



REPORT MIDTERM

AUTOMATIC FRUIT SORTING CONVEYOR SYSTEM

Group 6

Ngày 17 tháng 10 năm 2024



Sinh viên thực hiện 106200284 - Hồ Đức Vũ - 20KTMT2 106200241 - Nguyễn Minh Phương - 20KTMT1 106200242 - Nguyễn Văn Vĩnh Quang - 20KTMT1 106200271 - Lê Tuấn Nhật - 20KTMT2	
Giáo viên hướng dẫn Ths. Hồ Viết Việt	
Môn học PBL 5: Embedded system & IoT	
Đề tài AUTOMATIC FRUIT SORTING CONVEYOR SYSTEM	
Xuất bản Đà Nẵng, Ngày 17 tháng 10 năm 2024	Số trang 10

Mục lục

Nội dung báo cáo	3
1 Giới thiệu	3
2 Nguyên lý hoạt động của hệ thống	3
3 Cấu trúc Server	6
4 Sơ đồ khối phần cứng	8
5 Kết luận	9
Phụ lục	10

Nội dung báo cáo

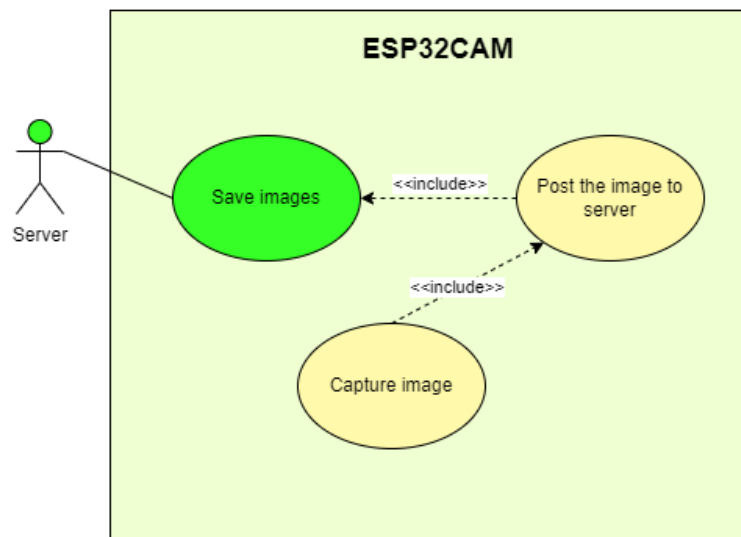
1 Giới thiệu

Ngày nay, việc tự động hóa máy móc trong sản xuất và đóng gói sản phẩm đã trở nên phổ biến trong các nhà máy, xí nghiệp. Cùng với đó là sự phát triển thần tốc của công nghệ *Internet of Things (IoT)* và *Trí tuệ nhân tạo* càng thúc đẩy sự phát triển mạnh mẽ nền công nghiệp sản xuất tự động, nơi mà hệ thống máy móc có thể hoạt động một cách độc lập và tương tác với nhau mà không hoặc ít cần sự can thiệp của con người.

Dựa trên xu hướng đó, nhóm đã lựa chọn đề tài *Automatic fruit sorting conveyor system* với mong muốn áp dụng những kiến thức, kỹ năng mà bản thân đã được và đang học, cùng với đó là bắt kịp xu hướng về tự động hóa, IoT và trí tuệ nhân tạo.

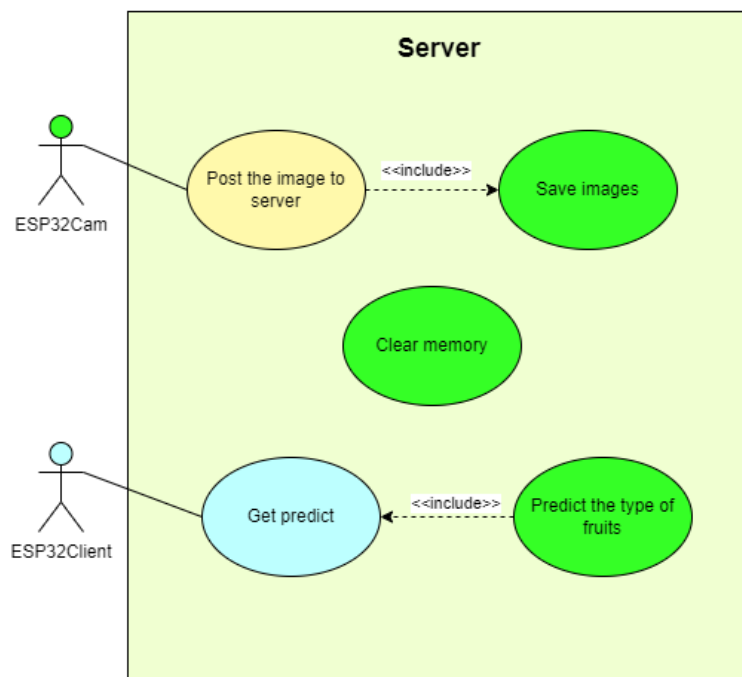
Đề tài *Automatic fruit sorting conveyor system* được thực hiện với mục đích phân loại trái cây một cách tự động trên băng chuyền. Với đầu vào của hệ thống là các loại trái cây cần phân loại, đầu ra của hệ thống là các hộp trái cây với mỗi hộp chứa 1 loại trái cây. Hệ thống sử dụng camera từ esp32cam để thu thập hình ảnh trái cây trên băng chuyền, sau đó hình ảnh sẽ được xử lý và đưa ra dự đoán về loại trái cây, quá trình xử lý sẽ được Server đảm nhiệm, sau đó kết quả được gửi về một bộ kit khác mà nhóm sử dụng là esp32 để thực hiện phân loại trái cây vào hộp tương ứng. Nguyên lý hoạt động của hệ thống được thể hiện chi tiết tại §2.

2 Nguyên lý hoạt động của hệ thống

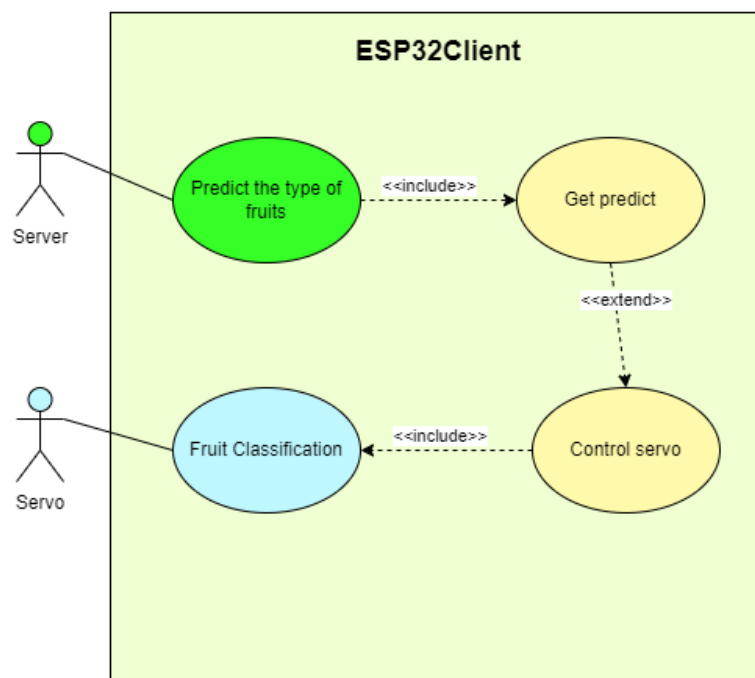


Hình 2.1: Use case diagram thể hiện chức năng của ESP32Cam

Hình 2.1, 2.2 và 2.3 là các Use case diagram lần lượt thể hiện các chức của từng khối trong hệ thống.



Hình 2.2: Use case diagram thể hiện chức năng của Server



Hình 2.3: Use case diagram thể hiện chức năng của ESP32Client

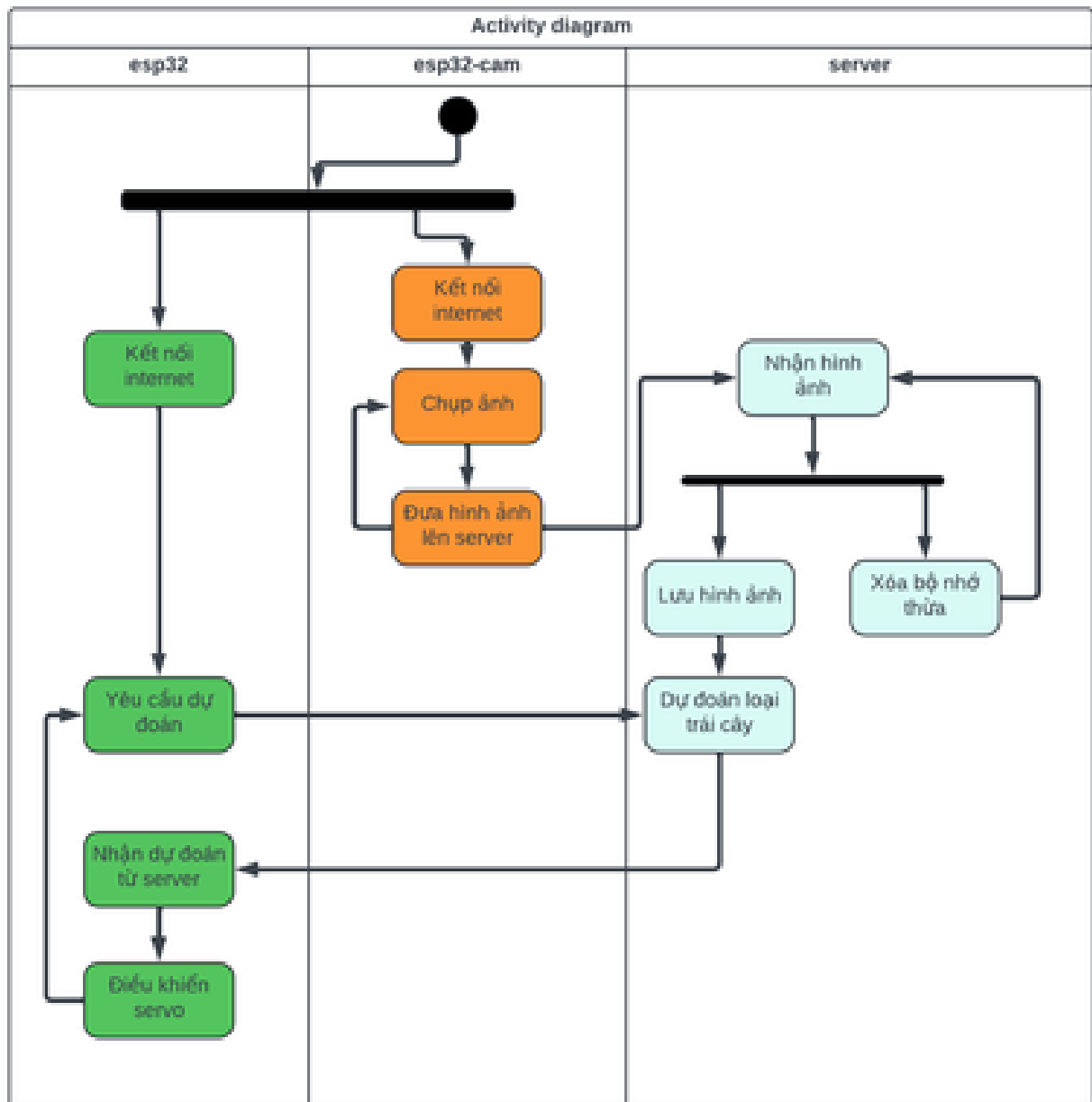
Khối đầu tiên là ESP32Cam, khối này thể hiện cho board ESP32Cam, với nhiệm vụ thu thập hình ảnh và gửi hình ảnh đó lên Server, đây là khối nhận tín hiệu đầu vào của hệ thống.

Khối tiếp theo là Server, khối này được xây dựng với mục đích lưu trữ hình ảnh thu được

từ ESP32Cam, sử dụng model trí tuệ nhân tạo đã được huấn luyện trước đó để dự đoán hình ảnh thuộc loại trái cây gì và dọn bộ nhớ tránh trường hợp Server bị tràn bộ nhớ do việc lưu trữ hình ảnh chiếm nhiều không gian bộ nhớ.

Khối cuối cùng là ESP32Client, khối này sử dụng ESP32 để nhận lấy dự đoán loại trái cây từ Server và điều khiển servo để phân loại trái cây dựa trên kết quả dự đoán.

Hình 2.4 thể hiện Activity Diagram mô tả luồng hoạt động của hệ thống.



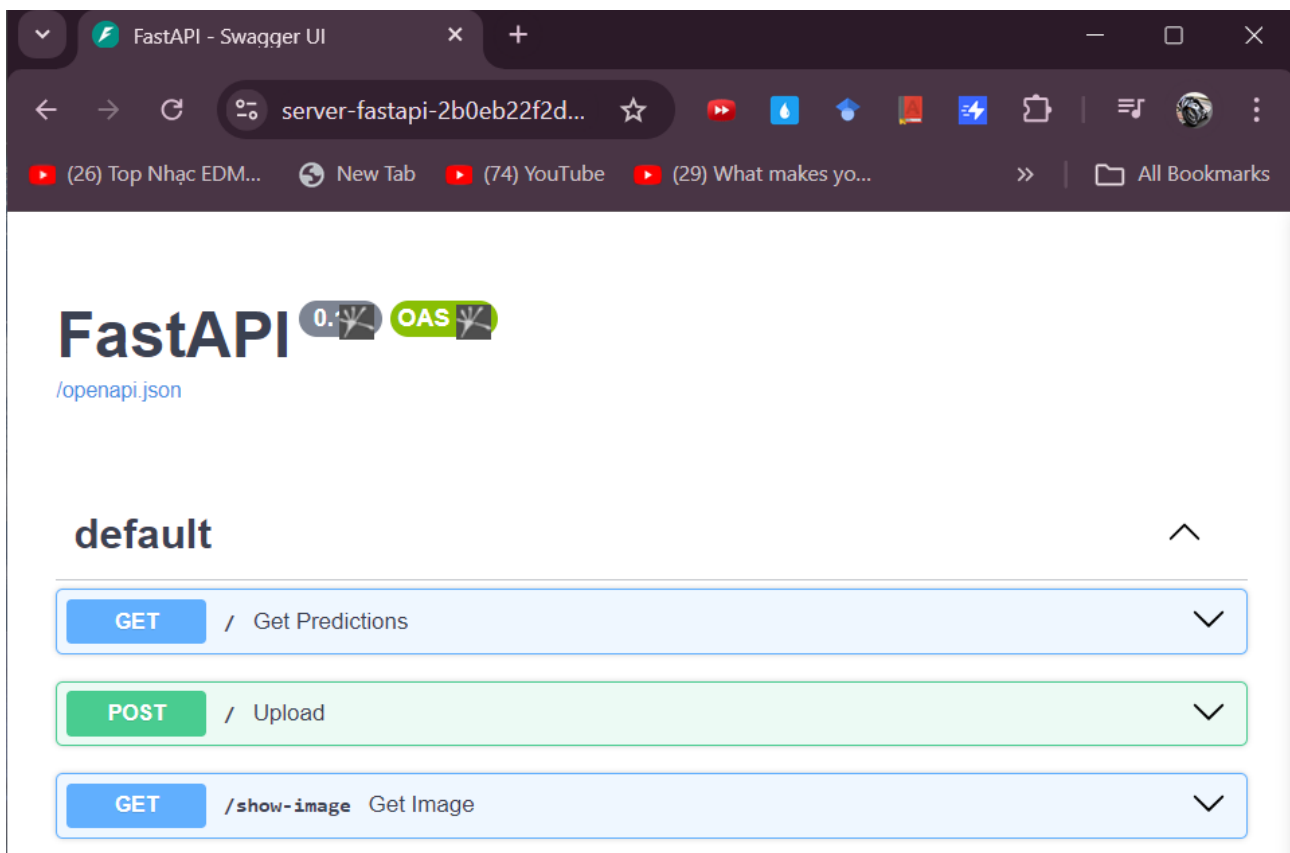
Hình 2.4: Activity diagram

Khi hệ thống bắt đầu hoạt động, các bộ xử lý bao gồm ESP32 và ESP32Cam sẽ kết nối với

Internet, sau đó ESP32Cam tiến hành thu thập hình ảnh từ đầu vào của hệ thống và gửi lên Server, quá trình thu thập và gửi hình ảnh lên Server cứ lặp lại cho đến khi hệ thống ngừng hoạt động. Khi Server nhận được hình ảnh từ ESP32Cam, nó sẽ lưu trữ hình ảnh này vào bộ nhớ và thực hiện dọn bộ nhớ nếu cần thiết. Khi có request từ ESP32, Server sẽ thực hiện dự đoán trên hình ảnh gần nhất và gửi giá trị dự đoán về ESP32, lúc này ESP32 nhận được Reponse từ Server sẽ thực hiện điều khiển servo dựa trên giá trị dự đoán, sau đó quá trình này sẽ lặp lại cho đến khi hệ thống ngừng hoạt động¹.

3 Cấu trúc Server

Đây là Server nhóm đã xây dựng bằng FastAPI (python) [4] và deploy trên heroku platform [5] <https://server-fastapi-2b0eb22f2d67.herokuapp.com/docs>.



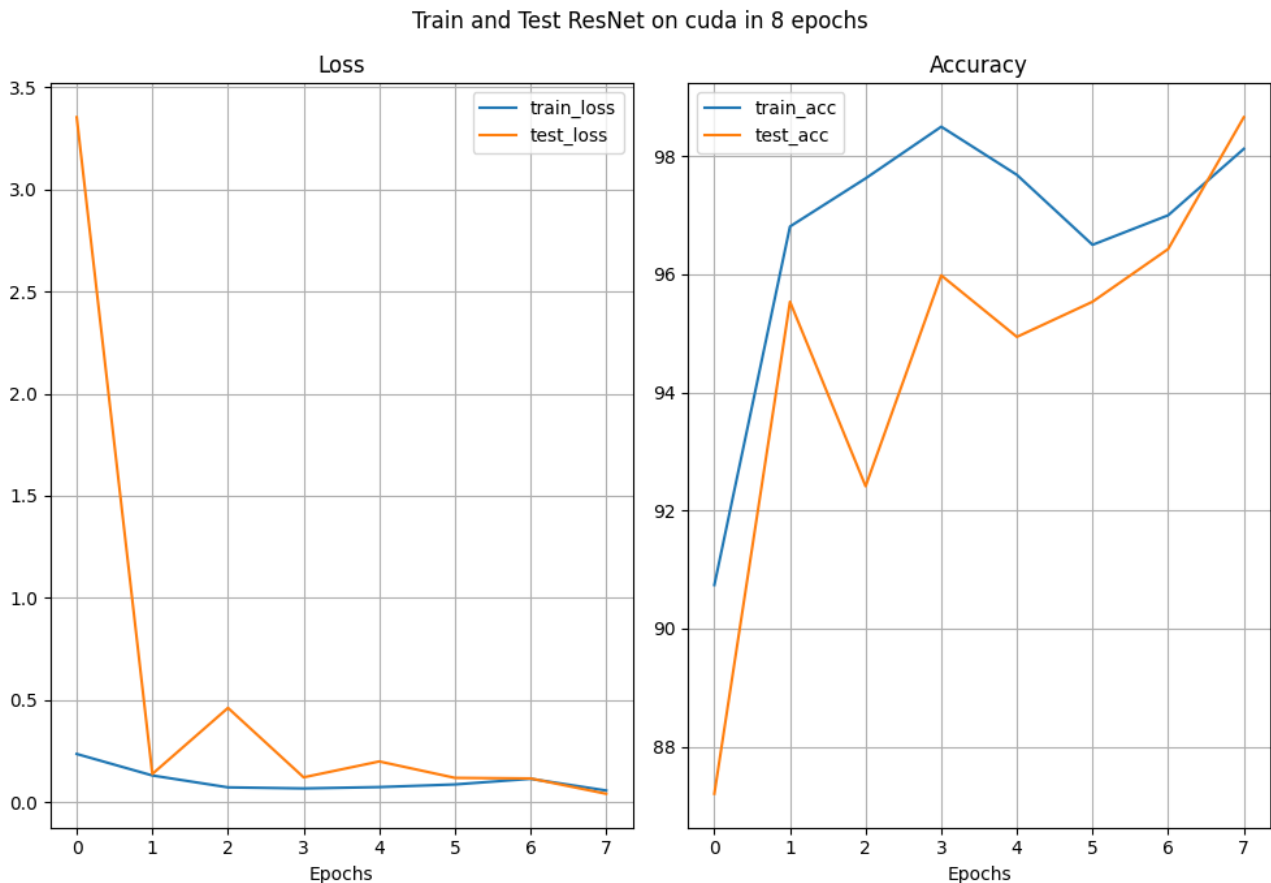
Hình 3.1: Cấu trúc API server

Hình 3.1 thể hiện cấu trúc của Server, bao gồm 2 phương thức chính. Thứ nhất là phương thức GET với chức năng dự đoán loại trái cây và trả về giá trị dự đoán với 0 là quả táo và 1 là quả cam. Thứ hai là phương thức POST, được xây dựng với mục đích nhận hình ảnh từ ESP32Cam và lưu trữ nó. Theo Use case diagram, Server có thêm một chức năng là dọn bộ nhớ ảnh, chức năng này được xây dựng để chạy trên background. Trong cấu trúc được thể hiện

¹Hệ thống ngừng hoạt động nghĩa là bị tác động từ bên ngoài như bị ngắt nguồn điện, hay người giám sát cố ý ngừng hệ thống...

tại hình 3.1 có thêm một phương thức GET là show-image, phương thức này được xây dựng để hiển thị hình ảnh gần nhất mà ESP32Cam gửi lên Server, với mục đích xác thực và kiểm tra.

Đối với phân phân loại trái cây, nhóm sử dụng PyTorch [6] để train nhiều mô hình, từ đó đánh giá chúng dựa vào các tiêu chí bao gồm: độ chính xác trong quá trình testing và predict, thời gian dự đoán, kích thước model. Nhóm tiến hành train và đánh giá 3 model bao gồm: ResNet50, ResNet18 và MobileNetV2. Hình 3.2 thể hiện đồ thị quá trình train ResNet50 model, tương tự như vậy cho 2 model còn lại, từ đó ta thu được kết quả được biểu diễn tại bảng 1.



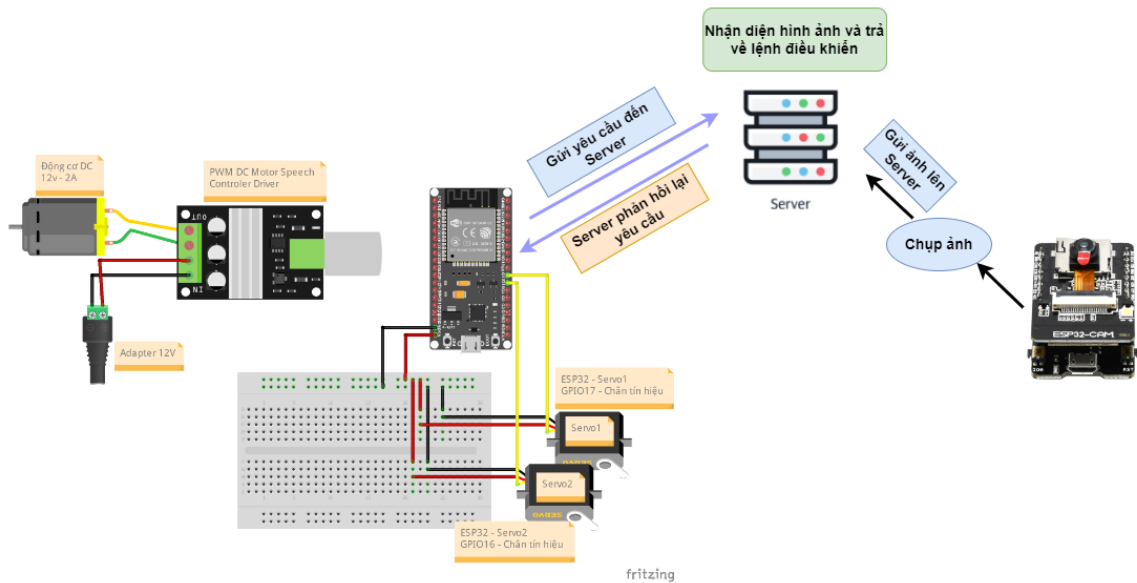
Hình 3.2: Training và Testing model

Name model	Accuracy Testing (%)	Accuracy Predict (%)	Time Predict (s)	Size (MB)
ResNet50	98.660714	97.001250	0.203992	89
ResNet18	97.321429	99.881893	0.178116	42
MobileNetV2	98.660714	99.923056	0.170975	8

Bảng 1: Đánh giá các mô hình AI.

Từ kết quả tại bảng 1, nhóm đã chọn mobileNetV2 để triển khai trên Server.

4 Sơ đồ khối phần cứng



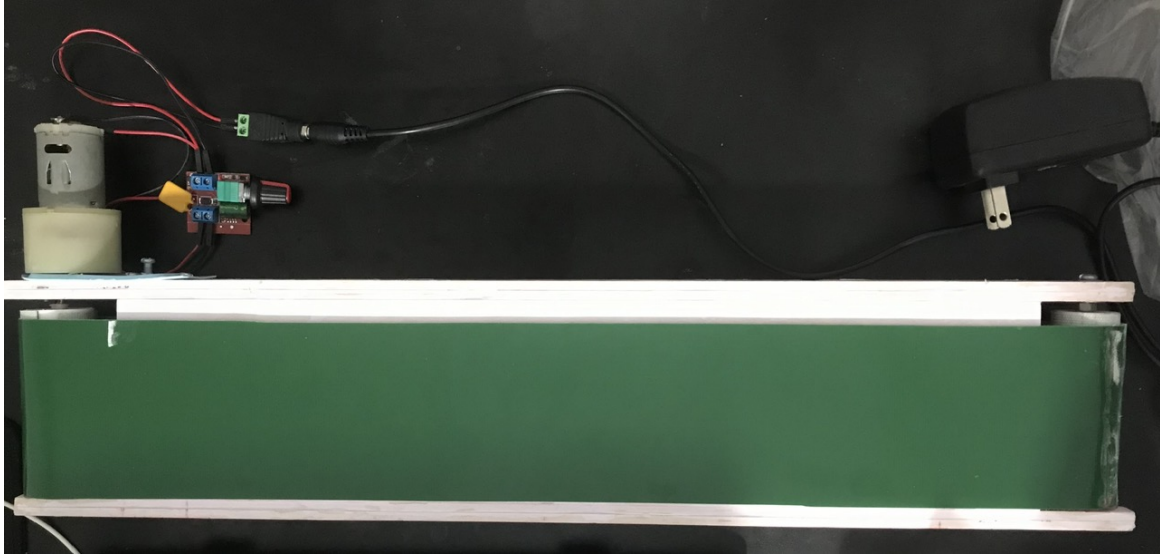
Hình 4.1: Sơ đồ mô tả tổng quát phần cứng

Hình 4.1 mô tả tổng quát về các bộ phận của phần cứng và cách mà hệ thống hoạt động, kết nối với nhau. Dưới đây là những mô tả chi tiết hơn về từng bộ phận, chức năng mà hệ thống đang có.

Để điều khiển băng tải hoạt động và chạy với tốc độ tùy chỉnh. Nhóm sử dụng một con Driver PWM DC Motor Speech Controller (12V - 2A) để điều khiển một động cơ DC (12V-2A) cũng chính là trục quay của băng tải. Và cấp nguồn cho nó là một Adapter 12V cấp vào chân âm dương của Driver.

ESP32 và ESP32 Cam được sử dụng với vai trò truyền nhận dữ liệu, tín hiệu để điều khiển quá trình phân loại. Đối với ESP32 Cam, có nhiệm vụ chụp hình ảnh đối tượng trên băng tải và gửi lên Server để thực hiện các bước xử lý mà nội dung ở phần 3 đã có đề cập đến. Còn về phía ESP32, lúc này sẽ gửi yêu cầu đến Server và Server phản hồi lại kết quả nhận được mà ESP32 Cam gửi lên. Lúc này ESP32 sẽ dựa vào nội dung phản hồi để điều khiển 2 Servo hoạt động. Quá trình này sẽ diễn ra liên tục cho đến khi hệ thống dừng hoạt động. Giả sử, Servo1 có nhiệm vụ là phân loại trái cam, Servo2 là trái táo. Nếu như nội dung phản hồi nhận được là 'orange', tức là quả cam thì lúc này Servo1 sẽ hoạt động còn Servo2 thì không và ngược lại đối với quả táo.

Hình 4.2 dưới đây là mô hình ảnh thực tế về phần cứng mà nhóm đã làm được. Nhóm chỉ mới hoàn thiện được một vài phần trong sơ đồ thiết kế phần cứng đã đề cập ở trên.



Hình 4.2: Hình ảnh phần cứng thực tế

5 Kết luận

Tính đến thời điểm hiện tại, nhóm đã triển khai hầu hết các chức năng của hệ thống, từ xây dựng hệ thống phần cứng bao gồm băng chuyền, các bộ xử lý và các linh kiện kèm theo, đến xây dựng Server.

Đề tài của nhóm đã đáp ứng được chủ đề của môn học PBL5 là hệ thống nhúng và IoT, với việc các bộ vi điều khiển kết nối đến Server thông qua Internet và trao đổi dữ liệu/tín hiệu thông qua nó.

Tuy nhiên hệ thống của nhóm hiện tại vẫn còn một số phần cần được đánh giá và cải thiện thêm, như cần giảm độ trễ trong quá trình tải ảnh từ ESP32Cam lên Server, tăng tốc độ và độ chính xác dự đoán của mô hình, thực hiện điều khiển servo trên ESP32 theo thời gian thực, đáp ứng đúng yêu cầu đầu ra của hệ thống là phân loại trái cây vào hộp trái cây phù hợp.

Phụ lục

- [1] Group's github repository: github.com/HODUCVU/Automatic-fruit-sorting-conveyor-system
- [2] Group's Server: <https://server-fastapi-2b0eb22f2d67.herokuapp.com/docs>
- [3] Dataset training model: [Apples Bananas Oranges](#)
- [4] FastAPI Tutorial: <https://fastapi.tiangolo.com/tutorial/>
- [5] Heroku Deploy Tutorial: <https://devcenter.heroku.com/articles/getting-started-with-python>
- [6] <https://pytorch.org/>