IntelliAgro Drone - User Manual

Title: IntelliAgro Drone: an AI based system for detections of diseases in crops and spraying

1. Introduction

This system automates crop disease monitoring using a Tello drone, deep learning-based object detection (YOLOv8), and BLE-controlled actuation via ESP32-C3. The drone captures real-time video of crops, processes it for disease detection, and, upon identifying a disease like leaf blight, sends a BLE signal to an ESP32-C3 module. This module then controls a servo motor to simulate pesticide spraying.

2. System Components

2.1 Hardware:

- Tello Drone: Captures aerial video of crops.

- Laptop: Hosts Flask server, YOLO model, BLE client.

- ESP32-C3: BLE receiver, controls servo motor.

- Servo Motor: Sprays pesticide upon disease detection.

2.2 Software:

- Python (Flask, OpenCV, Torch, Bleak)

- MicroPython (for ESP32-C3)

- Thonny IDE (for uploading to ESP32)

- YOLOv8 (Model File: OLD.pt)

3. Setup Instructions

3.1 ESP32-C3 Setup:

1. Flash MicroPython firmware to your ESP32-C3.

2. Open Thonny IDE and connect to the board.

3. Upload the following scripts:

= ble_advertising.py: A helper library for Bluetooth Low Energy scanning.

- detection.py: Main program to scan for BLE messages and control the servo motor.

4. Run detection.py in Thonny.

5. It listens for a BLE broadcast from device named ESP-D and interprets incoming messages (e.g., "leaf blight") to rotate the servo to the spray position.

3.2 Laptop (Flask + YOLO) Setup:

1. Connect to the Tello drone's WiFi network.

2. Install all dependencies:

pip install flask opencv-python ultralytics torch bleak

3. Organize your files as follows:

```
project root/
    ├── streamtello10.py
    ├── yolo_model/
    │       └── OLD.pt
    ├── static/
    │       ├── chart.umd.js
    │       └── chartjs_adapter-date-fns.bundle.js
    ├── templates/
    └── index.html
```

4. Run the main Flask server:

    python streamtello.py

5. Open a web browser and visit http://127.0.0.1:5000 to monitor the system.

4. System Working

4.1 Drone Path

- Hardcoded in auto_flight_sequence() function in streamtello.py

- Drone commands include: takeoff, move forward, rotate, land, etc.

4.2 YOLOv8 Detection

- YOLO model used: OLD.pt (placed in yolo_model/ folder)

- Classes Detected: leaf blight, curly leaf blight, fresh

- Uses ultralytics.YOLO() for detection

4.3 BLE Communication

- BLE signal is sent from Flask server using bleak package

- Connects to ESP-D and sends detected disease label via BLE as a UTF-8 string

4.4 ESP32 Control

- Receives disease label over BLE

- If label is leaf blight, servo rotates to 90 degrees and stays there for 3 seconds (spray simulation)

- Then, returns to 0 degrees (neutral position)

- If no disease is detected, the servo remains at 0 degrees

5. Web Dashboard

- The Flask web interface includes:

- Live drone camera feed

- Real-time detection result logging

- Disease chart visualization

- Charting is handled using Chart.js, loaded from:

- static/chart.umd.js: Main Chart.js library for rendering charts.

- static/chartjs_adapter-date-fns.bundle.js: Adapter to allow chart date formatting on time-based X-axis.

- Chart automatically updates to reflect the latest detections

- Control buttons available:

- Start/Stop Drone and video communication

6. Customization

Task	File/Location
Modify drone flight steps	auto_flight_sequence () in streamtello.py
Update YOLO model	Replace OLD.pt in yolo model/
Change BLE device name	Change ESP-D in both detection.py
Adjust servo timing/angle	Edit timing in detection.py (3-second delay)

7. Troubleshooting

Issue	Solution
Drone video not loading	Ensure you're connected to Tello WiFi
YOLO detection not working	Check OLD.pt file and class names
ESP32 not receiving BLE	Verify BLE device name ESP-D is correct
Servo not moving	Confirm wiring and correct GPIO pin
Chart not updating	Ensure JavaScript files are loaded in static/

8. Diagrams

8.1 System Architecture Diagram

[Drone (Tello)] --(WiFi)--> [Laptop (Flask + YOLOv8 + BLE Client)] --(BLE)--> [ESP32-C3] --> [Servo Motor (Spray)]

- Drone: Captures crop images.
- Laptop: Detects disease, sends BLE signals.
- ESP32-C3: Acts on BLE signal, rotates servo.

8.2 File Structure Overview

```
project_root/
├── streamtello.py            # Main control and Flask server
├── yolo_model/
│   └── OLD.pt                # YOLOv8 trained model
├── static/
│   ├── chart.umd.js          # Chart.js main library
│   └── chartjs_adapter-date-fns.bundle.js  # Date adapter for charts
├── templates/
│   └── index.html            # Web UI template
├── detection.py (ESP32)      # BLE message receiver and servo controller
├── ble_advertising.py (ESP32) # BLE helper script
```

9. Conclusion

This smart agriculture project combines deep learning, IoT, and embedded systems to provide automated disease detection and treatment. It demonstrates real-time analysis, BLE-based communication, and servo-based actuation for pesticide spraying. Designed for scalability, this system can help farmers monitor large crop fields efficiently.

End of User Manual