

# **Hand Gesture MediaPlayer**

## **A PROJECT REPORT**

*Submitted by*

Vishal Dolasia (23BHI10079)

**BACHELORS OF TECHNOLOGY**

**COMPUTING SCIENCE & ENGINEERING  
(HEALTH INFORMATICS)**



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING AND  
ARTIFICIAL INTELLIGENCE  
VIT BHOPAL UNIVERSITY  
KOTHRI KALAN, SEHORE  
MADHYA PRADESH - 466114**

NOVEMBER 2025

**VIT BHOPAL UNIVERSITY, KOTHRI KALAN, SEHORE  
MADHYA PRADESH – 466114**

# **ABSTRACT**

The Hand Gesture Control System is a real-time computer vision application that enables users to interact with their computers using simple hand gestures captured via a standard webcam. By replacing traditional physical input devices, this project aims to create an intuitive, contactless interface, enhancing accessibility and usability—particularly in environments such as presentations, interactive installations, or for individuals with physical limitations.

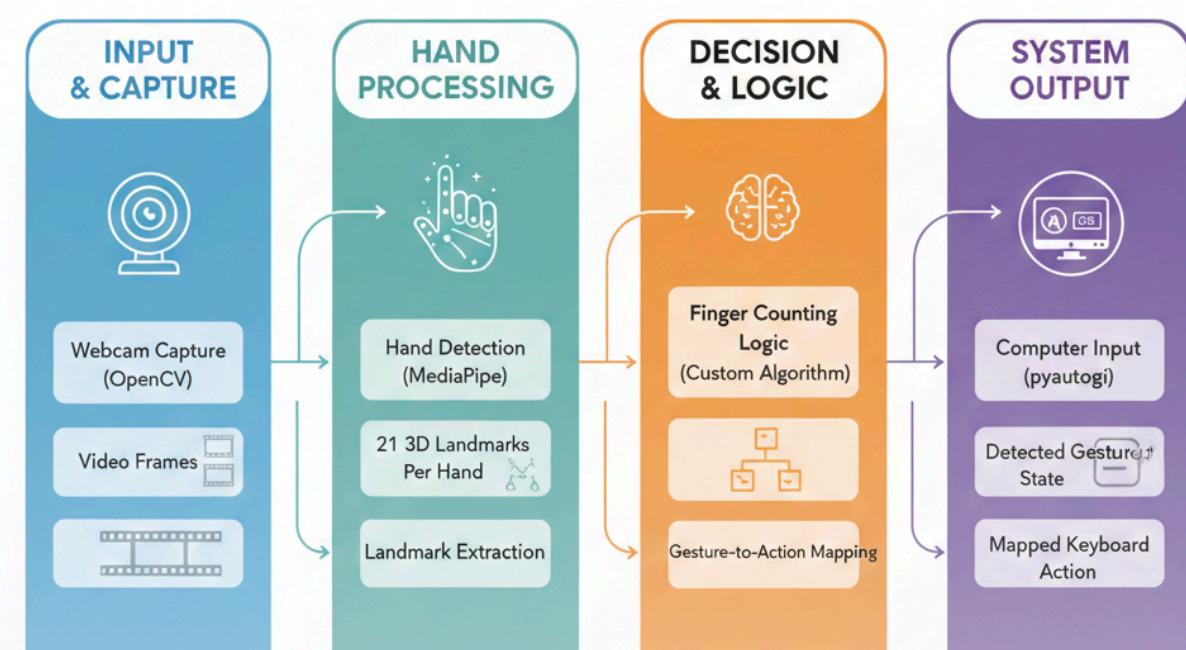
# **INTRODUCTION**

Traditional computer control mechanisms rely heavily on physical devices like keyboards and mice. However, there is a growing demand for hands-free alternatives that are both convenient and accessible. This project addresses this need by leveraging modern computer vision and gesture recognition, allowing users to perform basic control actions efficiently using only hand gestures.

# System Architecture

The Hand Gesture Control System follows a sequential data processing pipeline:

- **Webcam Capture:** A live video stream is acquired using OpenCV.
- **Hand Detection:** Google's MediaPipe framework detects hands and extracts precise key points (21 landmarks per hand).
- **Landmark Extraction:** The model outputs 3D positional data for each landmark.
- **Finger Counting Logic:** A custom algorithm analyzes the landmarks to determine which fingers are extended.
- **Gesture-to-Action Mapping:** each detected gesture is mapped to a keyboard action using the **pyautogui** library.
- **Computer Input:** **pyautogui** triggers corresponding keyboard actions on the computer.



# Core Technologies

- **Python:** Central scripting language for the project implementation.
- **OpenCV:** Handles webcam access and frame display in real time.
- **MediaPipe Hands:** Provides high-accuracy hand keypoint estimation for gesture tracking.
- **pyautogui:** Automates keyboard control based on recognized gestures

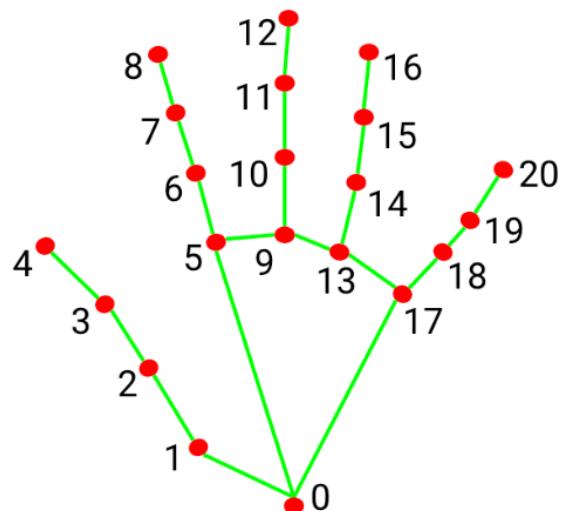
## Finger Counting Algorithm

The finger counting procedure is based on the analysis of MediaPipe's keypoint data:

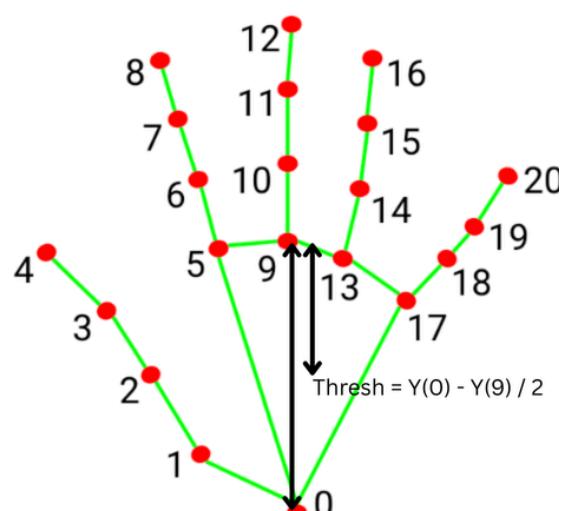
- **Threshold Calculation:** For non-thumb fingers, a dynamic "thresh" value is derived from the y-coordinates of the wrist and middle finger MCP.

$$\text{thresh} = \frac{(y_0 \times 100 - y_9 \times 100)}{2}$$

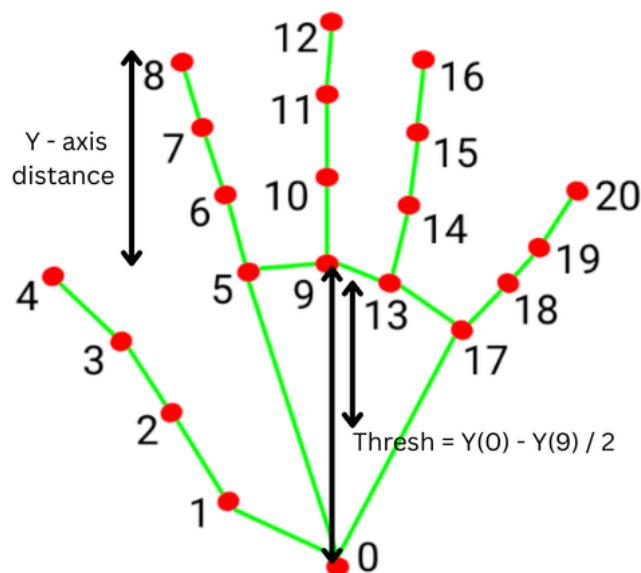
- **Non-Thumb Fingers:** Each finger is considered "up" if its MCP-to-tip vertical distance exceeds the threshold:
  - Index:  $y5 - y8 > \text{thresh}$
  - Middle:  $y9 - y12 > \text{thresh}$
  - Ring:  $y13 - y16 > \text{thresh}$
  - Pinky:  $y17 - y20 > \text{thresh}$



**Fig:- Landmark**

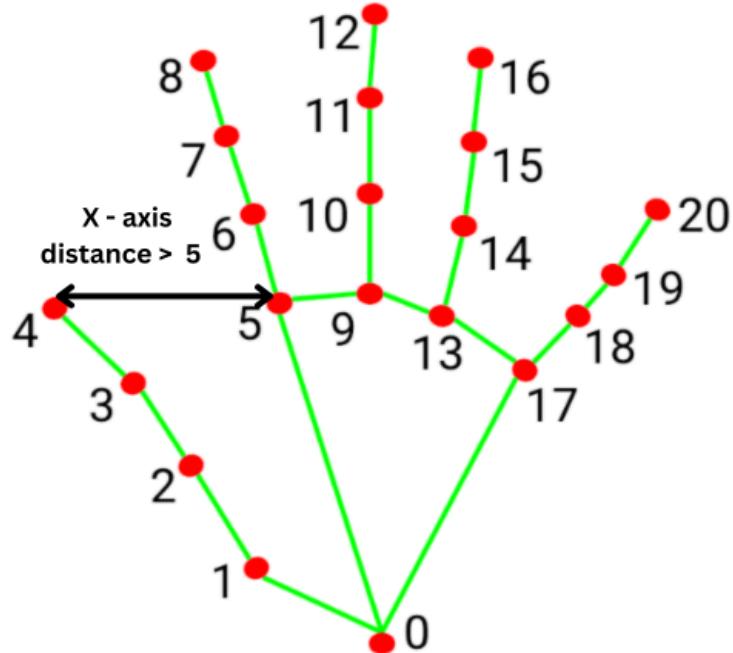


**Fig:- Threshold Value**



**Fig:- Y-Axis Distance**

- **Thumb:** The thumb's status uses the horizontal distance between the Index MCP and Thumb tip. Thumb is up if:  
 $x5 - x4 > 5$



**Fig:- X-Axis Distance**

## Gesture Mapping

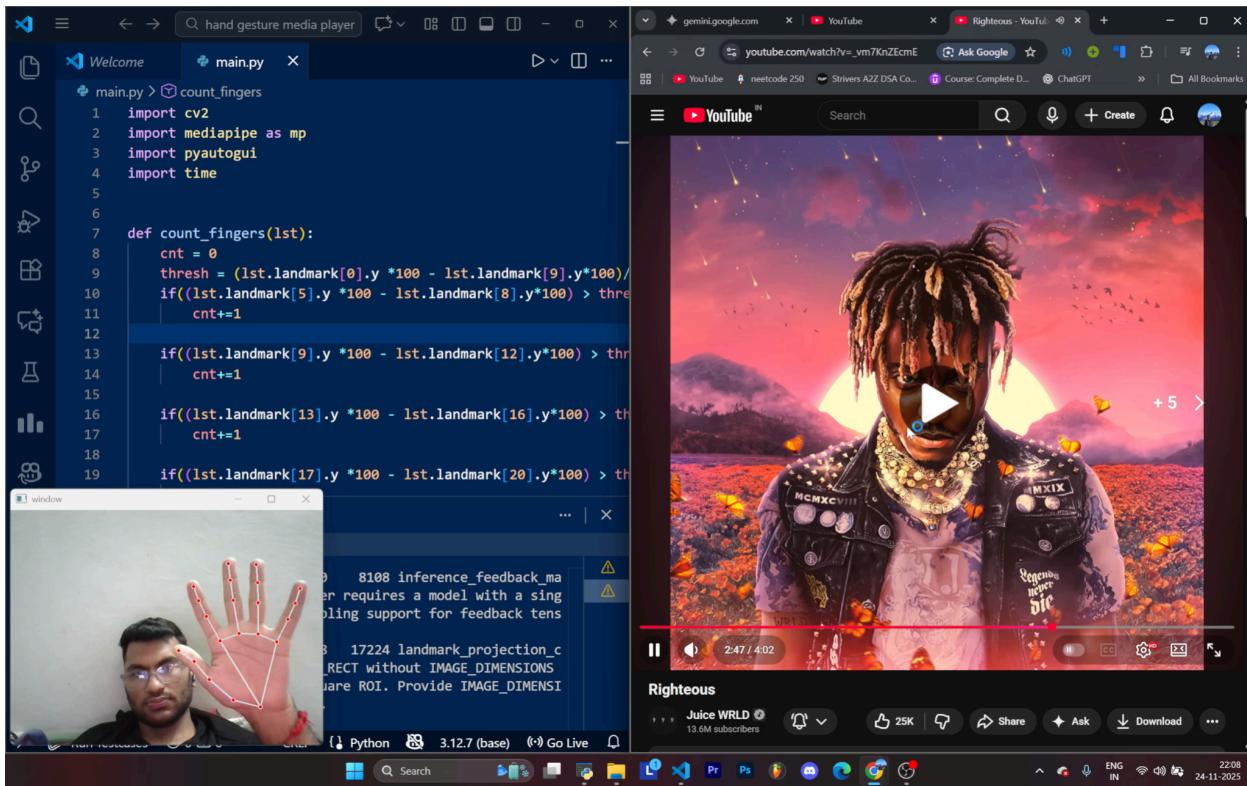
Finger Count	Key Mapped	Application
1	right	Navigation
2	left	Navigation
3	up	Navigation
4	down	Navigation
5	space	Select/Action

## Challenges and Solutions

ID	Challenge	Cause/Description	Solution
1	Gesture Flicker/Inaccuracy	Quick transitions caused unstable gesture recognition	Added a debouncing mechanism using time library (action triggers only if held > 0.2s)
2	Repetitive Key Presses	Loop spams input if finger count unchanged	Implemented state management with previous count to trigger only on change

## Results and Impact

The Hand Gesture Control System provides a functional, hassle-free input method for basic computer control. By integrating debouncing and state management, the system addresses common challenges in gesture recognition, ensuring stable and predictable performance.



Demo Video :- <https://youtu.be/LLp23ujYa40>

## Future Scope

- Supporting simultaneous two-hand gestures for more advanced controls.
- Incorporating dynamic gesture recognition through machine learning models for complex patterns (e.g., swipes, circles).
- Developing a user-friendly graphical interface for custom gesture and key mappings