Cj Gorospe
& Macy Varga

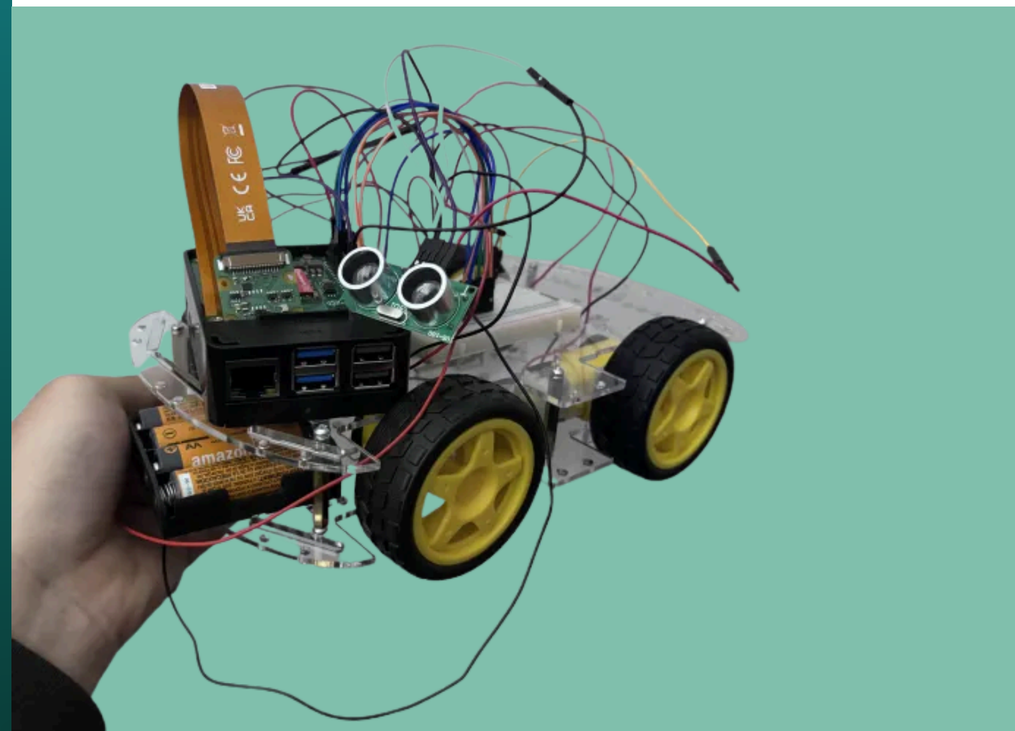
Autonomous Leaf Detection • Disease Classification • Obstacle Avoidance • Dashboard Control

Plant Health Monitoring Robot

What is the Project?

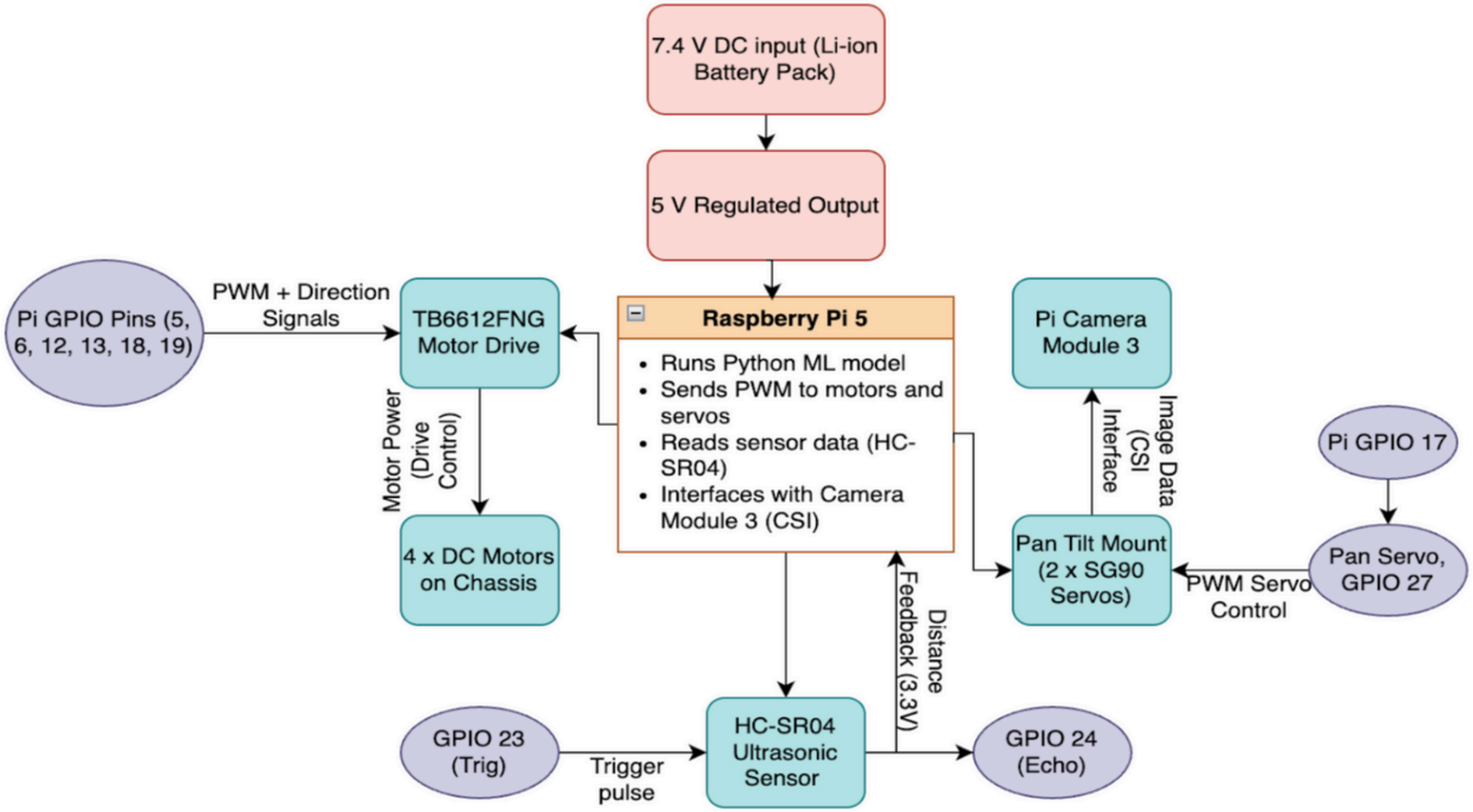
A Fully Autonomous Rasberry Pi Robot

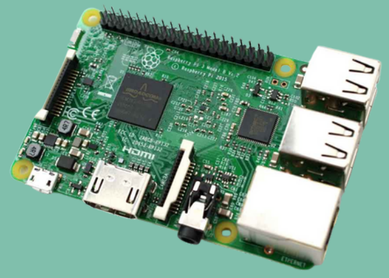
- Drive on Its own
- Detect leaves using Pi camera
- Classify those leaves as healthy or diseased
- Avoid obstacles
- Save images + classification results
- Provide a live dashboard for monitoring

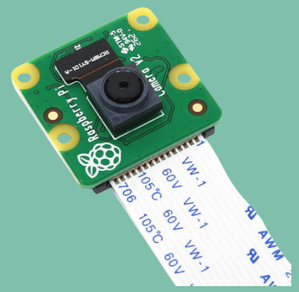


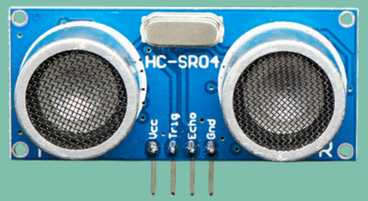
System Overview

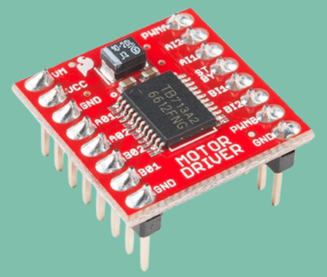
- Rasberry Pi running Python + TensorFlow Lite
- PiCamera3 for real time video
- TB6612 motor driver controlling 4 motors
- Ultrasonic sensor for obstacle detection
- Battery power system
- Dashboard control Interface
- Machine learning classification pipeline











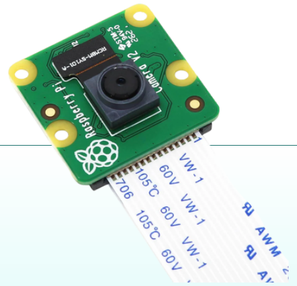
Components Used

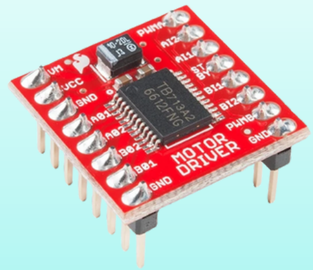
Hardware

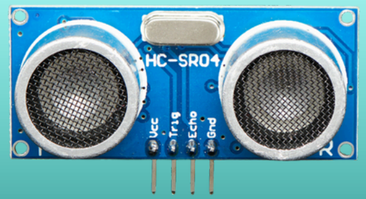
- Rasberry Pi 4
- PiCamera Module 3
- TB6612 H-Bridge Motor Driver
- 4 DC motors + chassis
- 4xAA battery pack for motors
- US-100 Ultrasonic sensor
- Breadboard + jumper wires

Software

- Python
- TensoreFlow / TensorFlowLite
- OpenCV
- Flack Web Server
- GPIO motor control







ML

Dataset & Model Approach

Dataset – Plant Village

Thousands of clean, labeled Images
Potato, tomato, and pepper leaves
Contains both healthy and diseased examples
Clean backgrounds → good for Initial training

Two Gate Classification System

Gate 1:

If model confidence < 0.8 → “Not Leaf”

Gate 2:

If confidence < 0.6 → “Unsure”
Else → Healthy or Diseased

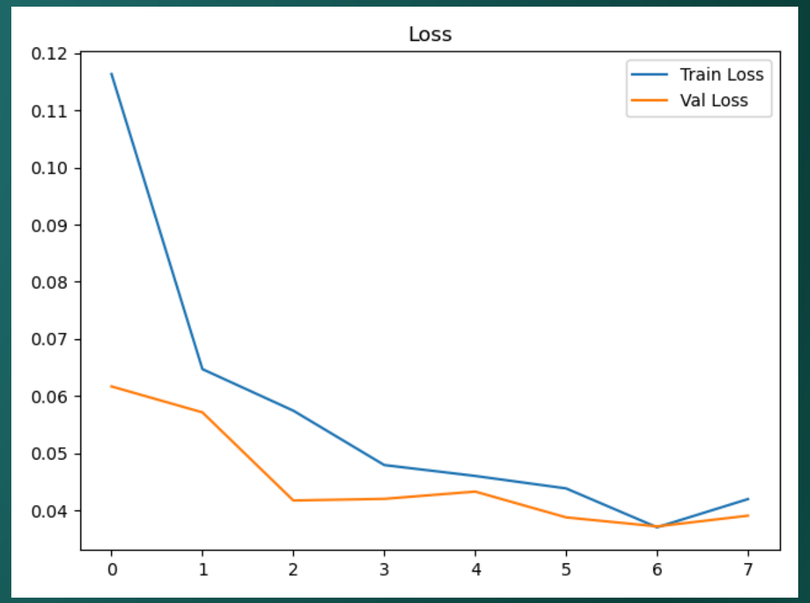
Data Processing for Model Training

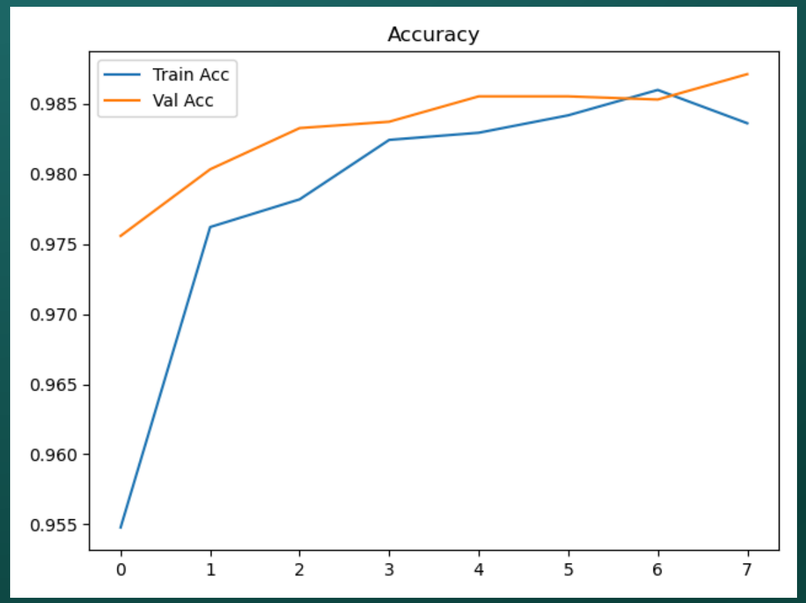
Normalize pixels to 0-1
data augmentation (rotation, shift, zoom, flip, brightness)
Resize Image to 224x224 → needed for MobileNetV2
80/20 train/val split

Model Architecture: MobileNetV2 with Transfer Learning

MobileNetV2: Lightweight CNN for mobile and embedded devices
Pretrained on ImageNet: dataset with 14M Images across 1,000 categories (body knows edges, textures, shapes, and color variations)
Remove head, freeze pretrained weights, add dense layers for classes: healthy and diseased

Model Results





Loss Curve

Both training and val loss consistently decrease over the eight epochs. Training loss begins about 0.1116 and drops to 0.042. Val loss follows very similar trend, ending slightly below training loss, due to augmented training images are harder than clean val images. Model Is learning effectively and no meaningful overfitting.

Accuracy Curve

Training accuracy Increases from 95% to 98.5%, while val accuracy Increases from 97.5% to 98.7%. Confirms that MobileNetV2 with transfer learning Is a very good fit for the model. ImageNet features efefctively capture leaf characteristics

Results show strong learning, minimal overfitting, and high classification performance, giving the robot a reliable baseline for disease detection

Interface

Rasberry Pi Interface Pipeline

The Pi performs:

1. Capture a frame with PiCamera
2. Resize to 224 x 224
3. Noralize Pixels
4. Convert to float32
5. Run interface with TFLite
6. Apply Gate 1 and Gate 2
7. Save result and Image
8. Send result to dashboard

To deploy the model on the robot, we convert It to TensorFlow Lite. This pipeline allows for real time on device classification without needing Internet or cloud processing.

Robotics

Embedded Systems & Robotic Components

Summary

- Raspberry Pi 4 as the main processor
- TB6612 motor driver controlling the 4 motors via PWM
- US-100 ultrasonic sensor for obstacle detection
- PiCamera2 for image capture
- A 4×AA battery pack powering motors
- Flask-based web dashboard for robot control

Embedded System Concepts Applied

- GPIO motor control
- PWM speed modulation
- Real time sensing and decision-making
- Multithreading (robot loop + dashboard server)
- Edge AI inference
- Autonomous navigation logic

The Robot Continuously:

- drives forward,
- checks for obstacles,
- captures images,
- performs classification,
- logs detections,
- and resumes navigation

Pinout & Wiring Guide

Universal Motor Driver System

This project now supports **two motor driver types**:

Driver	File Used	Notes
L298N	`motor/l298n.py`	Works with older red dual-H-bridge modules
TB6612	`motor/tb6612.py`	Newer Adafruit-style high-efficiency motor driver

You select the driver in:

`config.py`

```
MOTOR_DRIVER = "TB6612"   # or change to "L298N"
```

Then everywhere in the code, motors are used like:

```
from motor import forward, backward, turn_left, turn_right, stop
```

The rest is automatic — the correct driver is imported internally.

TB6612 Motor Driver Wiring

Motor	Red Wire	Black Wire
Motor A (Left)	A01	A02
Motor B (Left)	A01	A02
Motor C (Right)	B01	B02
Motor D (Right)	B01	B02

L298N Motor Driver Wiring

Motor	Red Wire	Black Wire
Motor A/B (Left)	OUT1	OUT2
Motor C/D (Right)	OUT3	OUT4

Pinout & Wiring Guide

Power System

Battery Pack (4×AA or 6V motor pack)

- Battery **+** → motor driver **VM** (TB6612) or **12V IN** (L298N)
- Battery **–** → motor driver **GND**

Pi and motor driver MUST share ground.
Do NOT power the Pi from the motor driver.

Raspberry Pi Power to Breadboard

Pi Pin	Connects To
5V	Breadboard **5V rail**
GND	Breadboard **GND rail**

US-100 Ultrasonic Sensor Wiring

US-100 Pin	Connects To
VCC	Breadboard **5V rail**
GND	Breadboard **GND rail**
TRIG	Raspberry Pi GPIO **5**
ECHO	Raspberry Pi GPIO **6**

Pinout & Wiring Guide

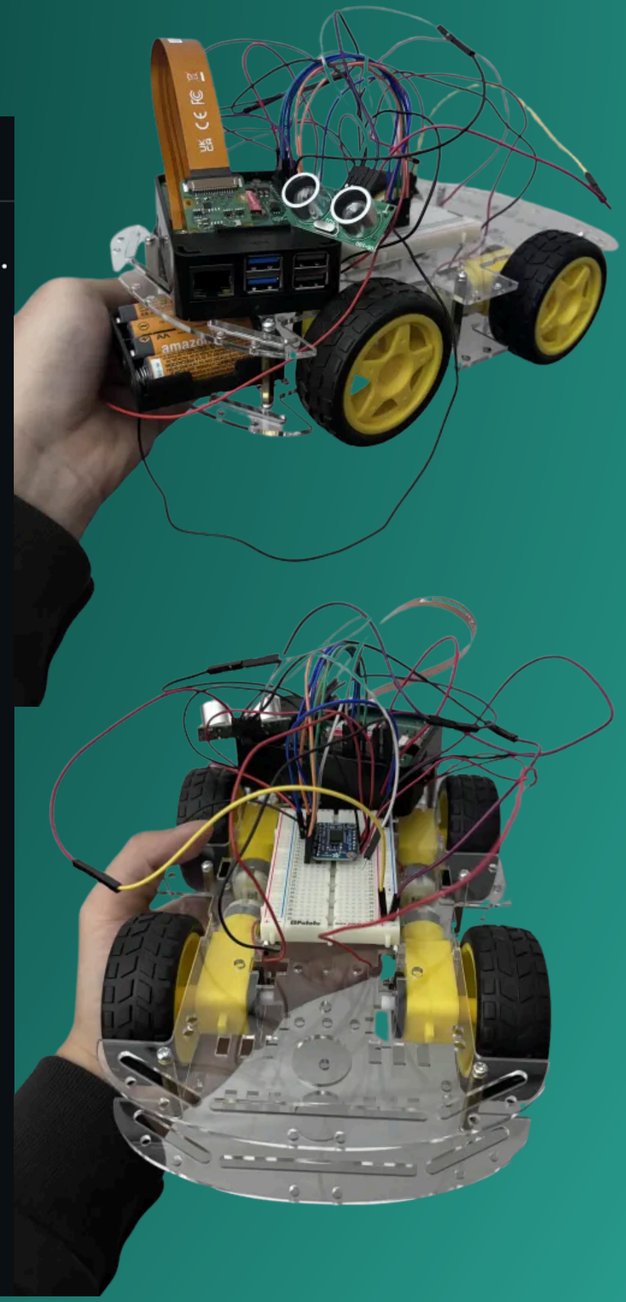
Raspberry Pi GPIO Summary

Both drivers use the **same direction and PWM pins**, but TB6612 requires one additional pin (**STBY**).

Purpose	GPIO Pin	Used By
Motor Left Direction 1	**17**	L298N + TB6612
Motor Left Direction 2	**27**	L298N + TB6612
Motor Right Direction 1	**22**	L298N + TB6612
Motor Right Direction 2	**23**	L298N + TB6612
Left Motor PWM	**12**	ENA (L298N) / PWMA (TB6612)
Right Motor PWM	**13**	ENB (L298N) / PWMB (TB6612)
Motor Driver Standby (STBY)	**24**	**TB6612 only**
Ultrasonic TRIG	**5**	L298N + TB6612
Ultrasonic ECHO	**6**	L298N + TB6612

Notes

- **TB6612 requires STBY** (GPIO 24). This pin *must* be pulled HIGH for motors to operate.
- **L298N does NOT use STBY.** If you're using L298N, ignore GPIO 24 entirely.
- PWM pins (**12** and **13**) must support hardware PWM — these GPIOs do.



Dashboard

Leaf Inspection Dashboard

Status: RUNNING

Start Robot | Stop Robot

Leaf Inspection Dashboard

Status: STOPPED

Start Robot | Stop Robot

Tomato

Potato

Pepper

Robot in Motion