Project Proposal: Touchless Gesture Control Interface for Media Playback

Ali Zekai Deveci
Student Number: 60291

January 2026

1 Topic Selection and Approval

Topic: Touchless Gesture Control Interface for Media Playback

Justification for Class Forum

I propose designing a **Touchless Gesture Control Interface** that allows users to control media playback (such as music or video) using hand gestures captured by a webcam.

Relevance to HCI

This project explores **Natural User Interfaces (NUI)** by replacing traditional mechanical input devices (mouse, keyboard) with computer vision-based interaction. It addresses key HCI concepts such as:

- **Accessibility:** For users with limited mobility.
- **Ergonomics:** Reducing repetitive strain.
- **Intuitive Mapping:** Direct mapping of physical gestures to digital actions.

The system will focus on providing immediate visual feedback to ensure a smooth user experience, which is a core principle of effective interaction design.

2 Initial Project Description

2.1 Project Objective

The objective is to design and implement a desktop application that enables users to control system media volume and playback status without physical contact. The system aims to provide a reliable, low-latency interaction method suitable for scenarios where touching devices is inconvenient (e.g., cooking, dirty hands) or impossible.

2.2 Test Field: Online Players

The primary test field for this application will be **Online Video Players** (specifically YouTube). The application will simulate keyboard shortcuts (e.g., 'k' for Pause, Arrow keys for Volume) to interface directly with the browser-based player, allowing users to control web content seamlessly.

2.3 Input/Output Specification

- **Input:**
  - Video stream from a standard laptop webcam (RGB data).
  - Hand landmarks and gesture recognition data (coordinates of finger joints).
- **Output:**
  - **System Actions:** Media control commands sending simulated keyboard events (Play/Pause, Volume Up/Down).
  - **Visual Feedback:** A GUI overlay on the video feed showing the detected gesture, confidence level, and triggered action (e.g., "Volume Up" text appearing when the user gestures).

2.4 Planned Functionality

1. **Hand Detection & Tracking:** Real-time tracking of hand landmarks using Google's `MediaPipe`.
2. **Gesture Recognition:** Implementing logic to classify states such as:
   - *Open Palm* $\rightarrow$ Play
   - *Closed Fist* $\rightarrow$ Pause (or toggle)
   - *L-Shape / Reverse L-Shape* (Volume Mode) + *Vertical Movement* $\rightarrow$ Volume Up/Down
   - *Pinch* (within Volume Mode) $\rightarrow$ Mute Toggle
3. **Interaction Smoothing:** Algorithms to prevent "jitter" or accidental repeated commands (debouncing and state tracking).
4. **GUI Mode:** A window displaying the camera feed with overlaid skeleton tracking and status indicators.

3 Tools and Environment

- **Programming Language:** Python 3.x
- **Libraries/Frameworks:**
  - **OpenCV:** For image processing and video capture.
  - **MediaPipe:** For efficient, pre-trained hand tracking and landmark detection.
  - **PyAutoGUI:** To interface with the operating system and simulate key presses.
  - **NumPy:** For vector calculations.
- **Hardware:** Standard USB Webcam or built-in Laptop Camera.

4 Constraints and Assumptions

- **Lighting:** The system assumes a reasonably well-lit environment for the camera to detect hands.
- **Single User:** The system is designed to track gestures from a single primary user to avoid conflicting commands.
- **Camera Position:** Assumes the user is sitting within 0.5 to 1.0 meters of the camera.
- **Performance:** Must run at a minimum of 15 FPS to ensure the interaction feels responsive.