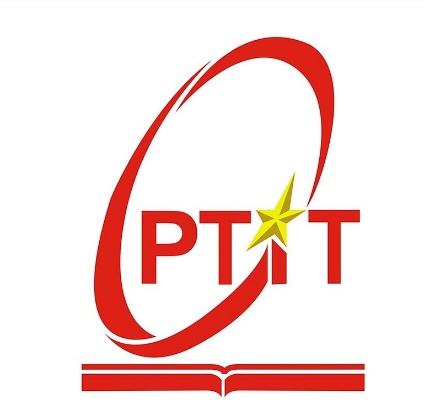
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

****

|  |  |
| --- | --- |
| **BÁO CÁO MÔN HỌC** | |
| **THỰC HÀNH CƠ SỞ** | |
| **Đề tài:** | |
| **Sử dụng KIT ESP32-CAM xây dựng ứng dụng phân loại rau củ trong cửa hàng nông nghiệp với sự trợ giúp của TinyML** | |
| **Họ và tên:** | **Vũ Khánh Kiên - B20DCDT104** |
|  | **Nguyễn Văn Danh - B20DCDT028** |
|  | **Nguyễn Thị Liên - B20DCDT115** |
| **Nhóm lớp:** | **09** |
| **Giảng viên:** | **TS. Nguyễn Đức Minh** |

**HÀ NỘI – 2023**

# LỜI MỞ ĐẦU

Trong nông nghiệp hiện đại, việc xác định loại trái cây và hoa quả là một công việc quan trọng để đảm bảo sự phát triển và năng xuất của ngành nông nghiệp. Tuy nhiên, việc nhận diện này thường tốn nhiều thời gian và công sức của con người. Với sự phát triển của các công nghệ nhận diện hình ảnh và trí tuệ nhân tạo (AI), các giải pháp tự động hóa này đã trở nên khả thi hơn bao giờ hết.

Trong bài viết này, chúng ta sẽ tìm hiểu về việc sử dụng ESP32 Cam và TinyML để phát triển một hệ thống nhận diện trái cây và hoa quả tự động trong nông nghiệp, giúp tối ưu hóa quá trình sản xuất và nâng cao hiệu suất của ngành nông nghiệp.

Trước khi đi vào nội dung, chúng em xin gửi lời cảm ơn tới các thầy cô “Trung tâm thí nghiệm thực hành” và thầy cô khoa “KTĐT 1” đã tâm huyết giảng dạy để chúng em có được nhiều kiến thức bổ ích. Xin chúc các thầy cô luôn thành công trên con đường sự nghiệp giảng đường của mình.

Mục Lục

[LỜI MỞ ĐẦU 1](#_Toc135346542)

[CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc135346543)

[I. Mô tả dự án 4](#_Toc135346544)

[II. ESP32 Cam Ai-Thinker 4](#_Toc135346545)

[III. Các chân ESP32 Cam 5](#_Toc135346546)

[IV. Đế nạp ESP32 Cam 5](#_Toc135346546)

[CHƯƠNG II: CÀI ĐẶT VÀ CHẠY THỬ SẢN PHẨM THỰC TẾ 7](#_Toc135346547)

[I. Các bước cài đặt dự án 7](#_Toc135346548)

[1. Thiết kế trên Edge Impulse 7](#_Toc135346549)

[2. Cài đặt trên Arduino IDE 13](#_Toc135346550)

[II. Chạy thử sản phẩm 16](#_Toc135346551)

[Chương III : Kết luận 19](#_Toc135346552)

[I. Ưu điểm 19](#_Toc135346553)

[II. Nhược điểm 19](#_Toc135346554)

[III. Hướng phát triển 19](#_Toc135346555)

[CÁC TÀI LIỆU THAM KHẢO 20](#_Toc135346556)

# CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

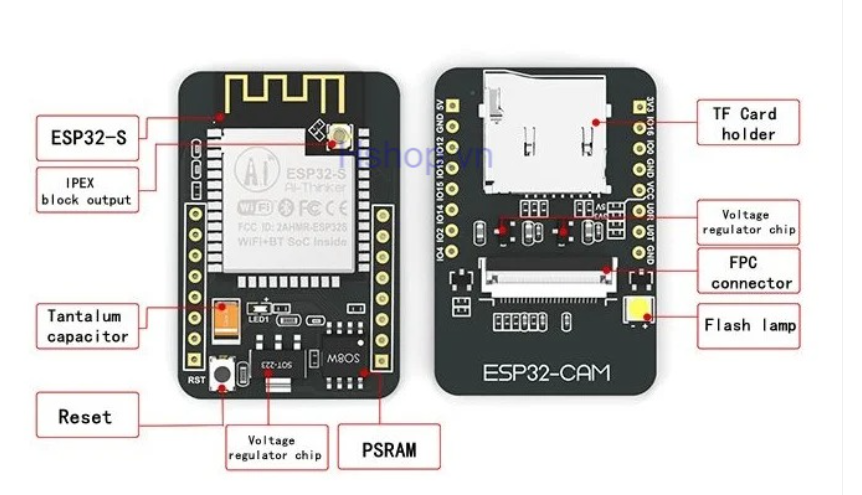
1. **Mô tả dự án**

* Dự án sử dụng ESP32 Cam Ai-Thinker với sự trợ giúp của Egde Impulse để phân loại rau củ

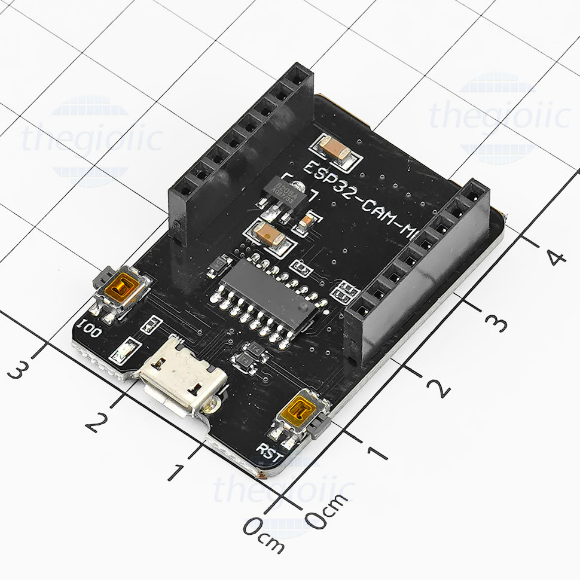
1. **ESP32 Cam Ai-Thinker**

**Thông số kỹ thuật:**

* Model: ESP32-CAM Ai-Thinker
* Module trung tâm: Ai-Thinker ESP32-S
* Power Supply: 5VDC (nguồn từ 2A trở lên)
* Điện áp giao tiếp GPIO: 3.3VDC
* SPI Flash: Default 32Mbit
* RAM: 520KB SRAM +4M PSRAM
* Bluetooth: Bluetooth 4.2 BR/EDR and BLE standards
* Wi-Fi: 802.11 b/g/n/
* Support interface: UART, SPI, I2C, PWM
* Support TF card: Maximum support 4G
* IO port: 9
* UART Baudrate: Default 115200 bps
* Image Output Format: JPEG( OV2640 support only ),BMP,GRAYSCALE
* Spectrum Range: 2412 ~2484MHz
* Antenna: Onboard PCB antenna, gain 2dBi
* Transmit Power:
  + 802.11b: 17±2 dBm (@11Mbps);
  + 802.11g: 14±2 dBm (@54Mbps);
  + 802.11n: 13±2 dBm (@MCS7)
* Receiving Sensitivity:
  + CCK, 1 Mbps : -90dBm;
  + CCK, 11 Mbps: -85dBm;
  + 6 Mbps (1/2 BPSK): -88dBm;
  + 54 Mbps (3/4 64-QAM): -70dBm;
  + MCS7 (65 Mbps, 72.2 Mbps): -67dBm
* Power Dissipation:
  + Turn off the flash lamp:180mA@5VDC
  + Turn on the flash lamp and turn on the brightness to the maximum:310mA@5VDC
  + Deep-sleep: Minimum power consumption can be achieved 6mA@5VDC.
  + Moderm-sleep: Minimum up to 20mA@5VDC.
  + Light-sleep: Minimum up to 6.7mA@5VDC
* Security: WPA/WPA2/WPA2-Enterprise/WPS
* Dimensions: 40.5mm x27mm x4.5mm

1. **Các chân ESP32 Cam**

**IV. Đế nạp ESP32 Cam**



Đế nạp chương trình ESP32-CAM USB Programming Adapter giúp mạch [ESP32-CAM](https://hshop.vn/products/kit-rf-thu-phat-wifi-ble-esp32-cam) nạp chương trình và giao tiếp truyền dữ liệu với máy tính qua giao IC chuyển USB-UART CH340 một cách dễ dàng, đế còn tích hợp hai nút nhấn cho chân RST và IO0 của [ESP32-CAM](https://hshop.vn/products/kit-rf-thu-phat-wifi-ble-esp32-cam) để sử dụng trong quá trình nạp chương trình.

**Thông số kỹ thuật:**

* Model: ESP32-CAM-MB
* Sử dụng để nạp chương trình và giao tiếp truyền dữ liệu với máy tính cho mạch [ESP32-CAM.](https://hshop.vn/products/kit-rf-thu-phat-wifi-ble-esp32-cam)
* Điện áp sử dụng: 5VDC từ cổng MicroUSB
* IC chuyển giao tiếp USB-UART: CH340
* Tích hợp hai nút nhấn RST và IO0 sử dụng trong quá trình nạp chương trình.
* Kích thước: 27x40mm

**Hướng dẫn sử dụng:**

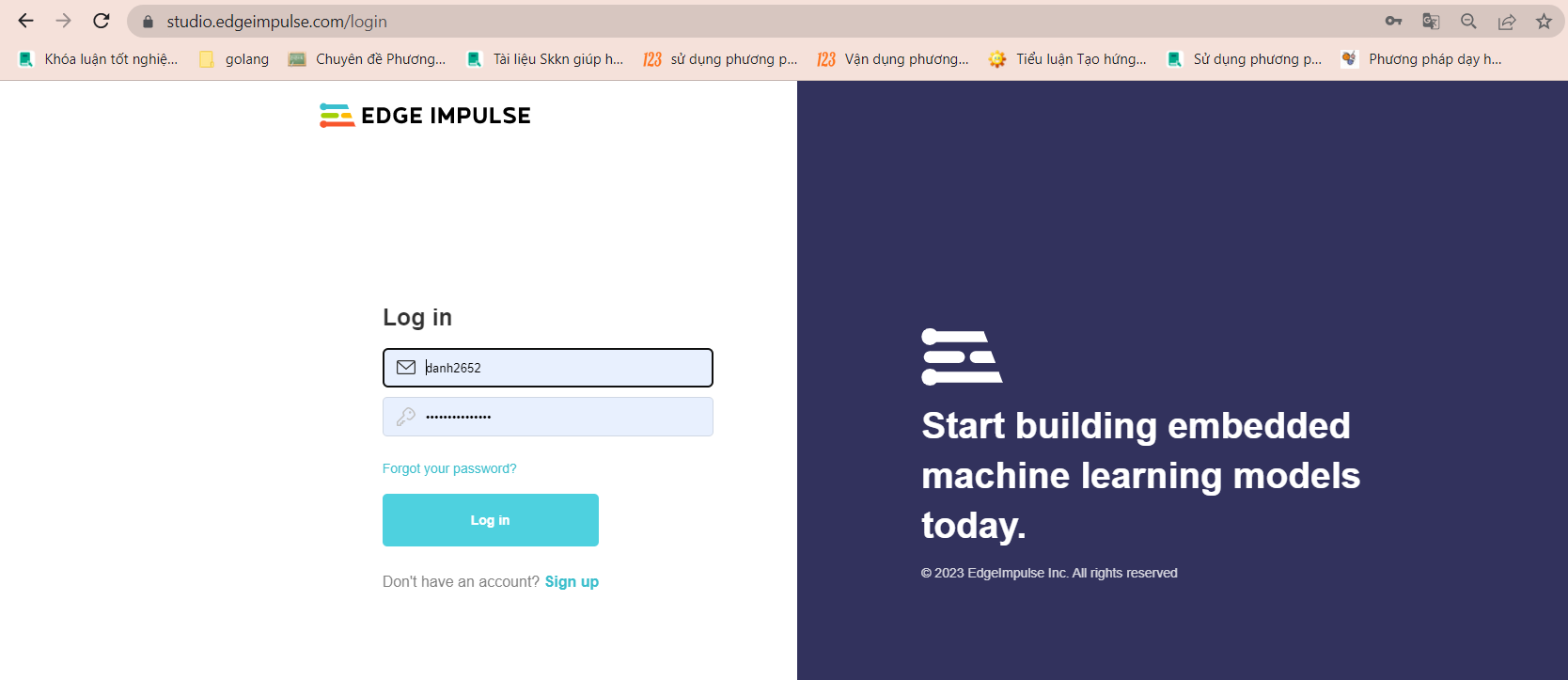
1. Tải Driver CH340 và cài đặt trên máy tính.
2. Trước khi nhấn nạp chương trình (Upload) trên phần mềm Arduino nhấn giữ nút IO0 trên đế nạp để vào chế độ nạp chương trình (Upload Mode) cho ESP32-CAM.
3. Vẫn nhấn giữ nút IO0 đồng thời nhấn nút nạp chương trình (Upload) trên phần mềm Arduino khi nào thấy hiển thị Connecting....thì nhấn nút RST (Reset) trên mạch (nhấn 1 lần không cần nhấn giữ) để tiến hành nạp chương trình cho đến khi hoàn tất.
4. Bỏ nhấn nút IO0 để thoát chế độ nạp chương trình (Upload Mode) và thấy chương trình hoạt động.

# CHƯƠNG II: CÀI ĐẶT VÀ CHẠY THỬ SẢN PHẨM THỰC TẾ

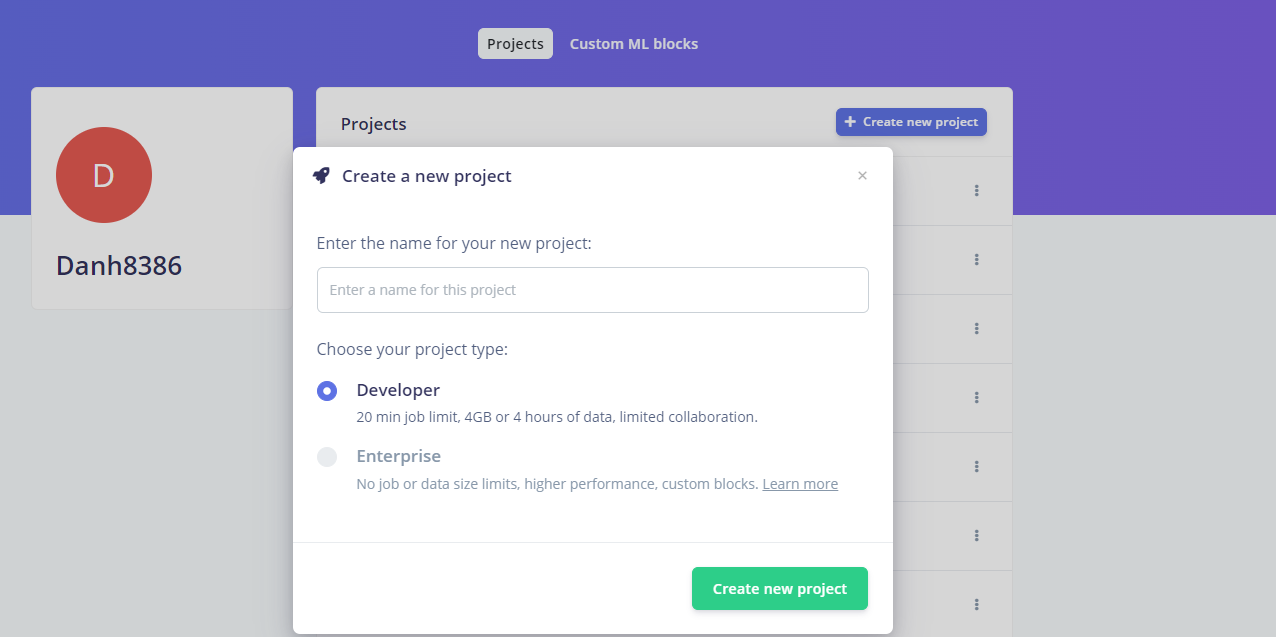
## Các bước cài đặt dự án

1. **Thiết kế trên Edge Impulse**

* B1: Đăng nhập vào Edge Impulse
* Đây là web tích hợp TinyML (Tiny Machine Learning), sử dụng TinyML để học và giúp ta thao tác với dự án dễ dàng hơn.

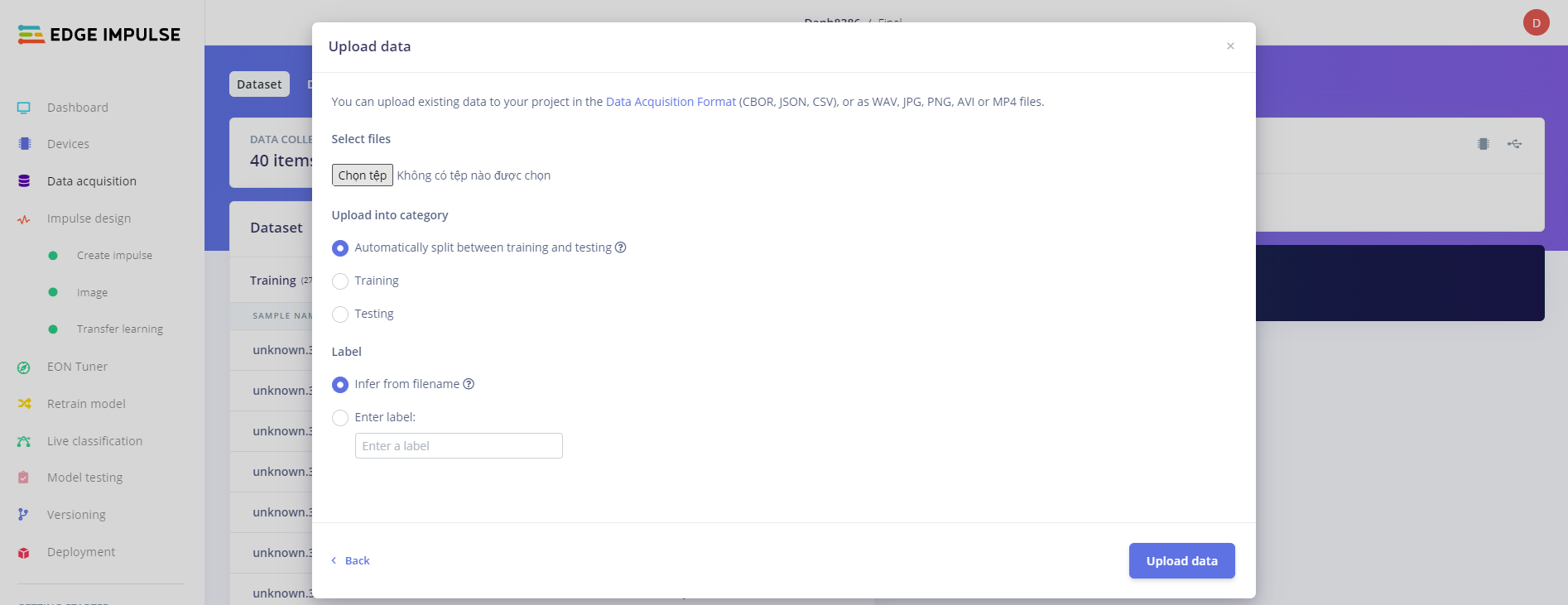


* B2: Tạo Project
* Sau khi đăng nhập vào web, chúng ta sẽ tạo 1 project theo tên đề tài sản phẩm mà chúng ta làm.

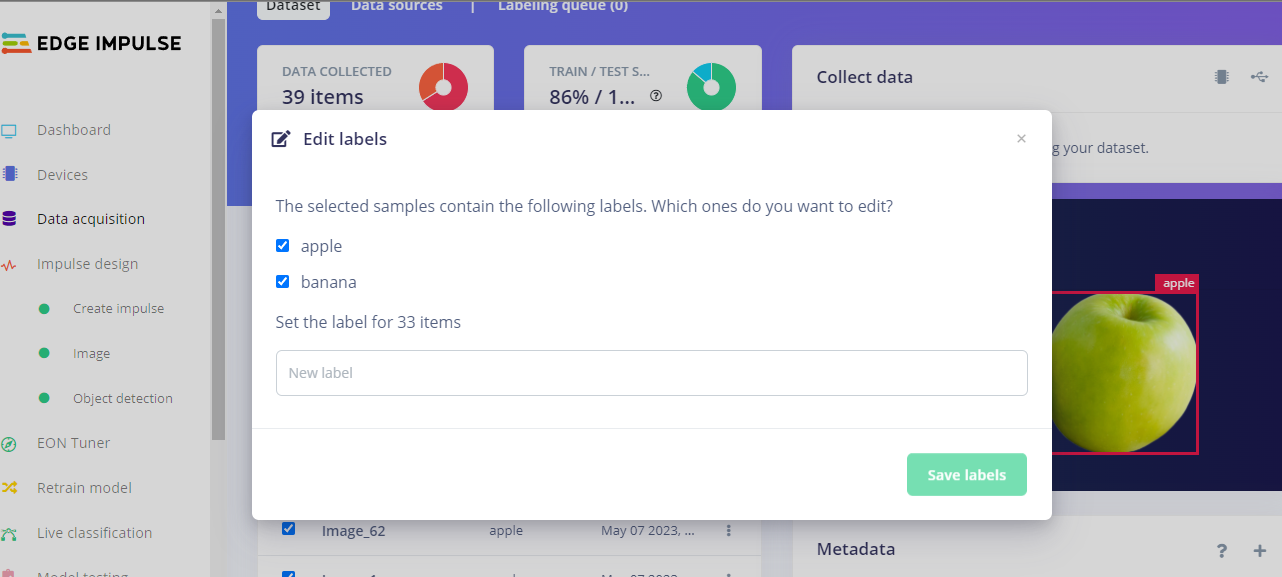
****

* B3: Chọn Data acquisition → Upload data
* Data acquisition là nơi mà ta sẽ các ảnh mà chúng ta muốn cho AI học để nhận diện nó, càng nhiều ảnh thì AI sẽ càng học được nhiều – nghĩa là nhận dạng càng chuẩn.

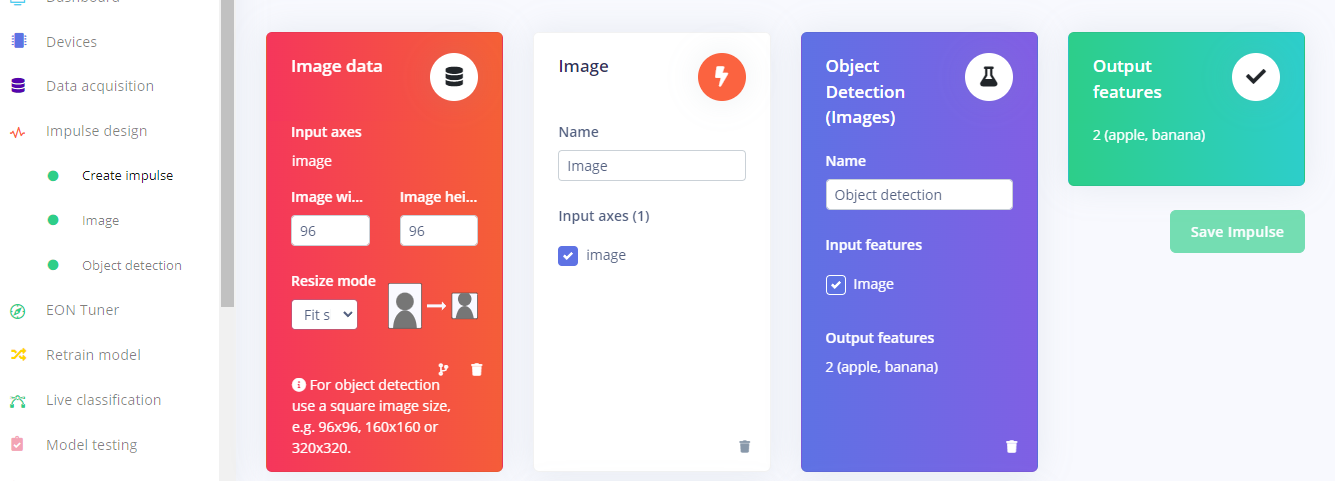
Sau đó chọn ảnh hoa quả mình muốn nhận diện rồi ấn Upload data



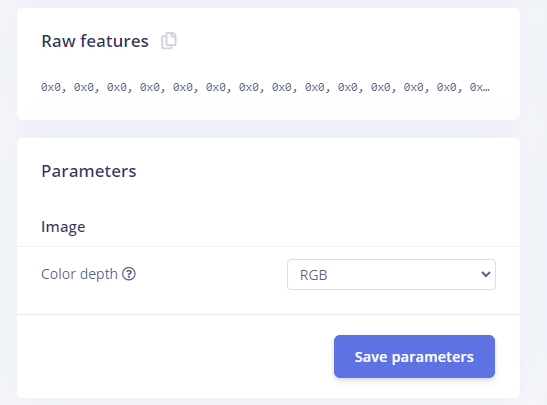
* B4: Đặt tên Label cho từng ảnh
* Việc đặt tên label cho từng ảnh là cần thiết, đặt label cho ảnh để AI học nhận diện và phân biệt từng sản phẩm mà ta upload vào kho dữ liệu.



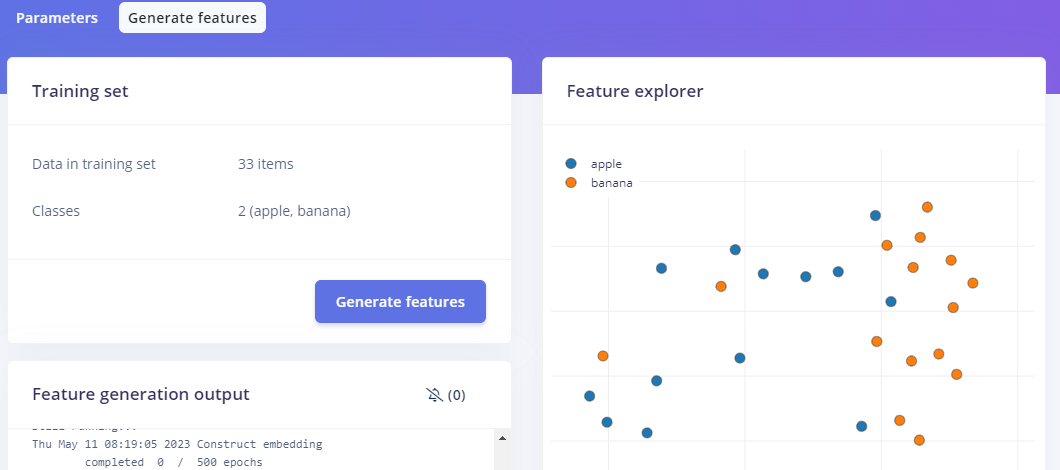
* B5: Chọn Impulse design → Create impulse → chọn các cài đặt như hình dưới → Save Impulse
* Đây là mục chúng ta tùy chỉnh dữ liệu, hầu như mặc định sẽ như hình dưới, ngoài ra, nếu chúng ta sử dụng phần cứng, camera cao cấp hơn thì chúng ta có thể cài đặt độ pixel cao hơn để ảnh AI nhận được qua camera sắc nét, chính xác hơn.



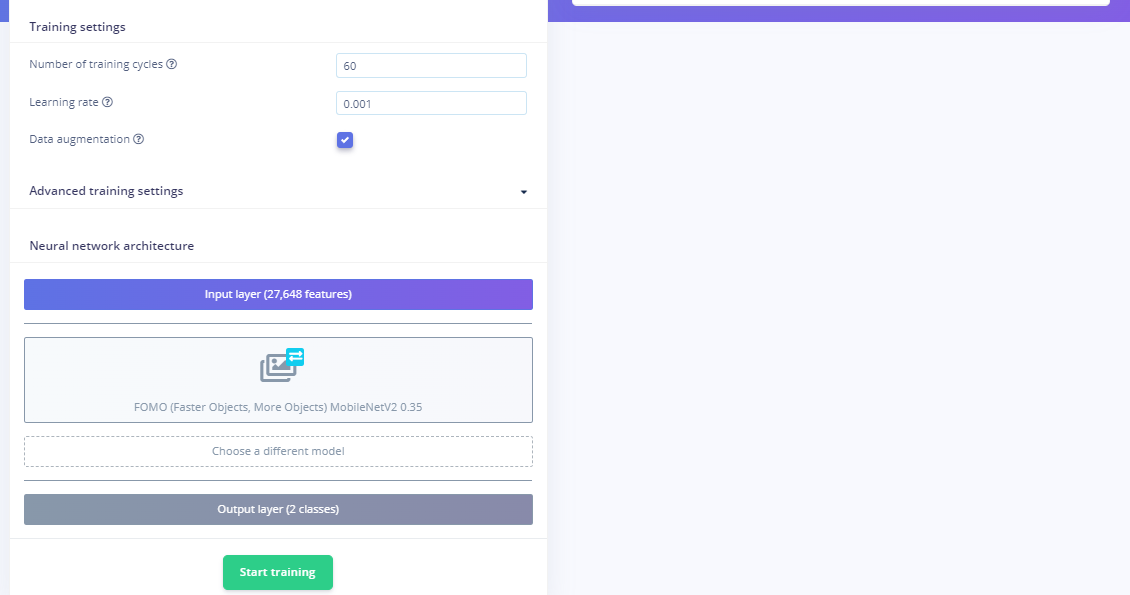
* B6: Sang mục Image → Save parameters
* Sau khi add các tùy chọn mà ta sử dụng cho AI, ta sẽ cấu hình từng tùy chọn.
* Chú ý, có thể chọn Grayscale thay thì RGB, AI sẽ nhận ảnh đen trắng để nhận diện nếu ta chọn Grayscale – đồng nghĩa project khi ta xuất ra sẽ tốn ít dữ liệu hơn ảnh RGB (ảnh màu).



Sau đó ấn Generating features



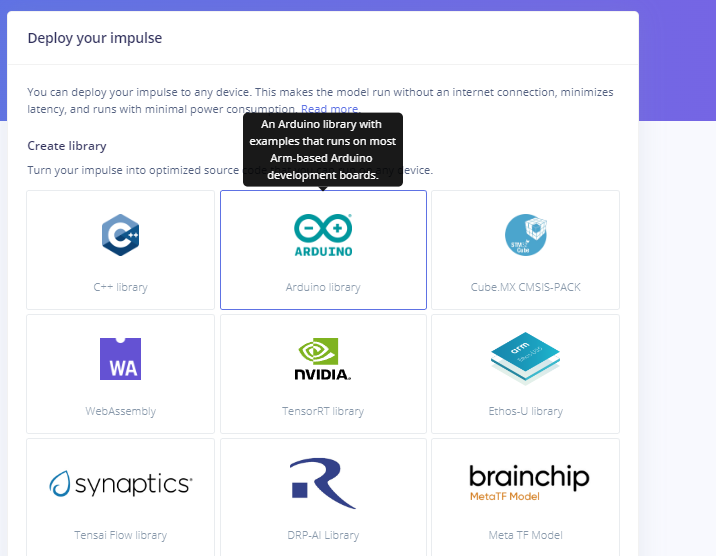
* Tại đây, nếu càng nhiều điểm chấm trên bảng Feature explorer → AI sẽ nhận diện sản phẩm càng chuẩn hơn (để có nhiều điểm ảnh, ta phải up càng nhiều dữ liệu).
* B7: Sang mục Object detection → chỉnh các thông số như ảnh dưới → chọn Start training
* Ở bước này chúng ta sẽ chọn model cho sản phẩm, model là công cụ quan trọng cần chọn lựa, model giống như bộ não của AI, phụ thuộc vào phần cứng mà ta sử dụng có cao cấp hay không, ta phải chọn model phù hợp với phần cứng và loại việc mà AI làm.



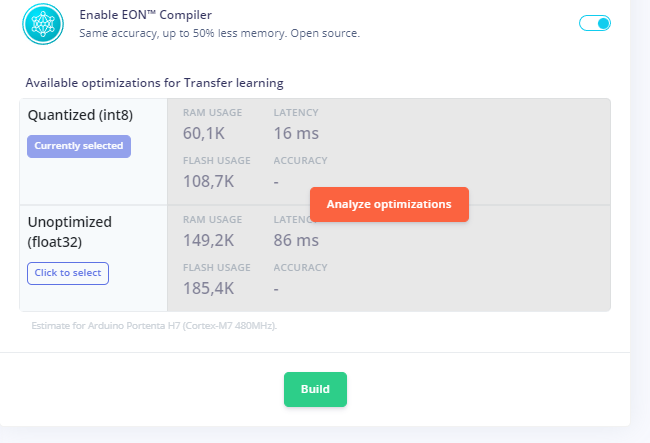
Sau khi load xong sẽ ra bảng như dưới



* Tại bảng này, khi ta thấy khả năng nhận diện từng loại quả ở xác suất ổn, trên 70% là được.
* B8: Chọn Deployment → chọn Arduino để xuất ra thư viện của Arduino IDE
* Deployment là mục ta xuất AI mà ta đã tạo ra thành 1 thư viện và sử dụng nó trong lập trình.



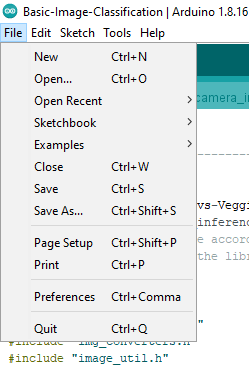
Kéo xuống và chọn Build → lưu file rar của thư viện



1. **Cài đặt trên Arduino IDE**

* Đầu tiên, để sử dụng ESP32 Cam AI-Thinker trên Arduino IDE, ta phải cấu hình Arduino IDE.

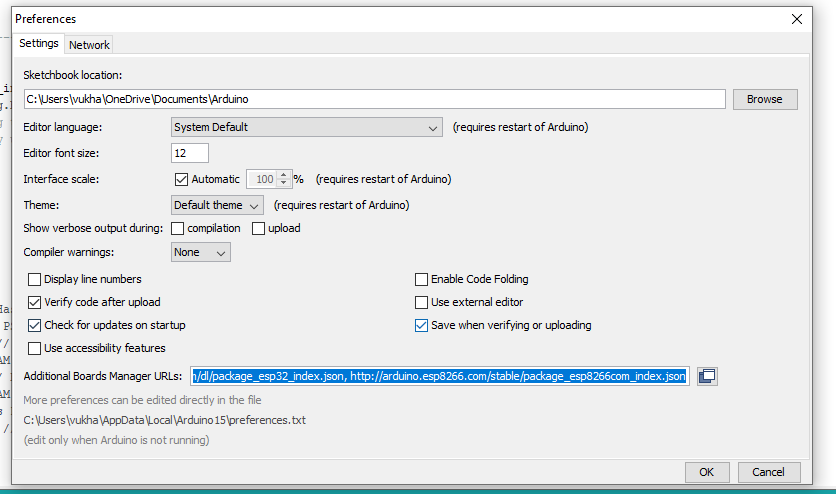
Vào code Arduino → chọn Preferences



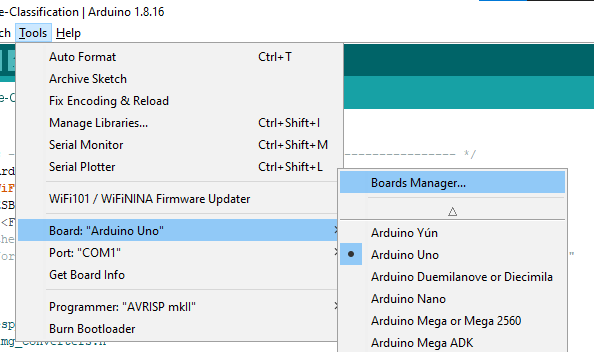
Copy dòng dưới:

<https://dl.espressif.com/dl/package_esp32_index.json>**,** <http://arduino.esp8266.com/stable/package_esp8266com_index.json>

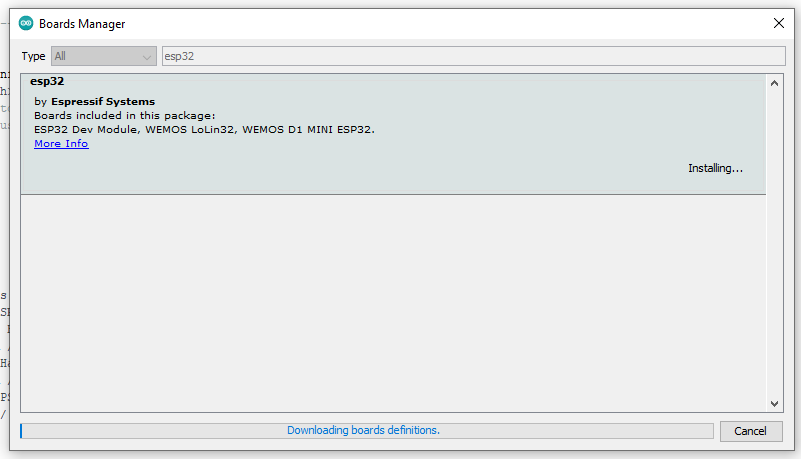
Sau đó tại mục **Additional Boards Manager URLs**  dán dòng copy trên → OK



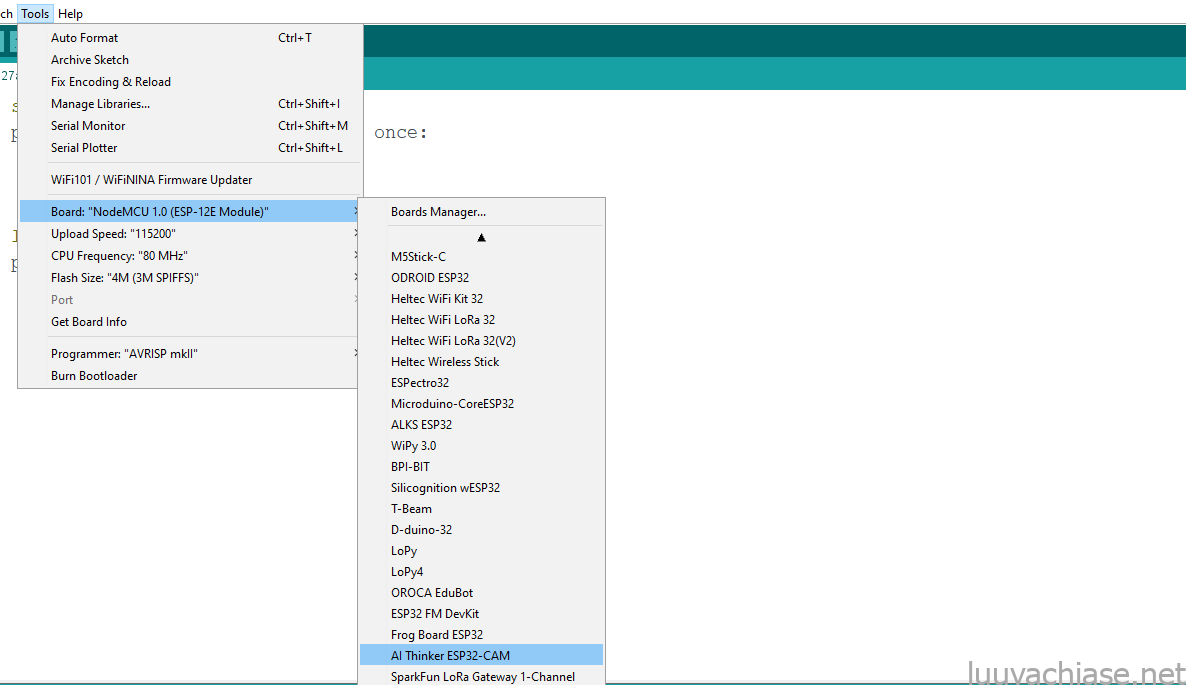
Sau đó tìm vào menu **Tools** → **Boards Manager**



Tìm esp32 → Cài đặt

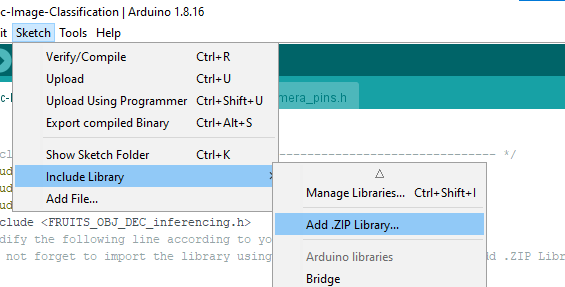


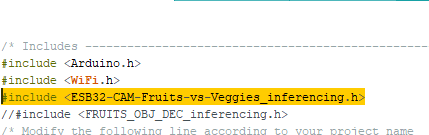
Vào **Tools** → **Boards** → **AI-Thinker ESP32-CAM**



## Chạy thử sản phẩm

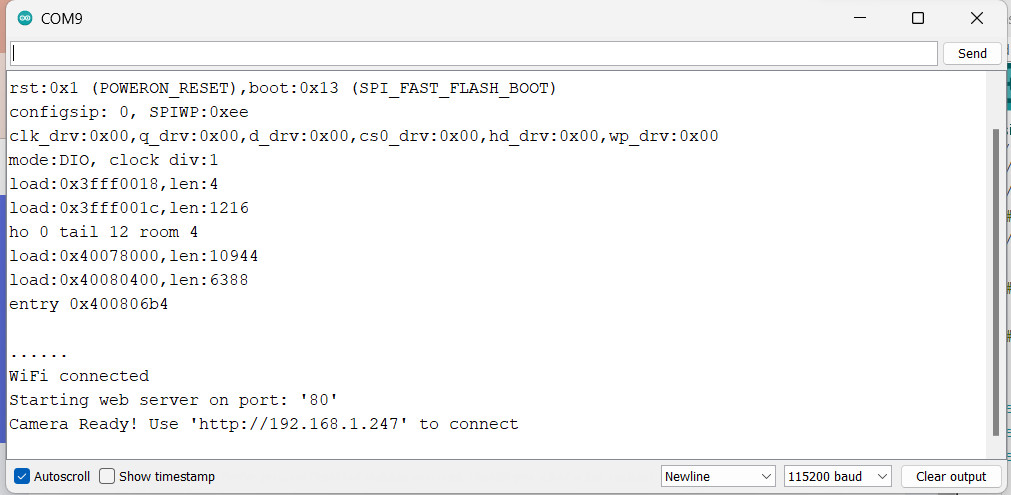
Add thư viện RAR đã tải ở bước 8 (phần 1) vào Arduino





Sau đó chọn COM và nạp chương trình

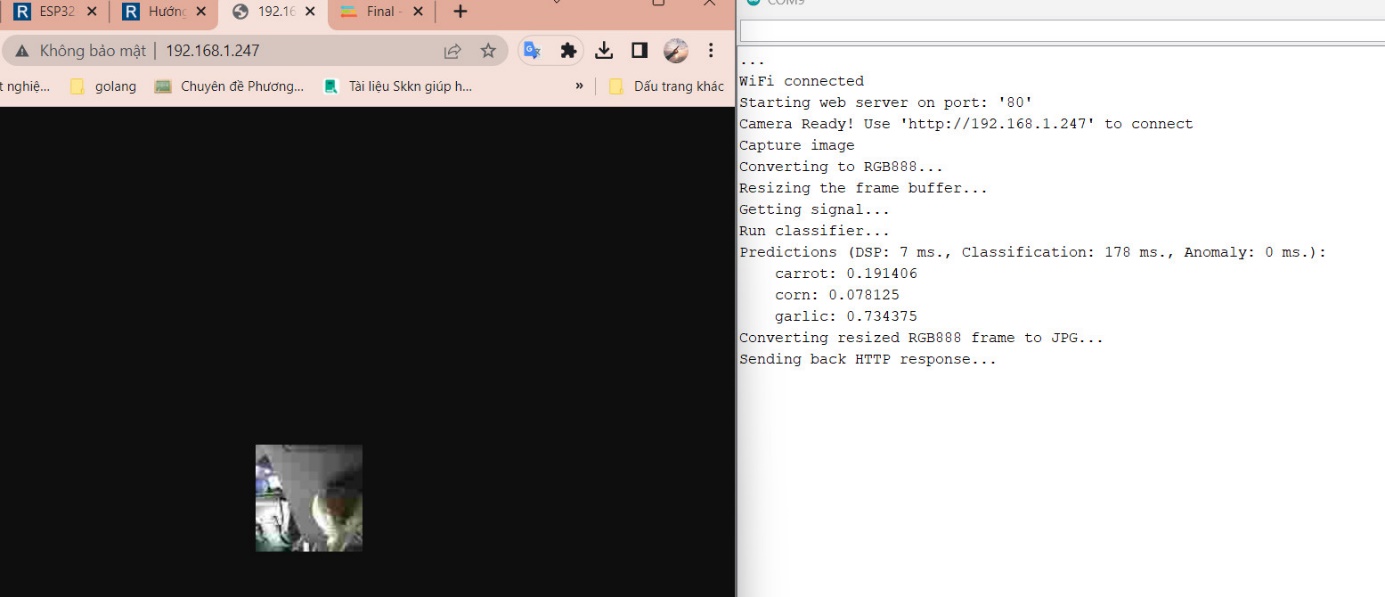
Sau khi nạp xong mở Serial Monitor kết nối wifi cho ESP32

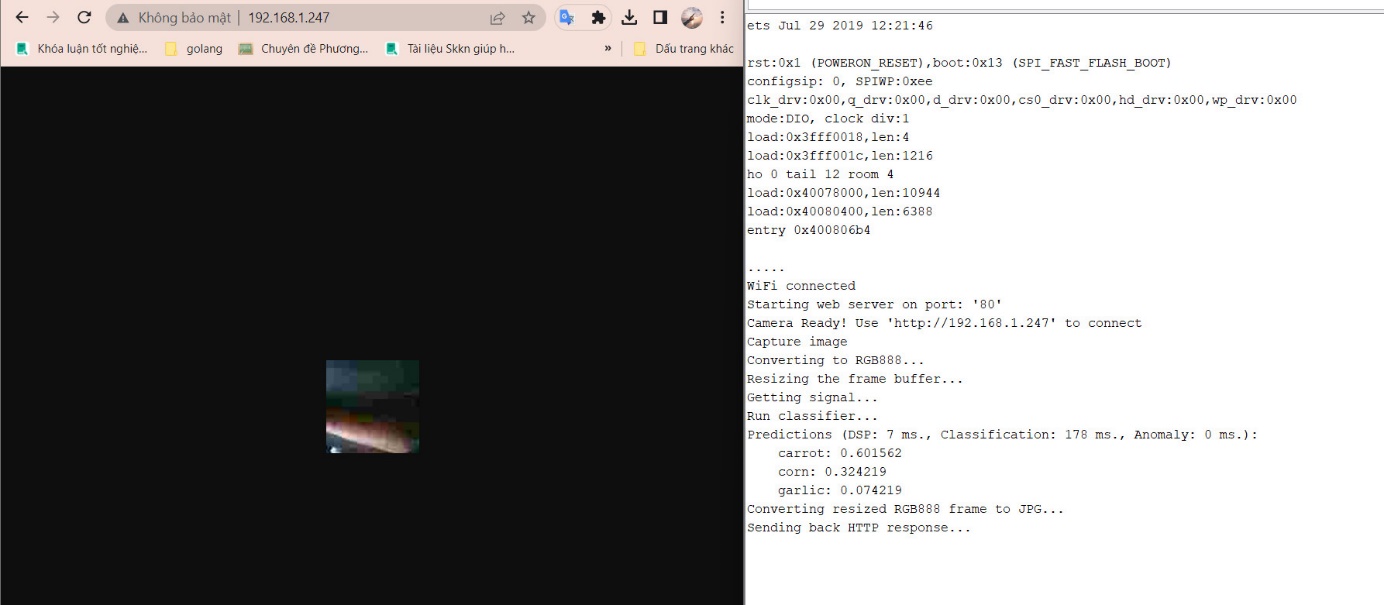


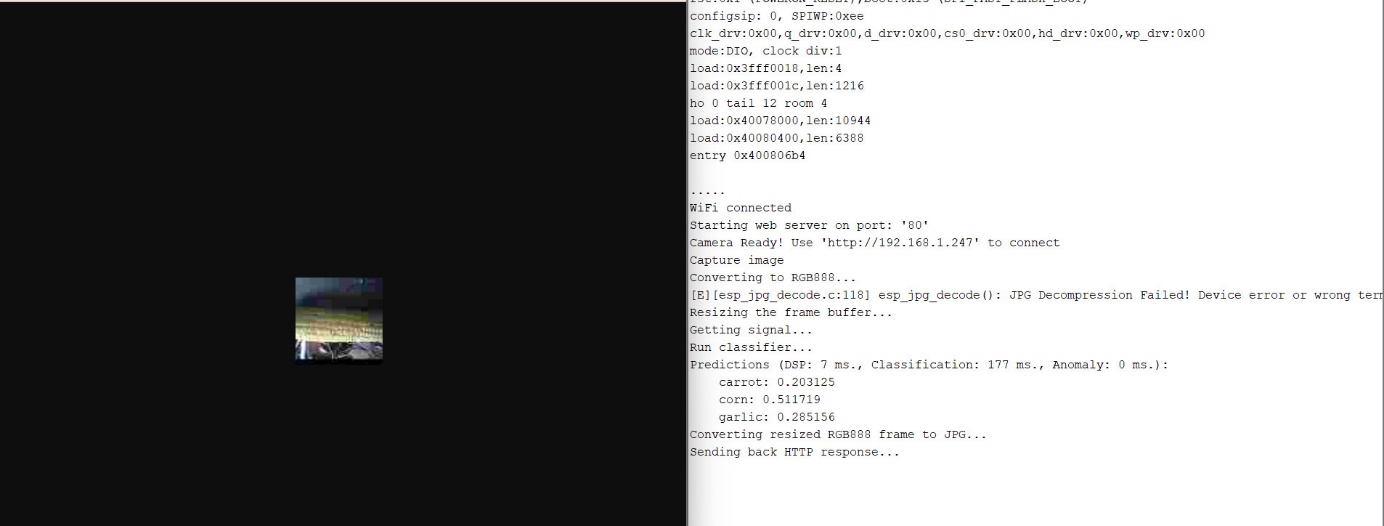
Copy link http trong Serial vào google để ra link chụp ảnh của ESP

Reload lại trang để chụp ảnh → kết quả nhận dạng sẽ hiện ra trên Serial









Xác suất nhận diện sản phẩm phụ thuộc vào phần cứng mà ta sử dụng, chỉ cần quả mà ta muốn nhận diện có xác suất lớn hơn nhiều so với các quả khác là dự án nhận diện rau củ bằng ESP32 Cam thành công.

# Chương III : Kết luận

## I. Ưu điểm

- Sử dụng ESP32 Cam kết hợp với Edge Impulse dễ dàng cài đặt và sử dụng.

- Độ bảo mật cao.

## II. Nhược điểm

- Sản phẩm chưa được thực sự nhanh nhạy.

- Phải thử nghiệm nhiều lần và đo nhiều lần mới cho ra kết quả chính xác.

## III. Hướng phát triển

- Nghiên cứu và phát triển thêm để khắc phục những nhược điểm trên.

- Thay đổi phần cứng và tối ưu cài đặt trong Edge Impulse để có thể tăng độ nhanh, chính xác của sản phẩm.

# CÁC TÀI LIỆU THAM KHẢO

* Hướng dẫn làm dự án:

<https://www.hackster.io/mjrobot/esp32-cam-tinyml-image-classification-fruits-vs-veggies-4ab970>

* Link code:

<https://github.com/edgeimpulse/example-esp32-cam/tree/main/Examples/Basic-Image-Classification>

* Kho ảnh dataset:

<https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition>