

## VIRTUAL CALCULATOR USING AI&ML

A.Thanmay Teja Reddy<sup>1</sup>, D.Ruthik Reddy<sup>2</sup>, D.Arunmai<sup>3</sup>,E.Keerthi Reddy<sup>4</sup>,CH.Subba Reddy<sup>5</sup>

<sup>1,2,3,4</sup>UG Students,Department of Computer Science and Engineering ,JBREC,Hyderabad,India

<sup>5</sup>Associate Professor,Department of Computer Science and Engineering ,JBREC,Hyderabad,India

\*\*\*

**Abstract** - In today's digital era, there is an increasing demand for contactless and efficient human-computer interactions. Traditional calculators require manual input, which may not be practical in situations where touch-free operation is essential, such as in healthcare, laboratories, or public systems. A virtual calculator addresses these needs by enabling gesture-based calculations through AI and machine learning technologies. **Problem Definition:** There is a need for an intelligent system that allows users to perform mathematical calculations without physical contact. Conventional calculators are limited in accessibility for people with disabilities and do not cater to the growing demand for touchless digital interfaces. **Objective:** The objective of this project is to develop an AI-powered virtual calculator capable of accurately detecting and interpreting finger movements in real time, enabling seamless, touch-free mathematical operations. **Proposed Solution:** This project integrates Artificial Intelligence (AI), Machine Learning (ML), and computer vision technologies to track hand and finger gestures through a camera. Using frameworks like MediaPipe and OpenCV, gestures are processed to recognize numerical inputs and operations, while a dynamic virtual interface displays the process and results instantly. The system ensures ease of use, high accuracy, and responsiveness. **Deliverable:** The final deliverable includes a functional virtual calculator prototype demonstrating real-time gesture recognition, source code with documentation, and a comprehensive project report and presentation for submission.

**Key Words:** Virtual Calculator,Machine Learning,Open cv,MediaPipe,Hand Tracking,Image Processing

### 1. INTRODUCTION

The primary objective of this project is to design and develop an AI-powered virtual calculator that enables users to perform mathematical calculations through hand gestures without any physical contact. The system utilizes computer vision and machine learning techniques to detect, track, and interpret finger movements in real time using a webcam or camera.

Another important objective is to ensure high accuracy and fast response by recognizing different hand gestures as numerical inputs and arithmetic operations. By integrating advanced frameworks like MediaPipe and OpenCV, the system provides a smooth and interactive user experience

with a dynamic virtual interface that displays inputs and results instantly.

The project also aims to improve accessibility for users with physical disabilities who may find traditional input devices difficult to use. It supports touchless interaction, making it highly suitable for environments such as hospitals, laboratories, and public systems where hygiene and safety are important.

Furthermore, the objective includes promoting modern human-computer interaction techniques by replacing conventional input methods with intelligent gesture-based systems. The project also focuses on scalability and future enhancements, such as adding support for complex mathematical operations, multi-user interaction, and integration with other smart systems.

### 1.1 Problem Statement

Traditional calculators and input systems require physical interaction, which may not always be convenient or hygienic. These systems also lack natural and intuitive ways of interaction.

To address this, the project aims to develop a virtual calculator that allows users to perform calculations using hand gestures through a webcam, enabling a contactless and more interactive user experience using AI and computer vision techniques.

### 1.2 Proposed System

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

### 2. LITERATURE REVIEW

Recent advancements in Artificial Intelligence and computer vision have led to the development of gesture-based interaction systems. Various studies have explored the use of hand gesture recognition for controlling digital

applications without physical contact. Technologies such as OpenCV and MediaPipe have been widely used for real-time hand tracking and landmark detection due to their accuracy and efficiency.

Previous research highlights the use of vision-based models to replace traditional input devices, enabling more natural human-computer interaction. Gesture-controlled systems have been successfully applied in areas like virtual keyboards, gaming, and smart interfaces. These studies demonstrate that hand gesture recognition is a reliable and effective approach for developing contactless applications.

Based on these developments, this project implements a virtual calculator using real-time hand gesture detection, contributing to the growing field of touchless and interactive systems.

### 3. METHODOLOGY

The system is developed using Python by integrating OpenCV and MediaPipe for real-time hand gesture recognition. A webcam captures live video, and MediaPipe detects and tracks hand landmarks such as fingertips. Based on finger positions and movements, gestures are identified and mapped to calculator inputs. OpenCV is used to display the virtual calculator interface and process user interactions. When the user taps virtual buttons using finger gestures, the system interprets the input and performs the corresponding arithmetic operations, displaying the result instantly.

#### 3.1 System Architecture

The system architecture consists of a webcam input module, hand detection module, gesture recognition module, and output display module. The webcam captures real-time video, which is processed using MediaPipe to detect hand landmarks. These landmarks are analyzed to recognize finger gestures, which are then mapped to specific calculator functions. OpenCV is used to generate the virtual calculator interface and display the results. All components work together in a sequential manner to provide a smooth and contactless user interaction experience.

The system is a desktop-based application that uses a webcam to capture real-time video input, processed using OpenCV. MediaPipe is used to detect and track hand landmarks for recognizing finger gestures. These gestures are mapped to calculator inputs, and the system performs the required operations. The results are displayed on a virtual calculator interface, ensuring a smooth and contactless user experience.

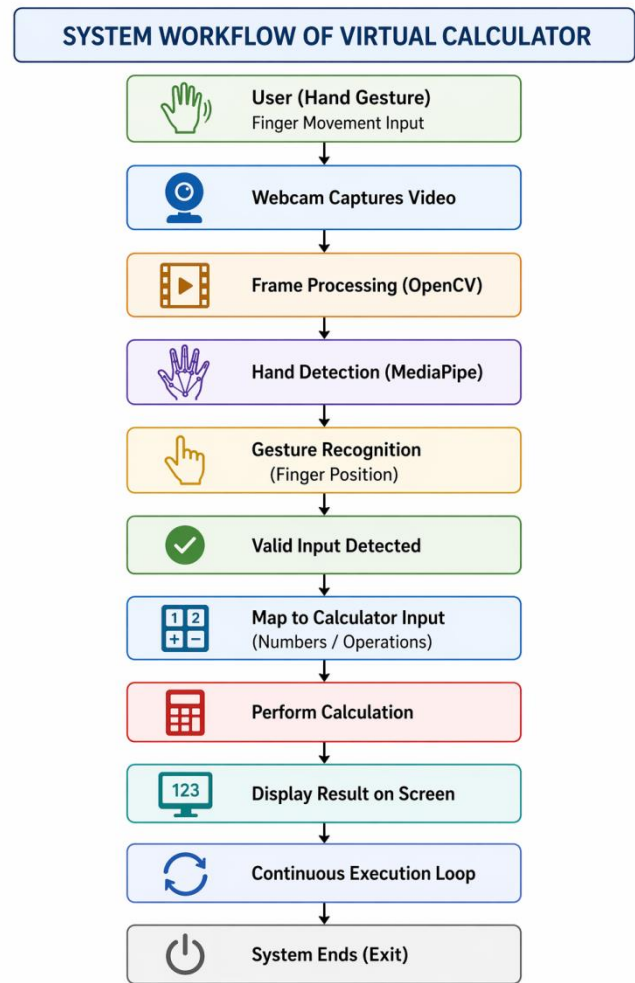


Fig -1: System Architecture

#### 3.3 Algorithm

The system starts by initializing the webcam and required libraries such as OpenCV and MediaPipe. It captures video frames continuously and detects the hand using MediaPipe. The hand landmarks are then identified, and fingertip positions are tracked. Based on these positions, gestures are recognized and validated. The detected gestures are mapped to corresponding calculator inputs, and the system performs the required arithmetic operations. Finally, the result is displayed on the virtual calculator interface, and the process repeats continuously until the program is terminated.

##### Algorithm-1

**Input:** Hand gesture input (finger positions), Webcam video frames

**Output:** Calculator input selection, Displayed result on screen

**Step 1:** Initialize the system and activate the webcam.

**Step 2:** Continuously capture video frames.  
**Step 3:** Process each frame using OpenCV.  
**Step 4:** Detect hand using MediaPipe.  
**Step 5:** Identify and track hand landmarks (fingertips).  
**Step 6:** Analyze finger positions to recognize gestures.  
**Step 7:** Check if the gesture matches predefined calculator inputs.  
**Step 8:** If valid, map the gesture to a specific number or operation.  
**Step 9:** Perform the corresponding arithmetic calculation.  
**Step 10:** Display input and result on the virtual calculator interface.  
**Step 11:** Repeat the process until the system is terminated.

### 3.4 System Workflow

The system begins by activating the webcam to capture real-time video input. Each frame is processed using OpenCV, and MediaPipe is used to detect and track hand landmarks. The system then analyzes finger positions to recognize specific gestures. Once a valid gesture is detected, it is mapped to a corresponding calculator input such as numbers or operations. The system performs the required calculation and displays the result on the virtual calculator interface. This workflow continues in a loop, ensuring smooth and real-time contactless interaction.

The system starts by activating the webcam to capture real-time hand gesture input from the user. Each video frame is processed using OpenCV to ensure smooth and accurate image handling. MediaPipe is then used to detect the hand and track key landmarks such as fingertips. Based on these landmarks, the system analyzes finger positions to recognize specific gestures. Once a valid gesture is identified, it is converted into a corresponding calculator input like numbers or arithmetic operations. The system then performs the required calculation and displays the result on a virtual calculator interface. This entire process runs continuously in a loop, enabling seamless and real-time contactless interaction until the user exits the system.

This presents the system workflow: User performs hand gestures → Webcam captures real-time video → OpenCV processes video frames → MediaPipe detects hand and tracks landmarks → System recognizes finger positions → Valid gesture is identified → Gesture is mapped to calculator input → System performs arithmetic operation → Result is displayed on the virtual calculator interface → Process repeats continuously until the system is exited.

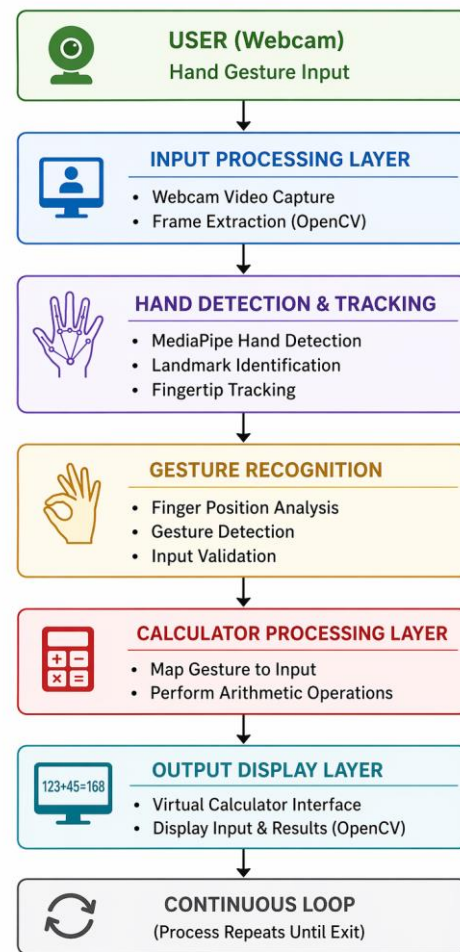


Fig - 2: Workflow

## 4. Implementation and Results

The system is implemented using Python by integrating OpenCV and MediaPipe for real-time hand gesture recognition. The webcam captures live video input, and MediaPipe is used to detect and track hand landmarks such as fingertips. OpenCV processes the video frames and displays a virtual calculator interface on the screen. Based on finger movements and positions, gestures are identified and mapped to corresponding calculator inputs.

The system produces accurate and real-time results by performing arithmetic operations based on the detected gestures. It provides a smooth and responsive user experience without requiring physical contact. The results demonstrate that the system effectively recognizes hand gestures and performs calculations efficiently, validating the practicality of the proposed approach.

### 4.1 Dataset Description

This project does not use a traditional pre-collected dataset, as it is based on real-time hand gesture detection. The input data is obtained dynamically through a webcam,

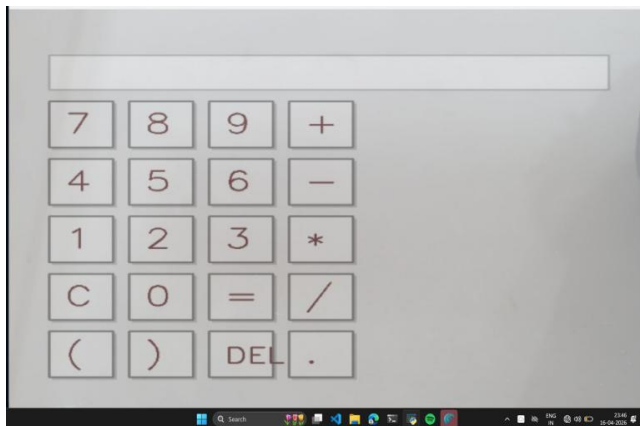
which continuously captures video frames of the user's hand movements. MediaPipe processes these frames to extract hand landmarks, including fingertip positions and joint coordinates.

These detected landmarks act as the input data for gesture recognition. The system relies on predefined gesture patterns rather than trained machine learning models, making it lightweight and efficient. This real-time data approach eliminates the need for large datasets while still ensuring accurate and responsive performance.

## 4.2 System Interface and Outputs

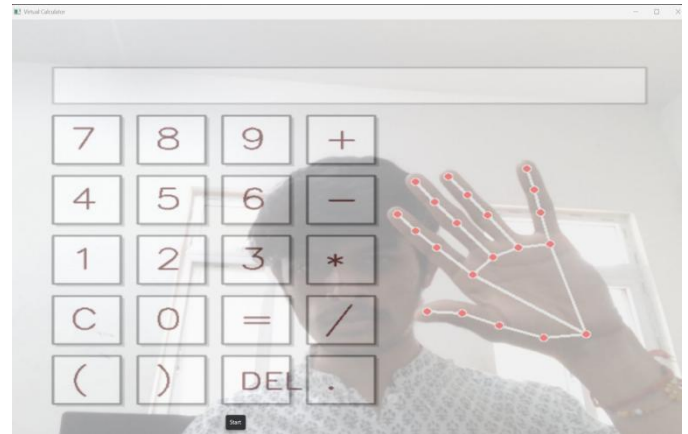
The system interface consists of a virtual calculator displayed on the screen using OpenCV, designed similar to a standard calculator for easy interaction. It includes buttons for numbers, arithmetic operations, and control functions, allowing users to perform calculations using hand gestures.

The output is displayed in real time on the interface as the user provides input through gestures. The system accurately processes the inputs and shows the corresponding results instantly, ensuring a smooth and responsive contactless user experience. The virtual calculator interface used in the system is shown below.



**Fig 1:** Virtual Calculator Interface

The figure shows the basic layout of the virtual calculator interface developed using OpenCV. It consists of buttons for numbers, arithmetic operations, and control functions such as clear and delete. The interface is designed similar to a standard calculator for easy understanding and user interaction.



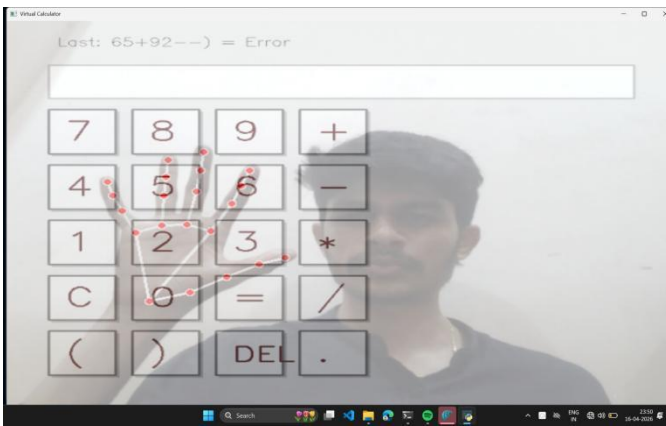
**Fig 2:** Hand Gesture Detection

The figure illustrates the hand gesture detection process using MediaPipe. The system tracks hand landmarks and fingertip positions in real time. These detected points are used to identify user gestures, which act as inputs for selecting buttons on the virtual calculator.



**Fig 3:** Calculation Output

The figure shows the system performing a calculation based on user gestures. The entered values and the computed result are displayed on the screen in real time. This demonstrates the accuracy and responsiveness of the system in executing arithmetic operations.



**Fig 4:** Error Handling

The figure represents a scenario where an invalid input or incorrect sequence of operations is detected. The system displays an error message, ensuring that the user is informed about improper inputs and improving the reliability of the application.

### 4.3 Performance Analysis

The performance of the system is evaluated based on accuracy, response time, and usability. The system demonstrates good accuracy in detecting hand gestures and mapping them to corresponding calculator inputs under proper lighting conditions. MediaPipe efficiently tracks hand landmarks, enabling precise gesture recognition.

The response time of the system is fast, as it processes video frames in real time and displays results instantly. The interface is user-friendly and easy to operate, providing a smooth contactless experience. However, performance may vary slightly due to factors such as lighting conditions, background noise, and camera quality. Overall, the system performs efficiently and meets the expected objectives.

### 4. CONCLUSION

The Virtual Calculator using AI and ML successfully demonstrates a contactless method of performing arithmetic operations using hand gestures. By integrating OpenCV and MediaPipe, the system efficiently detects hand movements and maps them to calculator inputs in real time. The project provides a user-friendly and interactive interface, eliminating the need for physical input devices.

The results show that the system is accurate, responsive, and practical for real-world applications, especially in environments where touchless interaction is preferred. Overall, the project highlights the potential of computer

vision in enhancing human-computer interaction and developing innovative solutions.

### 6. REFERENCES

- Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.
- OpenCV Official Documentation. Available: <https://docs.opencv.org/>
- MediaPipe by Google. Available: <https://developers.google.com/mediapipe> LinkedIn research
- Python, OpenCV, Mediapipe, NumPy, DeepLearning Documentations.