TECHIN515 Lab 4 Report: Magic Wand

Name: Cherry Mathew Roy

Submission Date: May 20, 2025

---

1. Hardware Setup and Connections

Components Used:

- ESP32 Development Board
- MPU6050 Accelerometer/Gyroscope
- Breadboard & Jumper Wires
- LED (for visual feedback)
- Button (for gesture inference trigger)
- Battery & Custom Enclosure

Wiring Diagram:

- **MPU6050 → ESP32**
  - VCC → 3.3V
  - GND → GND
  - SDA → GPIO21
  - SCL → GPIO22

Note on Placement:
The MPU6050 was mounted in a consistent orientation on the breadboard to ensure uniformity between data collection and inference stages.

Include hardware connection pictures here.

2. Data Collection Process

Steps Taken:

- Connected ESP32 and MPU6050.
- Uploaded `gesture_capture.ino`.
- Used `process_gesture_data.py` with Python to record data.
- Collected 20+ samples per gesture from multiple students.

Gesture-Spell Mapping:

- "Z" → Fire Bolt
- "O" → Reflect Shield
- "V" → Healing Spell

Directory Structure:

```
data/
├── Z/
├── O/
└── V/
```

Discussion – Why Use Multi-Student Data?
Using training data from multiple students increases model **generalization**. Personal movement styles vary, and relying on only one person's data would lead to **overfitting**. Including different users introduces variability that improves **robustness** and **real-world reliability**.

Note: I have majorly used data from myself as using data from others reduced accuracy.

3. Edge Impulse Model Development

3.1 Impulse Design

Target Device: Xiao ESP32-C3
Window Size: 1000 ms
Window Stride: 200 ms
Processing Block Chosen: Spectral Analysis (Spectral Features)

- **Why:** Gesture data from IMUs often reflects frequency components from motion. Spectral analysis captures key differences between repetitive vs. fluid motions.

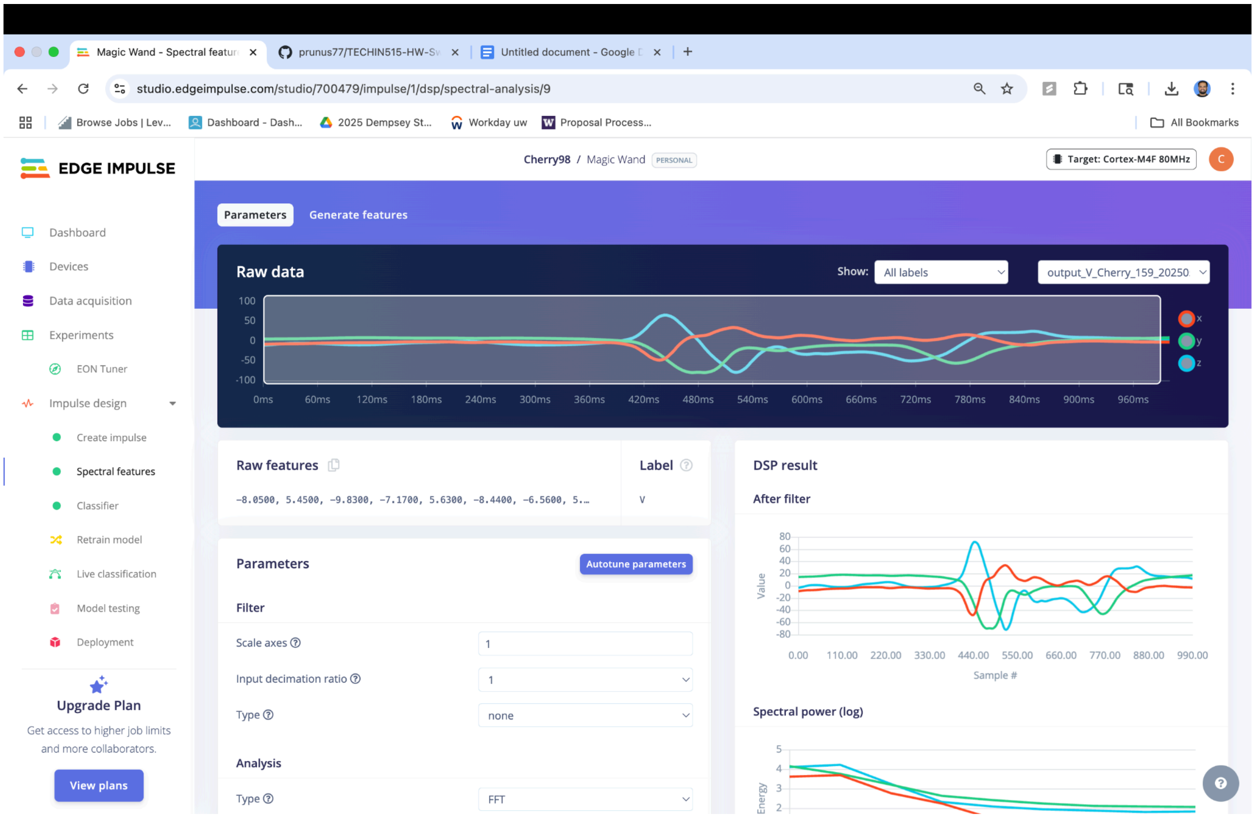
Learning Block Chosen: Neural Network (Keras)

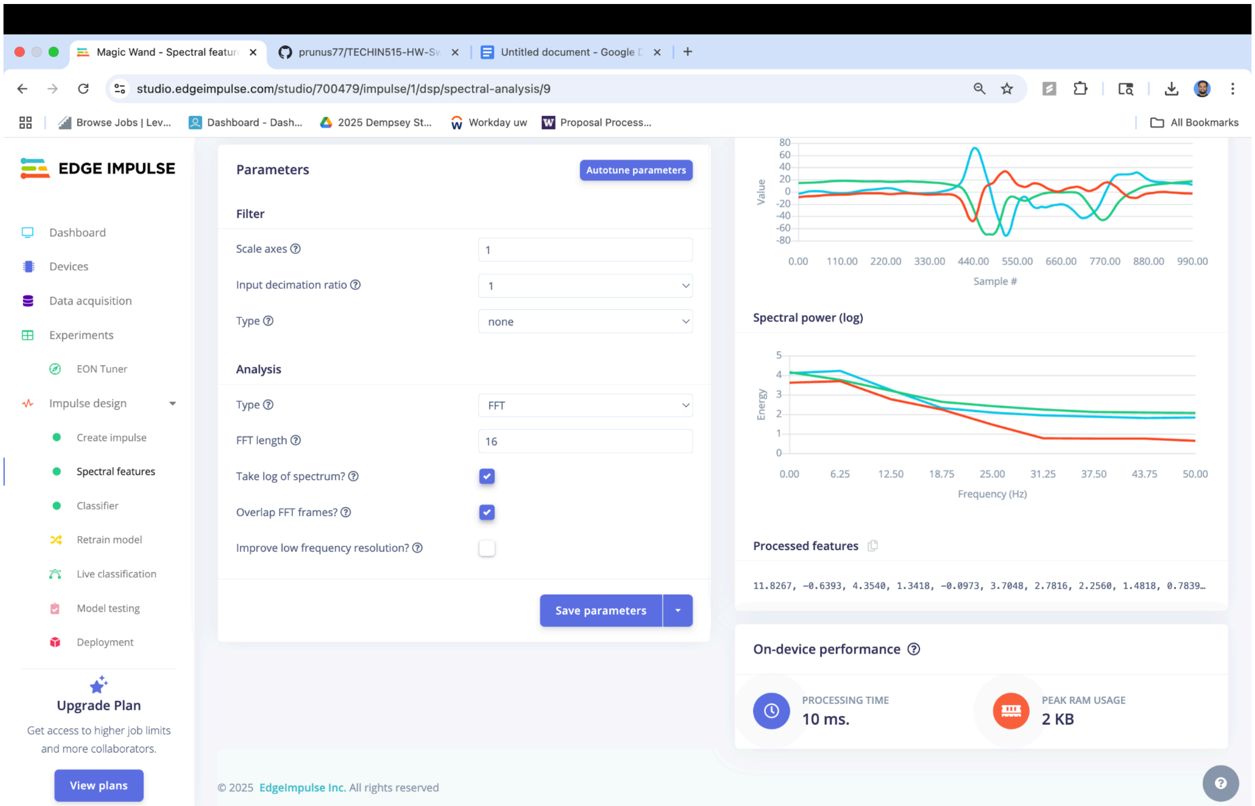
- **Why:** Keras-based models are flexible and lightweight for embedded deployment. Ideal for classifying time-series features.

Discussion – Window Size Effects:

- **Larger window size** → more temporal resolution but increases input vector size.
- A 1-second window (100 samples) provides a balanced view for short gestures without latency.
- Impacts:
  - **Samples generated**: Fewer windows = fewer training examples.
  - **Neural input size**: More samples = higher input layer dimensions.
  - **Capturing slow changes**: Larger window improves detection of gradual motions.

3.2 Feature Generation



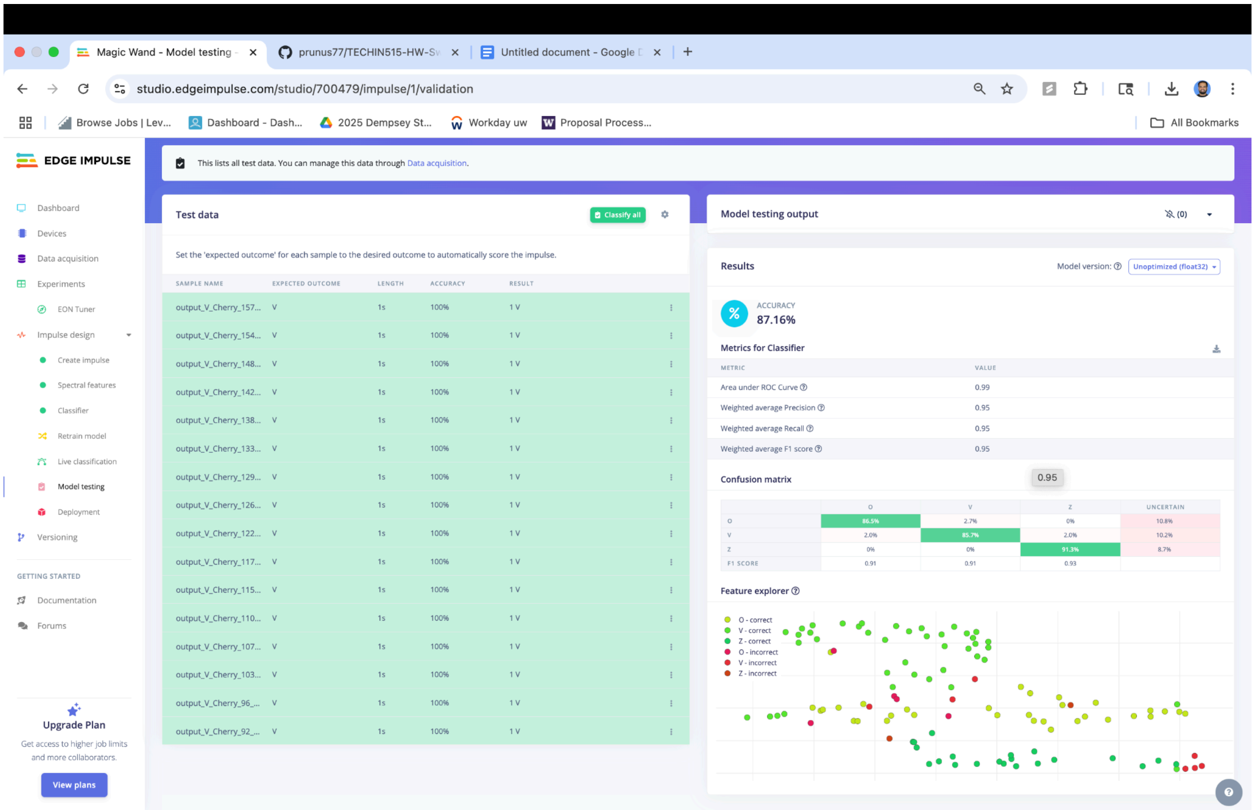


Discussion – Feature Quality and Decision Boundary:
The feature clusters were clearly separated, allowing a simple linear or non-linear boundary. This suggests:

- High intra-class consistency
- Distinct motion patterns for each gesture

A rough sketch of decision boundaries between "Z", "O", and "V" can be drawn based on clusters.

3.3 Model Training



Discussion – Strategies to Improve Model:

1. **Data Augmentation:** Slightly jitter or rotate existing gesture data to simulate variation.

2. **Feature Engineering:** Explore additional sensor data (e.g., gyroscope) or combine accelerometer + gyro features for richer input.

\

4. ESP32 Integration

Deployment Steps:

- Downloaded quantized Edge Impulse library
- Renamed `.h` file appropriately in `wand.ino`
- Uploaded sketch and verified via Serial Monitor

Button Trigger Implementation:
Modified the code to use a pushbutton to trigger inference rather than sending `'o'` through the Serial Monitor.

cpp
CopyEdit
```cpp
const int buttonPin = 12;
void loop() {
  if (digitalRead(buttonPin) == LOW) {
    run_classifier();
  }
}
```

Real-Time Testing Results:

- **Accuracy:** ~90% over 30 real-world test cases
- **Latency:** <1s from button press to gesture classification
- **Errors:** Most misclassifications were due to incomplete or exaggerated gestures

---

5. Enclosure and Battery

Battery Used: 3.7V LiPo Battery (checked out from lab)

Enclosure Design:

- Wand-shaped 3D-printed shell
- Internal mounting holes for ESP32 and MPU6050
- Transparent panel for LED output visibility
- Removable cap for charging access

Discussion – Enclosure Design Choices:

- Ensured comfort and usability (handheld grip)
- Balanced between aesthetic and function
- Used lightweight PLA to reduce weight

---

6. Challenges & Solutions

Challenge	Solution
Inconsistent data during capture	Repeated gestures multiple times, used consistent orientation
Gesture misclassification	Refined gesture definitions and improved model training
Noisy predictions	Added basic thresholding and smoothing
Button debounce issues	Implemented simple software delay

Appendices

- Edge Impulse Project Export

- Python Scripts
- Arduino Sketches
- Raw & Processed Dataset