Olive Quality Inspection System Using Edge Impulse

Executive Summary

This project presents an automated olive quality inspection system that leverages Edge Impulse AI, Raspberry Pi 4, and Arduino microcontrollers to enable real-time multi-olive analysis on edge devices. The system processes multiple olives simultaneously in a single camera frame, addressing the bottleneck of single-olive inspection in industrial olive processing factories.

Key Achievement: Successfully deployed machine learning inference on Raspberry Pi 4 running Edge Impulse models with hardware integration via Arduino servo control, achieving real-time performance without cloud dependency.

---

1. Introduction

The project was developed for the HackerEarth Edge AI Hackathon 2025, focusing on practical edge computing solutions. The goal is to demonstrate how edge devices (Raspberry Pi 4) combined with AI models (Edge Impulse) and microcontroller (Arduino) can solve real-world industrial automation challenges.

1.1 Hackathon Context

- **Challenge:** Build an intelligent system using edge devices and Edge Impulse AI
- **Platform:** Edge AI contest by Edge Impulse 2025
- **Team Size:** Individual contributor

---

2. Problem Statement

2.1 Real-World Challenge

Olive processing factories face a critical efficiency bottleneck:

- **Scale:** Millions of olives process through conveyor belts daily
- **Current Method:** Analyzing one olive at a time (manual or single-sensor approach)
- **Limitation:** Single-olive inspection is too slow for high-speed production lines
- **Cost:** Labor-intensive quality control reduces profitability

2.2 Technical Requirements

The solution must address:

1. **Speed:** Process multiple olives per frame in real-time
2. **Edge Computing:** Run inference locally without cloud latency
3. **Cost-Effectiveness:** Use affordable hardware (Raspberry Pi, Arduino)
4. **Scalability:** Support expansion from 2×2 to N×N grid analysis (100+ olives per frame)
5. **Automation:** Integrate with physical sorting mechanisms (servos, conveyors)

---

3. Solution Architecture

3.1 Core Innovation: Grid-Based Parallel Analysis

Instead of processing one olive per camera frame, the system divides each frame into a grid and analyzes all regions simultaneously.

Prototype Implementation:

- **Grid Size:** 2×2 (4 olives per frame)
- **Production Potential:** Scalable to N×N (e.g., 10×10 = 100 olives per frame)

3.2 System Workflow

1. Camera Captures Frame (Multiple Olives)
2. Raspberry Pi 4 Divides Image into 2×2 Grid
3. Edge Impulse Model Classifies Each Region (Good/Bad or Green/Black)
4. Classification Results Sent to Arduino via USB Serial Communication
5. Arduino Controls Servo Motors for Sorting
6. Olives Automatically Separated on Conveyor

3.3 Hardware Components

Component	Model	Purpose
Edge Device	Raspberry Pi 4 (4GB RAM)	Runs Edge Impulse model inference, image processing
Microcontroller	Arduino ESP32	Controls servo motors
Camera	USB Camera Module	Captures images of olives on conveyor
Servos	Servo Motors (8 total)	4 top servos (drop olives) + 4 bottom servos (sort good/bad)
Communication	USB Serial	Direct serial communication between Raspberry Pi and Arduino via USB cable

3.4 Software Stack

Layer	Technology
Model Training	Edge Impulse Studio (Cloud)
Model Deployment	Edge Impulse on Raspberry Pi 4
Image Processing	Image Processing Block, Transfer learning
Communication	USB Serial
Hardware Control	Arduino C/C++ (Serial communication)

---

4. Technical Implementation

4.1 Edge Impulse Model Development

Dataset Collection:

- Captured images of green olives (good) and black olives (bad/ripe)
- Collected labeled images per class
- Images captured under controlled lighting on white background
- Split: 80% training, 20% testing

Model Architecture:

- Model Type: Image Classification (MobileNetV1)
- Input Resolution: 96×96 pixels
- Classes: 2 (Green/Good, Black/Bad)
- Optimization: Quantized for Raspberry Pi 4 deployment
- Output Format: `.eim` file (Edge Impulse model)

Training Results:

- Accuracy: 95%+ on validation set
- Model Size: ~13 MB (efficient for edge deployment)

4.2 Grid-Based Image Processing

Algorithm:

1. Capture full frame from camera
2. Divide frame into 2×2 grid (or N×N)
3. For each grid cell:
   a. Extract ROI (Region of Interest)
   b. Resize to 96×96 pixels
   c. Run Edge Impulse inference
   d. Get classification result (Good/Bad)
4. Aggregate results into 2×2 classification matrix
5. Send binary classification (0/1) to Arduino for each cell
6. Arduino triggers corresponding servo to sort olive

5. Classification Simplification Note

Current Demo Classes:

- **Green = "Good"** (Immature or early harvest)
- **Black = "Bad"** (Fully ripe)

Important Clarification:
In reality, black olives are NOT bad, they are fully ripe and of high quality. For this prototype, we simplified the classification into two easily distinguishable visual classes to focus on the technical architecture and edge deployment challenge.

Real-World Extension:
Production systems could classify into 3+ categories:

- Immature (green)
- Ripe (dark green / early black)
- Fully Ripe (black)
- Defective (damaged, cracked)

The system is **extensible to multiple classes** without architectural changes.

5.1 Qualitative Achievements

✓ Successfully deployed Edge Impulse model on Raspberry Pi 4
✓ Implemented grid-based parallel image processing
✓ Achieved real-time olive classification and sorting
✓ Demonstrated scalability from 2×2 to potential N×N grids
✓ Built complete end-to-end IoT system (Edge Device + Microcontroller)

6. Lessons Learned

6.1 Technical Insights

1. **Edge Device Capability:** Raspberry Pi 4 is powerful enough for real-time computer vision tasks with optimized models (Edge Impulse quantization was key).
2. **Model Quantization:** Quantized models (int8) run 5-10x faster than full-precision on edge devices without significant accuracy loss.
3. **Grid Parallelization:** Dividing frames into regions enables significant throughput improvements—key innovation for scalability.

6.2 Hackathon Experience

- Edge Impulse provides excellent out-of-box ML model deployment
- Raspberry Pi 4 sufficient for most real-time edge AI tasks
- Servo response latency acceptable for conveyor applications
- Project validates that edge computing is production-ready for industrial automation

---

7. Conclusion

This project successfully demonstrates that **edge AI is practical and accessible for real-world industrial automation**.

By combining Raspberry Pi 4, Edge Impulse trained models, Arduino microcontrollers, and direct USB serial communication, we built a scalable, cost-effective olive sorting system that processes multiple items per frame in real-time—directly addressing a real problem in olive processing factories.

Key Takeaway

Edge computing eliminates cloud latency, reduces costs, and enables true real-time automation. The grid-based approach is novel and scales from prototype (2×2) to production (N×N), making this architecture viable for other conveyor belt industries (fruit, nuts, pharmaceuticals, electronics).

The future of industrial automation is on the edge.