1. The "Third Hand" Problem (Recording 2-Handed Gestures)

- **The Corner Case:** The original data collection workflow required the user to hold down the `'c'` key on the keyboard to record data. However, for gestures like "Malevolent Shrine" or "Infinite Void" which require **both hands** to be in the air, the user physically ran out of hands to press the key.
- **The Fix:** We implemented an **auto-record logic**. The system checks the `required_hands` attribute for the current label. If `required_hands > 1` and the computer vision system detects a valid two-hand vector, it bypasses the manual key press check and automatically saves the sample IF the hands are detected by mediapipe

2. The "Flickering" UI (Hysteresis)

- **The Corner Case:** Machine learning models output probabilities that fluctuate wildly frame-by-frame (e.g., 89% -> 91% -> 89%). If the threshold was a hard 90%, the "Domain Expansion" effects would strobe on and off rapidly, creating a jarring user experience.
- **The Fix:** We implemented **Hysteresis (State Persistence)**.
    - **Trigger:** The effect only *starts* if confidence is extremely high (>90%).
    - **Sustain:** Once started, a timer (`effect_timer = 30`) keeps the effect active for a set duration, smoothing over minor dips in confidence.
    - **Hard Cancel:** A specific "breaker" logic was added to immediately kill the effect if the model detects "None" or confidence drops below 50%, ensuring the UI feels responsive to intentional stops, not just accidental jitters.

3. Scale Invariance (The "Zoom" Problem)

- **The Corner Case:** A user training the model with their hand close to the camera (producing large coordinate values) would find the model failing when they stepped back (producing small coordinate values), even if the *shape* of the hand was identical.
- **The Fix:** We implemented **feature vector normalization**.
    - **Translation:** All points are re-centered relative to the wrist (Wrist becomes `0,0,0`).
    - **Scaling:** We calculate the distance between the wrist and the middle finger knuckle (a stable reference bone) and divide all coordinates by this value. This ensures a "thumbs up" looks mathematically identical whether it is 10cm or 100cm from the lens.

4. Handling Missing Data (The "One-Handed Null" Crash)

- **The Corner Case:** The system attempts to merge features from both hands into a single vector. If only one hand is visible, the second hand's data is missing, which could cause the array math to crash or the model to receive ragged input shapes.
- **The Fix:** We implemented a "Empty Hand" padder. If a hand is missing, the system injects a zero-filled vector (`[0.0] * 63`) into that slot. This ensures the model always receives a consistent input shape of 129 features, preventing runtime crashes.