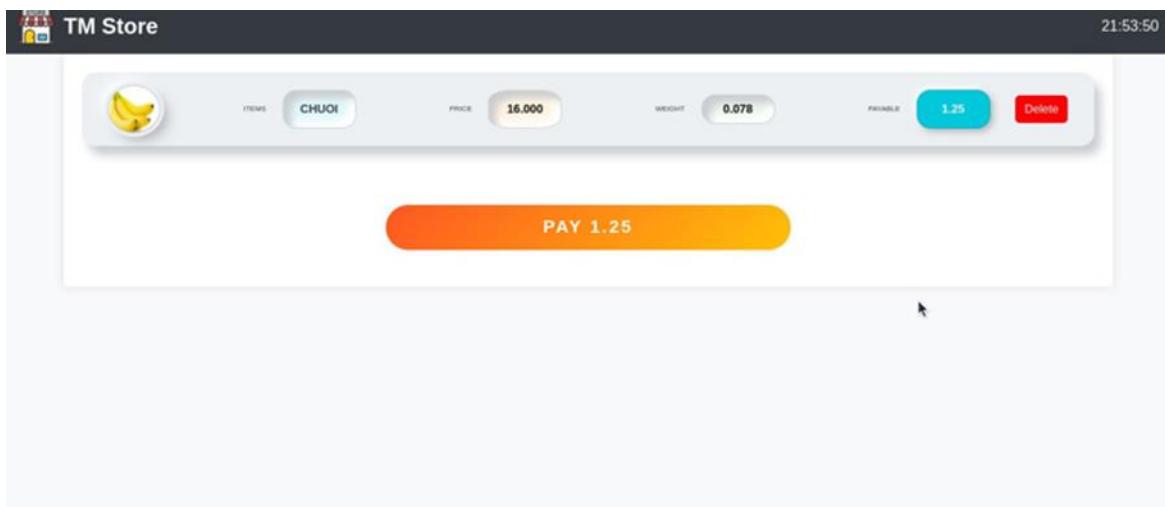


Hình 5. 20. Nhấn nút nhấn 1



Hình 5. 21. Nhấn nút nhấn 3

Sau khi đưa trái cây vào trong bàn cân thì ta nhấn nút 1, camera sẽ chụp ảnh và tiến hành nhận diện, lúc này LCD cũng hiển thị khối lượng của trái cây. Đèn xanh sẽ bật cho đến khi nào mà ảnh đã được nhận diện xong thì đèn sẽ tắt. Sau đó ta nhấn nút 3 để gửi kết quả nhận diện và khối lượng lên server và đưa ra màn hình WEB hình 5.22.



Hình 5. 22. Hiển thị thông tin sản phẩm

Màn hình hiển thị loại trái cây gồm tên, giá theo 1kg và khối lượng thức tế và đưa ra giá trị thực tế. Bên cạnh đó có nút Delete để khi ta có không muốn mua nữa có thể xóa đi và nút PAY để thanh toán hóa đơn trên.

#### **Mô tả quá trình nhận diện mã vạch**

Ta tiến hành nhấn nút 2 để chuyển sang chế độ nhận diện mã vạch. Nhấn lại một lần nữa để trả lại nút nhấn ta sẽ thoát chế độ này, mô tả như hình 5.23.



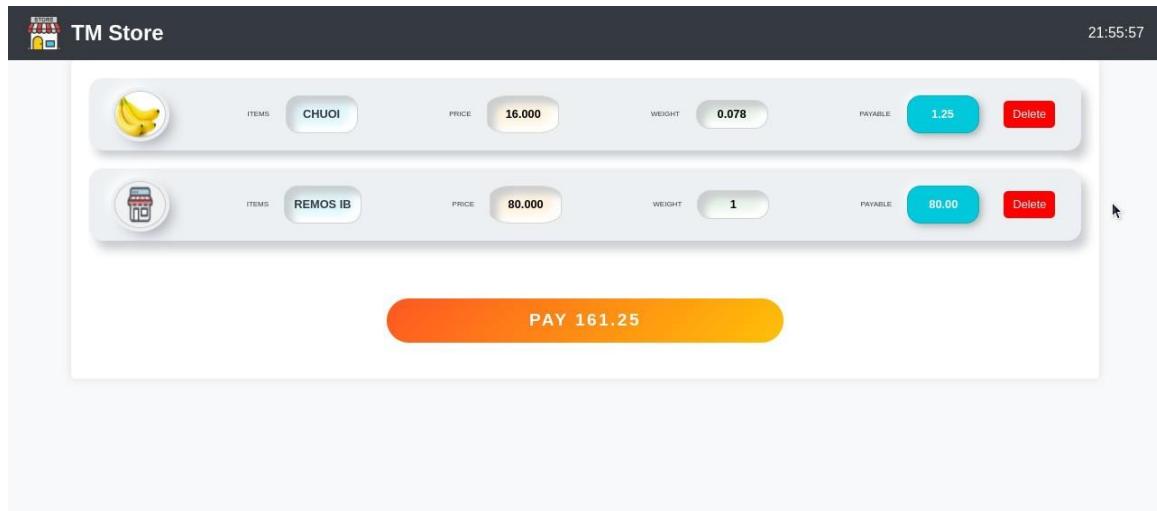
Hình 5. 23. Nhấn nút nhấn 2

Sau đó đưa vật phẩm muốn mua vào và hướng mã vạch lên camera, camera sẽ nhận diện và đưa ra kết quả nhanh chóng, ta quan sát hình 5.24 dưới đây. Nhưng sau 3 giây mới có thể quét tiếp, đó là tính năng để tránh việc quét liên tục và gửi lên liên tục, 3 giây là đủ để nhận diện và gửi lên WEB.



Hình 5. 24. Đưa vật phẩm và quay đầu mã vạch về hướng camera

Sau khi nhận diện và đã gửi lên WEB thì trên WEB sẽ hiển thị thông tin sản phẩm gồm tên, giá thành tương ứng đơn vị số lượng là 1. Ta cũng có thể xóa bằng cách nhấn Delete trên màn hình để xóa nếu không muốn mua nữa. Nếu muốn thanh toán ta nhấn PAY. Đó là mô tả cho hình 5.25 dưới đây.



Hình 5. 25. Hiển thị thông tin sản phẩm

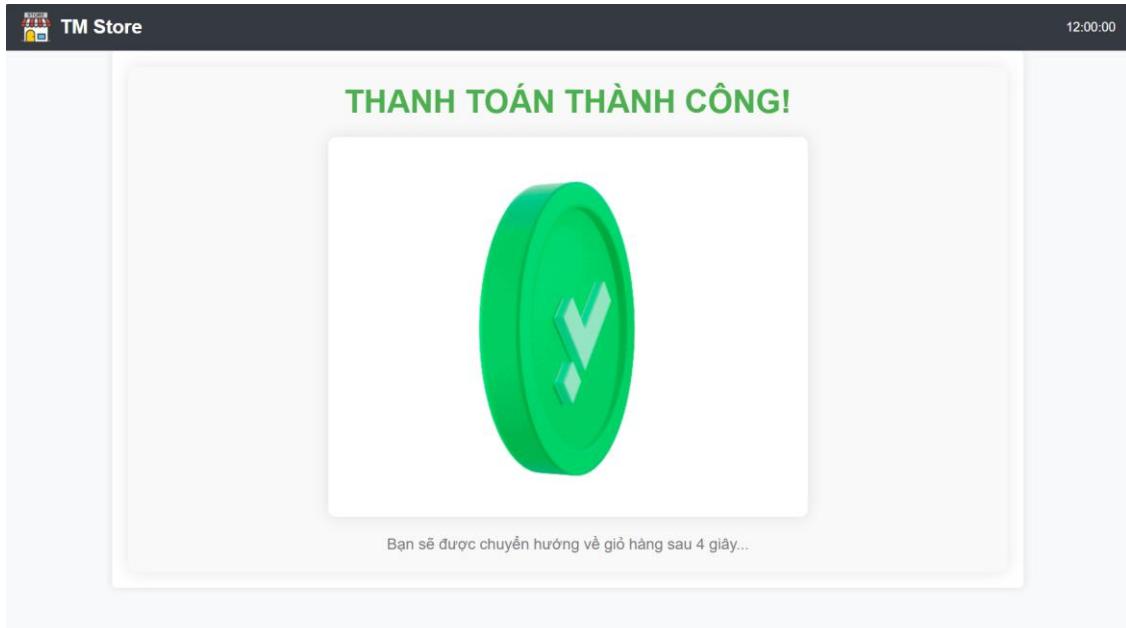
### Thanh toán bằng mã QR

Ở hình 5.26, ta nhấn thanh toán thì sẽ hiện ra mã QR cho người dùng quét thanh toán



Hình 5. 26. Mã QR thanh toán

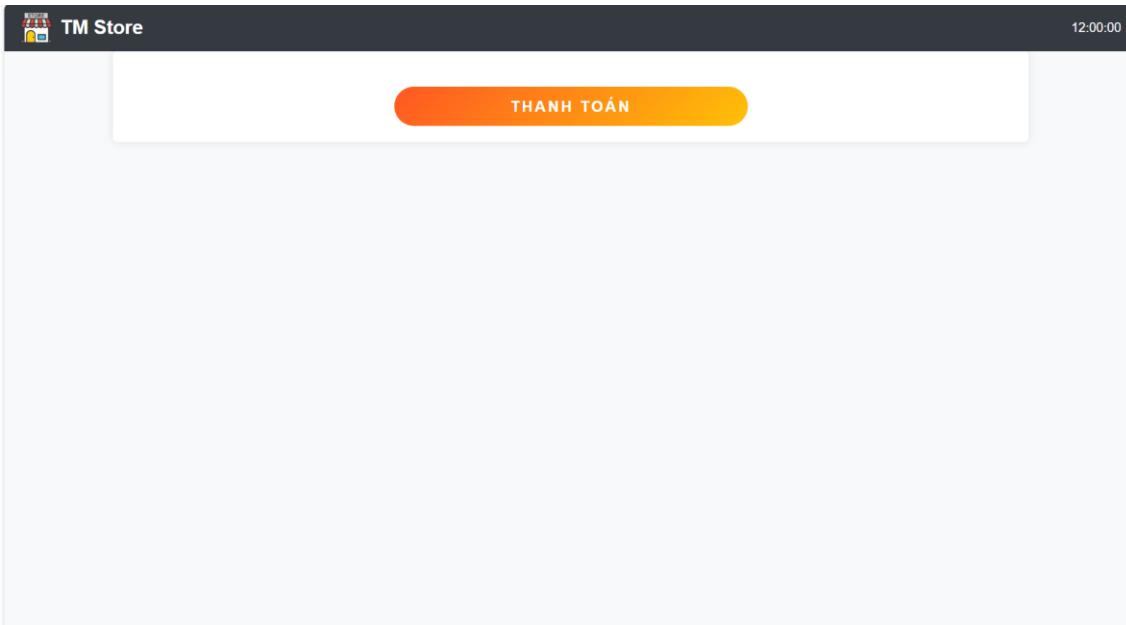
Khi quét mã và thanh toán thành công thì màn hình sẽ hiển thị ra thông tin thanh toán thành công để báo cho người dùng đã hoàn tất thanh toán, nó được mô tả như hình 5.27 sau.



Hình 5. 27. Giao diện hiển thị đã thanh toán

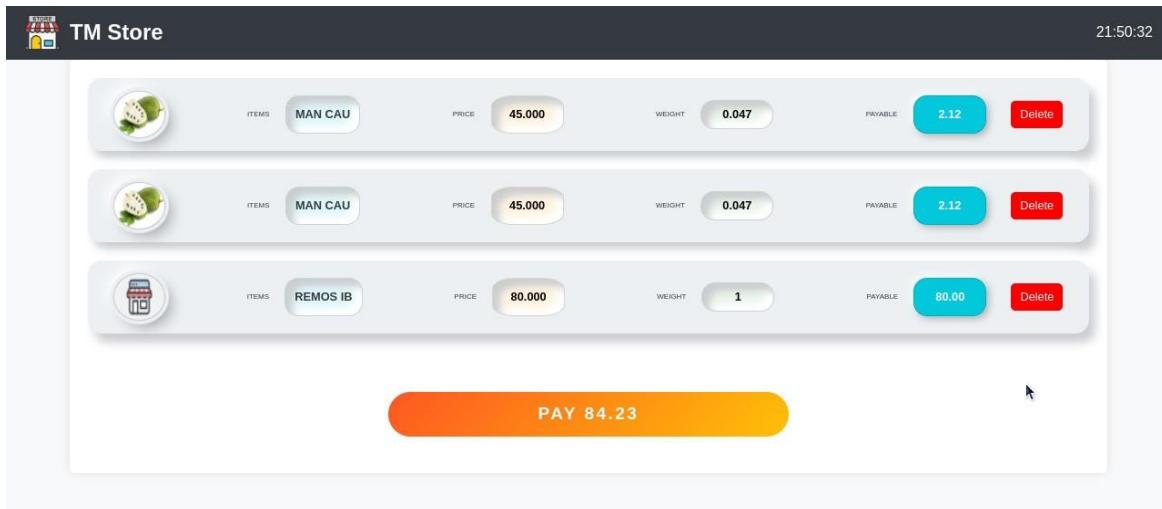
#### 5.4 Giao diện người dùng

Giao diện người dùng minh họa trong hình 5.28 thể hiện một thiết kế tối giản nhằm tạo điều kiện thuận lợi cho việc sử dụng và giảm thiểu sự lộn xộn thông tin mà khách hàng cần. Điều này giúp mọi người, từ trẻ nhỏ đến người cao tuổi, đều có thể dễ dàng thao tác trên màn hình.



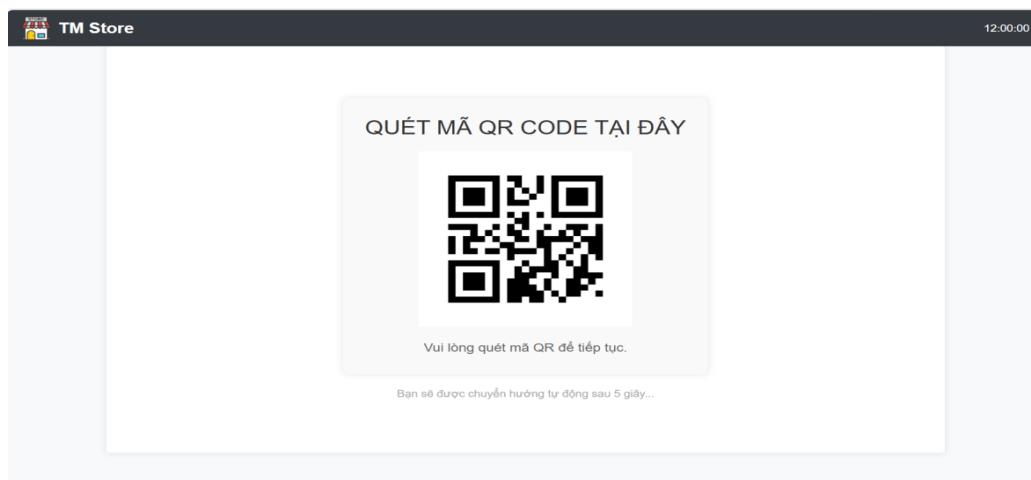
Hình 5. 28. Giao diện người dùng.

Về phần giao diện hiển thị thông tin sản phẩm được minh họa trong hình 5.29, đây là những thông tin thiết yếu mà hầu hết người dùng đều cần biết. Giao diện này nhằm mục đích hỗ trợ việc quản lý số lượng và các sản phẩm mà người dùng.



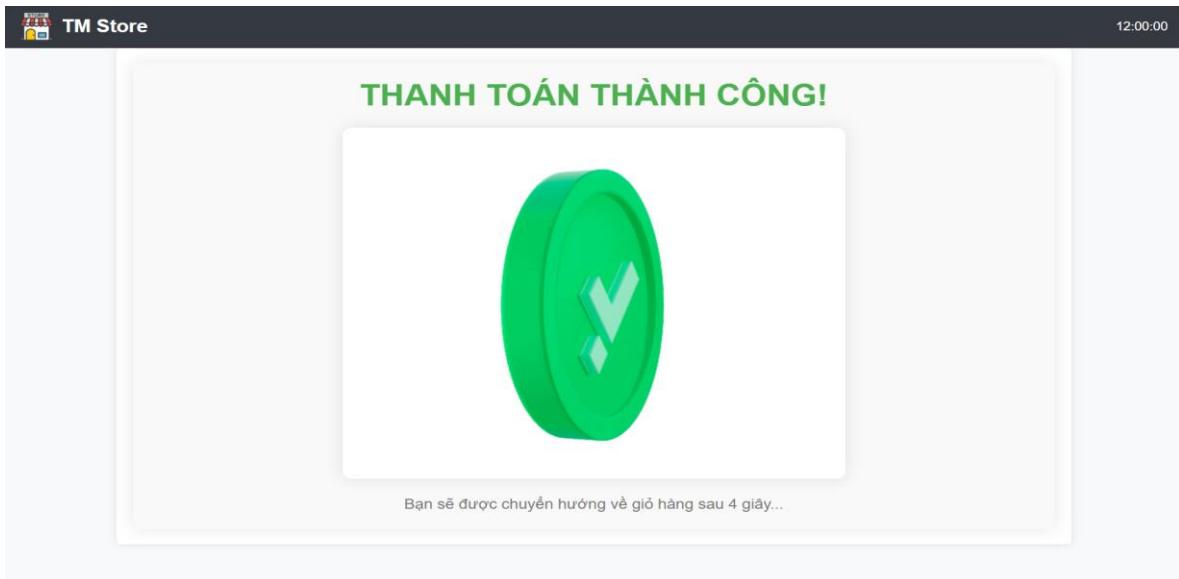
Hình 5. 29. Giao diện hiển thị thông tin sản phẩm

Trong hình 5.30, mã QR được hiển thị đơn giản và không gây khó chịu cho mắt người dùng. Điều này giúp người dùng có thể quét mã một cách nhanh chóng, chính xác và an toàn hơn. Hiện nay, các ứng dụng quét mã QR đã rất nhạy, chỉ cần người dùng di chuyển qua mã QR là thiết bị quét có thể phát hiện và nhận diện ngay. Việc giảm thiểu phức tạp trong thiết kế trang thanh toán cũng giúp khách hàng yên tâm hơn về việc không nhầm lẫn giữa các mã QR khác nhau.



Hình 5. 30. Giao diện web cho người dùng thanh toán.

Sau khi thanh toán hoàn tất, giao diện sẽ hiển thị thanh toán thành công như hình 5.31 dưới đây, báo cho người dùng biết đã hoàn tất thanh toán và đồng thời sẽ quay lại giao diện người dùng mới sau 4 giây.



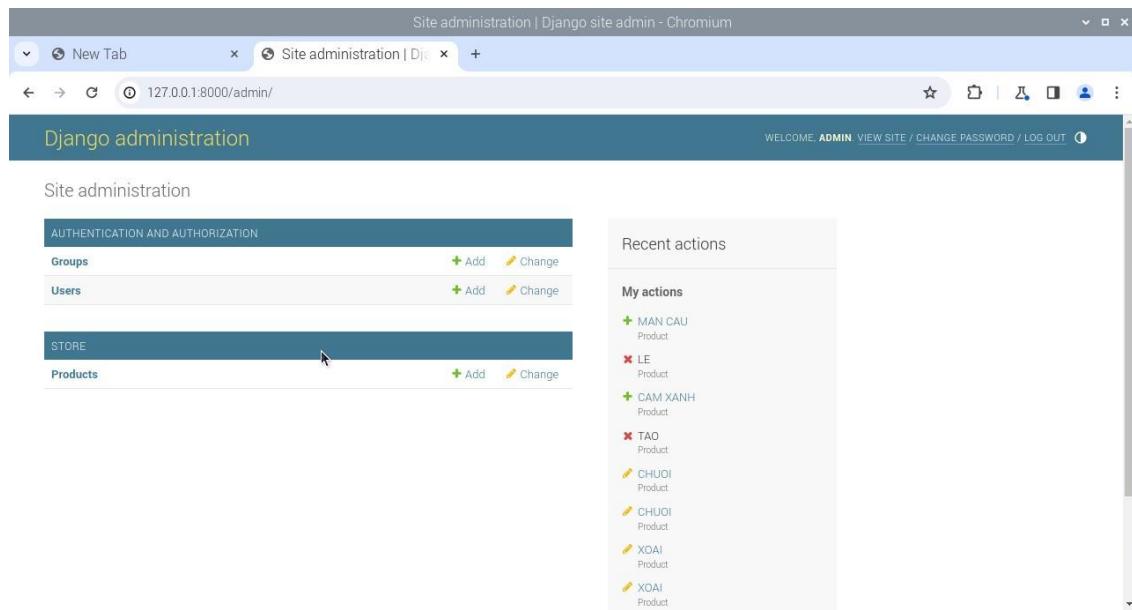
Hình 5. 31. Giao diện web khi người dùng thanh toán thành công.

Giao diện người dùng được thiết kế đơn giản, dễ hiểu và dễ sử dụng tạo sự thuận tiện cho khách hàng khi thao tác.

#### 5.4 Giao diện cho ADMIN

Trong giao diện ADMIN như hình 5.32, người quản trị có thể:

- Quản lý thông tin sản phẩm.
- Chính sửa thông tin sản phẩm.



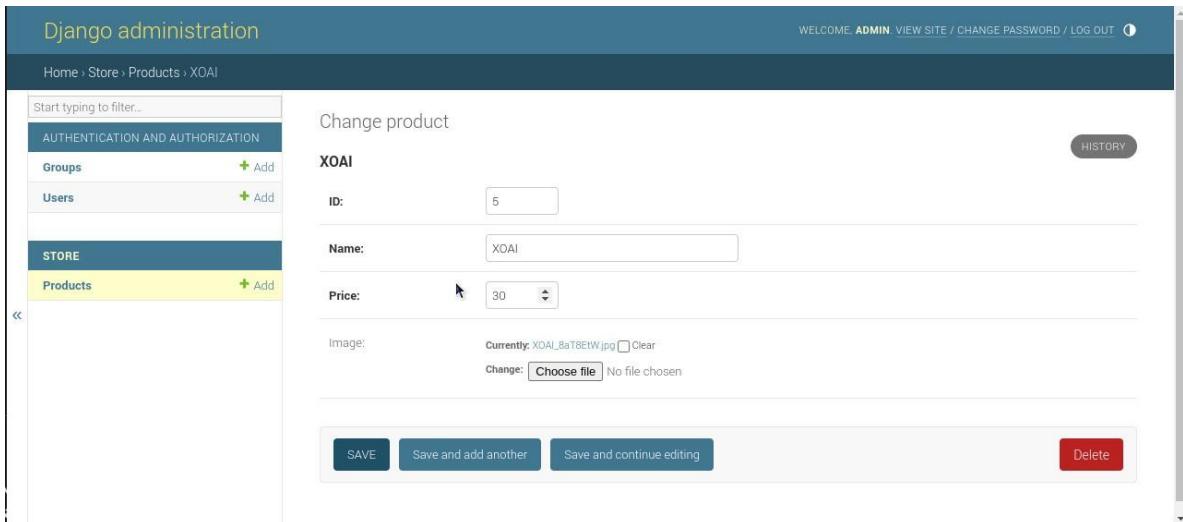
Hình 5. 32. Giao diện ADMIN

Dưới đây là hình ảnh biểu thị tất cả sản phẩm.

This screenshot shows the 'Select product to change' page for the 'Products' model. The left sidebar has a 'Start typing to filter...' search bar and lists 'Groups', 'Users', and 'Products'. The 'Products' entry is highlighted. The main area shows a list of products with checkboxes: 'PRODUCT', 'Remos IB', 'XOAI', 'TAO', 'MAN CAU' (which is selected), 'CHUOI', 'CAM XANH', and 'CAM VANG'. A total of '7 products' are listed. At the top right is an 'ADD PRODUCT' button.

Hình 5. 33. Giao diện quản lý sản phẩm

Hình 5.34 là khi quản trị viên nhập thông tin cho sản phẩm. Tại đây cho phép điều chỉnh về ID, tên và giá sản phẩm.



Hình 5. 34. Giao diện thông tin sản phẩm

Nhìn chung phần giao diện ADMIN có mục đích chính là điều chỉnh các sản phẩm có trong hệ thống, việc thêm hay xóa, thay đổi dễ dàng, nhanh chóng.

### 5.5 Terminal địa chỉ IP gửi server

Trên hình 5.35, khi địa chỉ API gửi lệnh yêu cầu POST tới server với thông tin về sản phẩm từ camera và giá trị khối lượng từ cảm biến, hệ thống sẽ thực hiện các công việc sau:

- Truy xuất dữ liệu từ cơ sở dữ liệu: Server sẽ lấy thông tin chi tiết của các sản phẩm từ cơ sở dữ liệu dựa trên các thông tin nhận diện được từ camera và giá trị khối lượng gửi lên.
- So sánh và xử lý dữ liệu: Sau khi lấy thông tin từ cơ sở dữ liệu, server sẽ so sánh với các thông tin mới nhận được từ API. Điều này giúp xác nhận chính xác thông tin về sản phẩm và khối lượng.
- Hiển thị thông tin trên terminal: Kết quả của quá trình xử lý sẽ được hiển thị trên terminal, cho phép quản trị viên xem và kiểm tra các thông tin sản phẩm và khối lượng đã gửi lên từ API.
- Hiển thị trên giao diện người dùng: Đồng thời, các thông tin quan trọng về sản phẩm và dữ liệu thanh toán cũng được hiển thị trên giao diện người dùng, giúp người dùng có thể xem lại và xác nhận các thông tin liên quan đến đơn hàng.

của mình.

Quá trình này giúp tăng tính chính xác và đáp ứng nhanh chóng các yêu cầu liên quan đến quản lý hàng hóa và thanh toán, đồng thời cung cấp thông tin chi tiết và minh bạch cho cả người dùng và quản trị viên.

```
[16/Jun/2024 22:13:25] "GET /api?id=None&loadcellValue=1 HTTP/1.1" 500 124525
8935006530935
1
<QuerySet [<Product: Remos IB>]>
[{'id': '8935006530935', 'image': <ImageFieldFile: png-transparent-computer-icons-online-shopping-others_rlu33vJ
ng>, 'name': 'Remos IB', 'price': 80, 'loadcell_value': '1', 'total': 80.0}]
[16/Jun/2024 22:13:25] "GET /api?id=8935006530935&loadcellValue=1 HTTP/1.1" 200 2125
[16/Jun/2024 22:13:26] "GET / HTTP/1.1" 200 2125
```

Hình 5. 35. Terminal hiển thị thông tin sản phẩm

## 5.6 Nhận xét, đánh giá hệ thống

Hệ thống phần cứng hoạt động ổn định và có tốc độ tính toán và lấy giá trị từ loadcell nhanh và ổn định, đáp ứng tốt các yêu cầu ban đầu.

Tuy nhiên, trong quá trình vận hành phần mềm, đôi khi gặp phải tình trạng đường truyền mạng bị nghẽn, dẫn đến chậm trong việc truyền và nhận dữ liệu. Màn hình hiển thị thông tin cũng đôi khi không ổn định. Dữ liệu được đẩy lên server liên tục và sẵn sàng để cập nhật.

Giao diện người dùng và giao diện ADMIN được thiết kế đơn giản, dễ sử dụng và quản lý, giúp cải thiện trải nghiệm người dùng và hiệu quả quản lý hệ thống.

Tóm lại, mặc dù phần cứng hoạt động tốt, nhưng phần mềm cần xử lý các vấn đề về đường truyền mạng và ổn định hiển thị thông tin để đảm bảo sự liên tục và chính xác trong quản lý dữ liệu và giao tiếp.

## CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Hệ thống máy tính toán tự động là một bước tiến quan trọng trong việc nâng cao hiệu quả và trải nghiệm mua sắm cho người dùng tại các cửa hàng và siêu thị. Dưới đây là một số điểm nổi bật của hệ thống này:

- Sử dụng cảm biến khói lượng và camera nhận diện sản phẩm:
  - + Hệ thống sử dụng cảm biến khói lượng để đo chính xác cân nặng của sản phẩm, giúp người dùng nhanh chóng biết được thông tin này mà không cần thao tác thêm.
  - + Camera tích hợp giúp nhận diện sản phẩm tự động mà không cần nhập mã hoặc quét mã vạch, mang lại sự tiện lợi và giảm thiểu sai sót.
- Cập nhật thông tin sản phẩm liên tục và nhanh chóng:
  - + Hệ thống hiển thị thông tin sản phẩm được thiết kế để cập nhật liên tục, đảm bảo rằng người dùng luôn có thông tin mới nhất về sản phẩm, giá cả và các chương trình khuyến mãi.
- Giao diện người dùng thân thiện:
  - + Giao diện của hệ thống được thiết kế đơn giản nhưng trực quan, dễ nhìn và dễ sử dụng. Điều này giúp mọi đối tượng người dùng, từ trẻ em đến người lớn tuổi, đều có thể thao tác một cách dễ dàng.
  - + Dù giao diện đơn giản, nhưng vẫn cung cấp đầy đủ các thông tin cần thiết mà người dùng mong muốn.
- Thanh toán bằng mã QR:
  - + Hệ thống hỗ trợ chức năng thanh toán bằng mã QR, giúp hạn chế việc sử dụng tiền mặt, tăng cường tính an toàn và tiện lợi trong giao dịch.
- Khả năng mở rộng cơ sở dữ liệu sản phẩm:
  - + Hệ thống có khả năng thêm mới các sản phẩm không có sẵn trong cơ sở dữ liệu, cho phép camera nhận diện nhiều loại sản phẩm hơn, đa dạng hóa danh mục hàng hóa và đáp ứng nhu cầu của người dùng tốt hơn.
- Hiệu quả và tiết kiệm nhân lực:

- + Việc áp dụng hệ thống máy thanh toán tự động giúp các cửa hàng và siêu thị tiết kiệm nhân lực, giảm bớt số lượng nhân viên cần thiết cho việc thanh toán thủ công.
- + Quản lý cửa hàng cũng trở nên dễ dàng hơn nhờ vào việc sử dụng các thiết bị hiện đại và công nghệ tiên tiến, phù hợp với xu hướng phát triển của thời đại công nghệ 4.0.

Lợi ích của hệ thống máy thanh toán tự động:

- Nâng cao trải nghiệm người dùng: Giúp người mua hàng tiết kiệm thời gian, giảm bớt các thao tác không cần thiết.
- Tăng cường hiệu suất hoạt động: Giảm thiểu sai sót trong quá trình thanh toán và quản lý sản phẩm.
- Tiết kiệm chi phí: Giảm bớt số lượng nhân viên cần thiết, từ đó giảm chi phí vận hành.
- Đáp ứng nhu cầu thị trường: Hệ thống có khả năng mở rộng, cập nhật nhanh chóng để phù hợp với nhu cầu của thị trường và người tiêu dùng.

Kết luận:

Hệ thống máy thanh toán tự động là một giải pháp toàn diện và hiện đại, giúp cải thiện hiệu quả hoạt động của các cửa hàng, siêu thị, đồng thời mang lại sự tiện lợi và hài lòng cho người tiêu dùng. Việc áp dụng công nghệ này không chỉ giúp tiết kiệm nhân lực mà còn hỗ trợ quản lý cửa hàng một cách hiệu quả hơn trong thời kỳ công nghệ 4.0.

## **6.2 Hạn chế và hướng phát triển**

### **6.2.1 Hạn chế**

Dưới đây là một số hạn chế của đề tài cần lưu ý:

- Đường truyền mạng không ổn định: Đường truyền mạng đôi khi không ổn định, gây ảnh hưởng đến việc hiển thị thông tin sản phẩm và quá trình truyền và nhận dữ liệu. Sự gián đoạn này có thể làm giảm trải nghiệm của người dùng khi hệ thống hoạt động không liên tục và mượt mà.

- Thiếu chức năng in hóa đơn: Hệ thống hiện tại chưa hỗ trợ in hóa đơn cho khách hàng, đây là một điểm trừ khi người dùng cần bằng chứng mua hàng hoặc quản lý tài chính cá nhân.

Mặc dù hệ thống máy tính toán tự động mang lại nhiều lợi ích vượt trội, nhưng vẫn còn một số hạn chế cần khắc phục để nâng cao trải nghiệm người dùng. Bằng cách cải tiến những điểm yếu này, hệ thống sẽ trở nên hoàn thiện hơn, đáp ứng tốt hơn nhu cầu của khách hàng và tăng cường hiệu quả hoạt động của cửa hàng, siêu thị.

### 6.2.2 Hướng phát triển

- Nâng cao khả năng nhận diện của camera:

Cải tiến thuật toán nhận diện hình ảnh để tăng độ chính xác và tốc độ nhận diện.

Sử dụng các camera chất lượng cao hơn và công nghệ xử lý hình ảnh tiên tiến.

- Cải thiện cảm biến khói lượng:

Sử dụng các cảm biến khói lượng có độ chính xác cao hơn và khả năng chống nhiễu tốt hơn.

Thực hiện kiểm tra và hiệu chỉnh định kỳ để đảm bảo sai số được giảm thiểu.

- Ôn định đường truyền mạng:

Sử dụng các giải pháp mạng dự phòng hoặc kết nối ôn định hơn để đảm bảo hệ thống hoạt động liên tục.

Tối ưu hóa quy trình truyền dữ liệu để giảm thiểu tác động của sự cố mạng.

- Bổ sung chức năng in hóa đơn:

Tích hợp máy in vào hệ thống để cung cấp hóa đơn giấy cho khách hàng.

Cung cấp tùy chọn gửi hóa đơn qua email hoặc tin nhắn để tiện lợi và thân thiện với môi trường.

## TÀI LIỆU THAM KHẢO

### Sách tham khảo:

- [1] Nguyễn Thanh Hải, “Giáo trình Xử Lý Ảnh”, Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh, 2014.
- [2] Nguyễn Quang Hoan, “Xử Lý Ảnh”, Học viện Công nghệ Bưu chính Viễn thông, 2006.
- [3] Adrian Holovaty, Jacob K. Moss, “The Django Book”
- [4] Daniel Rubio, “Beginning Django”
- [5] Ultralytics, "YOLOv8: Trainable Bag-of-Freebies for Object Detection", 2022.
- [6] Glenn Jocher, "YOLOv8", 2022.

### Datasheet tham khảo:

- [7] Datasheet Raspberry Pi  
4:<https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>
- [8] Datasheet RPiCamMod2:  
<https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/RPiCamMod2.pdf>
- [9] Datasheet HX711: <https://s.net.vn/H7Hh>
- [10] Datasheet ESP32: <https://s.net.vn/kBqg>
- [11] Datasheet LCD 16x2: <https://www.vishay.com/docs/37299/37299.pdf>
- [12] Datasheet I2C: <https://s.net.vn/XFWZ>
- [13] Datasheet Button: <https://s.net.vn/uAYN>
- [14] Loadcell: <https://hshop.vn/products/cam-bien-loadcell-5kg>
- [15] HCSR04: <https://hshop.vn/products/cam-bien-sieu-am-srf04>
- [16] Led: <https://icdayroi.com/led-sieu-sang-mau-xanh-duong-10mm>
- [17] Buzzer: <https://icdayroi.com/buzzer-12095-5v>

### Bài viết tham khảo:

- [18] Phạm Thị Thanh Thảo – Phan Trần Hoài Vũ, Đồ án tốt nghiệp “Nghiên cứu, thiết kế và thi công mô hình đếm và phân loại sản phẩm theo cân nặng, màu sắc”, TPHCM, 6/2019: <https://s.net.vn/LDzV>

- [19] Encord, "YOLO Object Detection Guide," Encord Blog, 2023. [Online]. Available: <https://encord.com/blog/yolo-object-detection-guide/>
- [20] HackMD, "Viewperm," HackMD, 2024. [Online]. Available: <https://hackmd.io/@58ZC49ZfS86wYX--LRGGOg/Viewperm>
- [21] Wikipedia, "OpenCV," Wikipedia, 2024. [Online]. Available: <https://en.wikipedia.org/wiki/OpenCV>.
- [22] Wikipedia, "Pyzbar," Wikipedia, 2024. [Online]. Available: <https://en.wikipedia.org/wiki/Pyzbar>.
- [23] Nguyễn Hiền Minh – Phan Thanh Phong, Đồ án tốt nghiệp “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm”, TPHCM, 6/2019: <https://s.net.vn/q1d3>
- [23] SPI vs. I2C vs. UART: Differences Between These Communication Interfaces: <https://www.nextpcb.com/blog/spi-i2c-uart>

## **PHỤ LỤC 1**

Hướng dẫn sử dụng toàn hệ thống:

- Bước 1: Đặt sản phẩm lên giữa bàn cân.
- Bước 2: Nhấn nút chọn chế độ:
  - Nhấn nút 1 để nhận diện trái cây.
  - Nhấn nút 2 để nhận diện mã vạch.
- Bước 3: Khi đèn nút 1 tắt. Nhấn nút 3 để gửi. Nếu hệ thống kêu lên 1 tiếng “Bíp” thì là hệ thống đã gửi thành công. Còn đối với mã vạch thì không cần nhấn nút gửi sau khi nhận diện thành công hệ thống sẽ tự gửi và cũng sẽ kêu lên tương tự như nhận diện trái cây.
- Bước 4: Lấy sản phẩm đã nhận diện ra khỏi vị trí bàn cân và tiếp tục thực hiện bước 1, 2 và 3 cho các loại sản phẩm còn lại.
- Bước 5: Nhấn nút “PAY” ngay trên màn hình bên dưới thông tin sản phẩm để tiến hành thanh toán.
- Bước 6: Quét mã QR để thanh toán.

**Chú Ý:** Sau khi quét mã và thanh toán thành công thì giao diện sẽ hiển thị là đã thanh toán thành công và trở về lại về giỏ hàng trống ban đầu.

## **PHỤ LỤC 2**

Video vận hành hệ thống:

[https://drive.google.com/drive/folders/1uptKhCPKP\\_fQeu1FcUO402RKuV3TiQUI?usp=sharing](https://drive.google.com/drive/folders/1uptKhCPKP_fQeu1FcUO402RKuV3TiQUI?usp=sharing)

Code WEB:

[https://drive.google.com/drive/folders/1UTFXhoKJKaKshAYi7nX1Yd3EU\\_ACWujp?usp=sharing](https://drive.google.com/drive/folders/1UTFXhoKJKaKshAYi7nX1Yd3EU_ACWujp?usp=sharing)

Code VXL:

[https://drive.google.com/drive/folders/1zaUdH4gupbAfd3\\_NhWUEaTz8n24v9W2S?usp=sharing](https://drive.google.com/drive/folders/1zaUdH4gupbAfd3_NhWUEaTz8n24v9W2S?usp=sharing)