**Human Computer Interaction CS449 – CS549 Assignment 5: Development of Gesture-based interaction using Mediapipe**

**Team Members:**

Efe Temur

Efekan Ergün  
Yağız Toprak Işık

#### 

#### **1. Overview of the System**

This project implements a hand gesture recognition system using Mediapipe's hand-tracking capabilities. The system identifies gestures based on hand landmarks and maps them to specific interactions or functionalities. It includes a radial menu for advanced interaction options.

##### **Gestures and Interactions**

* **Pointing Up**: Scrolls the screen upward.
* **Pointing Pinky Finger (any rotation)**: Scrolls the screen downward.
* **Pointing Right**: Moves to the next track which is connected currently.
* **Pointing Left**: Moves to the previous track.
* **Victory Pose**: Activates the radial menu which will include all the options.
* **Closed Fist**: Deactivates the radial menu or clicks the mouse when the menu is inactive.
* **Thumbs Up**: Clicks the mouse.
* **Chill Gesture**: Toggles play/pause functionality in a media player.
* **Radial Menu**: Offers additional options such as "Play/Pause," "Next," "Previous," "Scroll Up," "Scroll Down," "Mute," "Volume Up," and "Volume Down."

The text describing the current action is displayed dynamically above the hand. And it changes when a new gesture is recognized.

#### **2. Implementation Details**

The system uses Mediapipe's hand model to detect **21 key points** on each hand. These points include the wrist and the tips of each finger, along with their positions in 3D space (**x**, **y**, and **z** coordinates). The model is designed to work well even when parts of the hand are hidden or overlapping.

#### **Important Landmarks:**

1. **Wrist (Landmark 0)**: This is the main reference point to calculate how other points are positioned.
2. **Index Finger Tip (Landmark 8)**: Used for recognizing pointing gestures like "Pointing Up" or "Pointing Right."
3. **Thumb Tip (Landmark 4)**: Helps detect gestures like "Thumbs Up" and "Chill."
4. **Pinky Finger Tip (Landmark 20)**: Used for gestures like "Pointing Pinky Finger."
5. **Middle Finger Tip (Landmark 12)**: Works with other points to identify gestures like the "Victory Pose."

#### **How It Works:**

* The system calculates the positions of these key points in 3D (x, y, z).
* It measures the distances between points, including depth (z), to recognize gestures.
* A distance threshold is set to make sure the system is not too sensitive to small movements, improving accuracy.

This approach ensures that gestures are detected consistently and reliably, even when hands are partially hidden or in tricky positions.

### **Radial Menu Implementation**

* The radial menu appears around the wrist when activated.
* Menu options are arranged in a circle, equally spaced for easy selection.
* The "Victory Pose" gesture activates the menu, and hovering over an option for 0.6 seconds confirms the selection.
* A cooldown prevents the same gesture from triggering actions repeatedly.

### **3. Code Organization**

#### **Gesture Recognition:**

* **Function:** classify\_gesture
  + Calculates distances between hand landmarks.
  + Applies thresholds to classify gestures like "Thumbs Up" or "Victory Pose."

#### **Radial Menu:**

* **Function:** draw\_radial\_menu
  + Draws a circular menu around the wrist.
  + Highlights the hovered option dynamically.
* **Function:** detect\_menu\_selection
  + Identifies which menu option the user is pointing at.

#### **Interaction Mapping:**

* Gestures are mapped to keyboard and mouse events using **keyboard** and **pyautogui** libraries.
* For example:
  + "Pointing Up" triggers scrolling up.
  + "Victory Pose" activates the radial menu.

#### **Real-Time Feedback:**

* Action text is dynamically displayed above the hand using wrist coordinates.

For more info on the code you can refer to our github link at the end of the report

### **4. Contribution and GitHub Workflow**

#### **Team Members:**

* **Efe**:
  + Created the initial codebase with MediaPipe integration.
  + Implemented basic gestures and added a radial menu UI.
  + Expanded gesture functionality.
* **Efekan**:
  + Enhanced gesture recognition accuracy and functionality.
  + Improved the UI usability and tested extensively in real-world scenarios like watching reels.
  + Refined and optimized the codebase for better performance.
* **Yağız** :
  + Worked on improving the UI to display the current action connected with gestures.
  + Focused on overall code development and integration.
  + Tested the system with different apps for usability and functionality.

#### **GitHub Repository:**

Our project is hosted on GitHub to ensure efficient collaboration and track contributions. The repository includes:

* Full code documentation.
* A commit history showing each member's contributions.
* Github Link: <https://github.com/Efe-Temur/hand-tracking>

**Sample Video Usage and Brief Explanation:**

<https://www.youtube.com/watch?v=S-7KmjE1bPU>

As it can be seen on the video, the user is trying all the possible gestures for the specific usage such as youtube reels in our example.

The user tries with the radial menu and without it by checking the commands from the note at the bottom left of the screen. By peace sign, user can get into menu and control with open hand all the interface, then he tries all possible signs to get along with the interface and control the environment.

Note: Since the system is recorded on a video recording program when the user muted the sound only the user was notified about it.