RFC: Camera-Agnostic Hand Gesture Recognition System

Status: Draft?

Summary

This RFC proposes a camera-agnostic hand gesture recognition system that uses hand landmark geometry instead of raw images to classify custom static and quasi-static gestures. The system prioritizes reliability, stability, and intentional activation over per-frame accuracy, and separates offline model training (Jupyter) from online runtime inference (Python application).

Motivation

Naive gesture recognition systems that operate on raw RGB frames suffer from:

- Sensitivity to lighting and background
- High false-positive rates
- Poor generalization across users and cameras

This design addresses those issues by:

- Using geometric representations (hand landmarks)
- Treating gesture recognition as a decision system, not just a classifier
- Explicitly supporting gesture rejection (`none`)

---

3. Goals and Non-Goals

Goals

- Camera-agnostic gesture recognition
- Support for single- and two-hand gestures
- Stable gesture triggering under natural hand jitter
- Offline training with reproducible inference behavior

Non-Goals

- End-to-end vision models on RGB
- Complex temporal choreography (e.g., multi-step seals)
- Perfect per-frame classification accuracy

High-Level Architecture

Camera Frames
↓
MediaPipe Hands (landmarks)
↓
Feature Engineering
(normalization + geometry)
↓
Two-Hand Feature Merging
(tuple of 129 features)
↓
Gesture Classifier
(sklearn)
↓
Confidence Estimation
(softmax) eg. 0.90 gojo, 0.05 sukuna, 0.05 unknown
↓
Decision Logic
(thresholds + stability + cooldown)
↓
Gesture Intent Output

Key Design Decisions

Geometry over Pixels

- Use MediaPipe hand landmarks instead of raw images
- Improves robustness to lighting, background, and camera variance

Two Hands = One Pose

- Merge left and right hand features into a single vector
- Include cross-hand geometric features (distance, symmetry)

- Classify **configurations**, not individual hands

Simple Models, Strong Decisions

- Small MLP classifier with ReLU activations -> actually just scikit with rbf-svm will be fine
- No temporal modeling inside the network
- Temporal stability handled at the system level

Explicit Rejection

- Include a `none` / `unknown` class
- Do not force classification when confidence is low

Offline vs Runtime Separation

Offline (Jupyter Notebooks)

- Dataset inspection and cleaning
- Feature validation
- Model training and tuning
- Confidence analysis and threshold selection
- Export model weights and metadata

Runtime (Python Application)

- Real-time MediaPipe inference
- Feature extraction (shared code)
- Model inference only
- Decision logic and gesture triggering

Notebooks are **never** part of the deployed system.

Data and Training Strategy

- Record gesture **sessions**, not isolated frames
- Extract features offline using MediaPipe
- Aggregate short frame windows into stable samples
- Split datasets by recording session, not randomly
- Optimize for separability and confidence behavior, not max accuracy

- Add recording mode for 2 handed gestures where user is not able to access keyboard

---

Decision Logic

- Use confidence thresholds with hysteresis
- Require gesture confidence across multiple frames
- Apply cooldown after successful trigger
- Reject ambiguous or unstable predictions

This converts probabilistic outputs into **deliberate intent**.

Risks and Mitigations

Risk	Mitigation
False positives	Stability windows + rejection
Handedness swaps	Canonical left/right ordering
Overfitting to one user	Multi-user data collection
Gesture ambiguity	Explicit `none` class
Twitchy UX	Cooldowns and temporal gating

Alternatives Considered

- **RGB-only CNNs:** rejected due to fragility and data requirements
- **Template matching:** rejected due to poor scalability
- **Temporal deep models (LSTM/Transformer):** unnecessary for static gestures

---

Future Extensions

- Optional depth (`z`) features
- Dynamic gesture support via windowed aggregation

- Export to ONNX / TorchScript
- Integration with ROS or desktop applications

Decision

Proceed with landmark-based geometric gesture recognition using offline-trained classifiers and runtime decision logic, prioritizing **predictable behavior over raw accuracy**.