

Hackathon Project Phases Template

Project Title: Gesture Based Human Interface Using OpenCV

Team Name: RIZOBES

Team Members:

- Midididdi sushanth srinivas
 - Bysani varsha
 - K Sai yashvardhan raju
 - K Prasanna kumar
-

Phase-1: Brainstorming & Ideation

Objective:

Develop a gesture-based human interface system using OpenCV that allows users to control multimedia playback through predefined hand gestures.

Key Points:

1. Problem Statement:

- Traditional input devices like remotes, keyboards, and mice require physical interaction, which can be inconvenient in certain scenarios.
- Hands-free interaction is essential for accessibility and improving user experience in various applications.

2. Proposed Solution:

- Implement a computer vision-based gesture recognition system using OpenCV.
- Define hand gestures such as thumbs-up, thumbs-down, fist, and open palm to control video playback functions:
 - **Open Palm** → Pause video
 - **Thumbs Up** → Increase volume
 - **Thumbs Down** → Decrease volume
 - **Fist** → Mute video
- Utilize machine learning or image processing techniques to detect and classify gestures in real time.

3. Target Users:

- Users who require hands-free control for accessibility.
- Individuals who want an intuitive way to interact with digital content.
- Media players, smart home applications, and public display systems.

4. **Expected Outcome:**

- A functional prototype that allows users to control video playback through hand gestures.
 - Improved user convenience and accessibility compared to traditional input methods.
-

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the gesture-based control system.

Key Points:

1. Technical Requirements:

- **Programming Language:** Python
- **Frameworks & Libraries:** OpenCV, Mediapipe, Numpy
- **Hardware Requirements:** Webcam for real-time gesture recognition

2. Functional Requirements:

- Capture real-time video input from a webcam.
- Process and recognize predefined hand gestures.
- Map gestures to control media playback actions.
- Ensure system responsiveness with minimal lag.

3. Constraints & Challenges:

- Accuracy in gesture detection under varying lighting conditions.
 - Handling different hand orientations and skin tones.
 - Ensuring real-time performance with minimal latency.
-

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Key Points:**1. System Architecture:**

- Capture live video stream using OpenCV.
- Process frames and extract hand landmarks using Mediapipe.
- Classify gestures based on predefined patterns.
- Trigger corresponding media control actions.

2. User Flow:

- Step 1: Start the application with the webcam enabled.
- Step 2: Perform hand gestures within the camera frame.
- Step 3: The system detects the gesture and executes the corresponding command.
- Step 4: The user can interact continuously without physical input devices.

3. UI/UX Considerations:

- Simple, minimalistic interface displaying detected gestures.
 - Visual feedback when a command is successfully executed.
 - Customizable gesture mapping for user preferences.
-

Phase-4: Project Planning (Agile Methodologies)**Objective:**

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To
Sprint 1	Building project	High	4 hours	Day 1	Team Member 1
Sprint 1	Building project	Medium	3 hours	Day 1	Team Member 2
Sprint 2	Prerequisite	High	4 hours	Day 2	Team Member 3
Sprint 2	File management	High	3 hours	Day 2	Team Member 4

Phase-5: Project Development**Objective:**

Implement core features of the gesture-based control system.

Key Points:

1. Technology Stack Used:

- **Frontend:** Python GUI (Tkinter or PyQt)
- **Backend:** OpenCV, Mediapipe
- **Programming Language:** Python

2. Development Process:

- Implement hand tracking using OpenCV and Mediapipe.
- Develop real-time gesture classification logic.
- Map gestures to corresponding multimedia control functions.
- Optimize for accuracy and low latency.

3. Challenges & Fixes:

- **Challenge:** Low accuracy in gesture detection.
 - **Fix:** Use additional training data and fine-tune the model.
- **Challenge:** System lag due to processing time.
 - **Fix:** Optimize image processing pipeline and reduce resolution if necessary.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the gesture-based interface works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status
TC-001	Functional	Detect 'Thumbs Up' gesture	Volume increases	<input checked="" type="checkbox"/> Passed
TC-002	Functional	Detect 'Thumbs Down' gesture	Volume decreases	<input checked="" type="checkbox"/> Passed
TC-003	Functional	Detect 'Fist' gesture	Video mutes	<input checked="" type="checkbox"/> Passed
TC-004	Functional	Detect 'Open Palm' gesture	Video play/pauses	<input checked="" type="checkbox"/> Passed
TC-005	Performance	Response time under 200ms	Gesture recognition is real-time	<input type="checkbox"/> Needs Optimization

TC-006	UI Testing	Show detected gestures on screen	Correctly displayed	<input checked="" type="checkbox"/> Passed
--------	------------	----------------------------------	---------------------	--

Final Submission

1. Project Report (Based on this template)
 2. Demo Video (3-5 Minutes)
 3. GitHub/Code Repository Link
 4. Presentation
-

This revised document reflects the project goals, objectives, and implementation details for "Gesture Based Human Interface Using OpenCV." Let me know if you need any refinements!