

Mushti Motion Classifier Submission Console

Anonymous

Abstract

We present a browser-based hand-motion classifier with an integrated submission console tailored to the ACL Rolling Review process. The system detects a closed-fist (Mushti) gesture from webcam landmarks, classifies vertical wrist motion into steadiness or courage, and exposes all thresholds as live, persistent controls. We pair the live demo with a submission readiness workspace that encodes ARR requirements (anonymity, limitations, checklist compliance, and reviewer obligations) to help authors prepare compliant submissions. The contribution is a lightweight, self-contained web app that links experimental behavior to submission compliance workflows.

1 Introduction

The ACL Rolling Review (ARR) process requires more than technical novelty; it imposes strict formatting, anonymity, checklist, and reviewer responsibility requirements. At the same time, interactive demos are increasingly important for communicating system behavior. We introduce a two-part web application: (1) a real-time hand-motion classifier built on MediaPipe landmarks, and (2) an ACL-focused submission console that tracks readiness and artifacts. The design goal is to keep experimentation fast while ensuring submission requirements are explicit and verifiable.

2 Related Work

Our classifier uses MediaPipe Hand Landmarker for real-time hand tracking. We build on established gesture classification practices that measure landmark distances and movement over time. The submission console reflects ARR and ACL guidelines, which emphasize anonymized review, limitations disclosure, responsible NLP checklists, and reviewer participation.

3 System Overview

The system consists of two views:

- **Live Demo:** Webcam feed, landmark overlay, Mushti detection, motion classification, action log, and detailed metric explanations.
- **Submission Console:** ARR readiness checklist, paper metadata, contribution tagging, ethics and AI-use disclosures, artifact tracking, timeline planning, and exportable summary.

Both views share a single codebase and persistent settings stored in local storage to preserve calibration and planning state across sessions.

4 Mushti Detection

We detect a closed fist by computing ratios of wrist-to-tip and wrist-to-MCP distances for each finger. A finger is considered curled when the ratio is below a threshold. The overall Mushti state is true when a required count of fingers (including thumb) meet their thresholds. Per-finger and thumb thresholds are configurable and persisted.

5 Motion Classification

When a Mushti is detected, we track wrist Y displacement over a sliding time window. If the displacement magnitude exceeds a threshold and sufficient samples are present, we log a label. Direction determines the label: downward motion is steadiness, upward motion is courage. All thresholds and timing parameters are loaded from JSON and can be adjusted live.

6 Submission Console

The console encodes ARR requirements as checklists and metadata fields. It includes:

- Readiness checklist: limitations, template compliance, anonymity, dual-submission check, responsible checklist, and reviewer obligations.
- Metadata: title, paper type, area, deadline, venue, and preprint policy.
- Ethics + AI-use disclosure fields to support checklist and acknowledgements.
- Artifact tracker for anonymized code/data/appendix packaging.
- Timeline planning and exportable submission summary.

This design keeps compliance visible during demo preparation and supports rapid iteration.

7 Implementation

The application is a static web app (HTML/CSS/JS) with MediaPipe Tasks for hand landmarks. Config files (`mushti-requirements.json`, `movements.json`, and `requirements-info.json`) centralize thresholds and metric definitions. Live controls update in-memory settings and persist to local storage; reset restores JSON defaults.

8 Evaluation

We provide a qualitative evaluation through interactive use. The system responds in real time to fist closure and vertical motion; thresholds can be tuned to accommodate user variability. The submission console is evaluated by verifying that each ARR requirement has a direct representation in the UI and can be marked complete.

9 Limitations

The classifier uses simple ratio thresholds and a single camera view; occlusions or unusual hand orientations can reduce accuracy. The evaluation is qualitative and does not provide benchmarked accuracy. The submission console does not guarantee compliance; it reflects user-input state and cannot validate anonymity or formatting automatically.

10 Ethical Considerations

The system operates on local webcam input and does not transmit video off-device. Users should be informed about camera use and may stop the camera at any time. No personal data is stored beyond local browser settings.

11 AI Assistance Disclosure

If generative tools are used for writing or coding, they should be disclosed in the Responsible NLP checklist and acknowledgements. Authors remain responsible for correctness and proper citation.

12 Conclusion

We present a compact, reproducible system that combines a hand-motion classifier with an ACL/ARR submission workflow console. The result is a practical, demo-ready artifact that emphasizes both experimental calibration and submission compliance.