

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI – 590 018,**

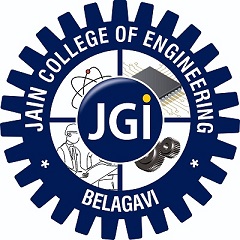
**KARNATAKA**



**Shri Bhagwan Mahaveer Jain Educational & Cultural Trust ®**

**JAIN COLLEGE OF ENGINEERING,**

**BELAGAVI**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**FINAL YEAR**

**(2023 – 2024)**

**PROJECT REPORT**

***On***

**"AI Based Virtual Mouse Using Python"**

**PROJECT GUIDE**

**Prof. S. B. Hugar**

Assistant Prof. Dept. of E&CE

JCE, Belagavi

**PROJECT MEMBER**

Mr. Kiran Karadi 2JI20EC054

Mr. Kiran Patil 2JI20EC055

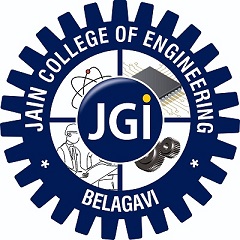
Mr. Akshay Magadum 2JI20EC013

Ms. Devika Kocheri 2JI20EC039

**Shri Bhagwan Mahaveer Jain Educational & Cultural Trust ®**

**JAIN COLLEGE OF ENGINEERING, BELAGAVI**

**T.S.NAGAR, MACHHE – 590014, KARNATAKA**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**CERTIFICATE**

This is to certify that the **Project Work** entitled **“AI Based Virtual Mouse Using Python”** carried out by **Mr. Kiran K (2JI20EC054), Mr. Kiran P (2JI20EC055),   
Mr. Akshay M (2JI20EC013), Ms. Devika K (2JI20EC039)** are bonafide students of **Department of Electronics and Communication Engineering, Jain College of Engineering, Belagavi,** in partial fulfilment for the award of **Bachelor of Engineering** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2023-2024**. It is certified that all corrections/suggestions indicated for project assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

**Prof. S. B. Hugar Prof. V. R. Bagali Dr. J. Shivakumar**

Project Guide HOD, Dept. of E&CE Principal & Director

Dept. of E&CE, JCE BGM JCE, Belagavi JCE, Belagavi

Name of the examiners Signature with date

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARATION**

We **Mr. Kiran K (2JI20EC054), Mr. Kiran P (2JI20EC055), Mr. Akshay M (2JI20EC013), Ms. Devika K (2JI20EC039)** students of **8th semester B.E. Electronics & Communication Engineering, Jain College of Engineering, Belagavi** hereby declare that the dissertation entitled **“AI Based Virtual Mouse Using Python”** has been carried out in a batch and submitted in the partial fulfilment of the requirement for the award of Bachelor’s Degree in Electronics & Communication Engineering under Visvesvaraya Technological University, Belagavi during the academic year **2023 – 24.**

**Name USN Signature**

Mr. Kiran Karadi 2JI20EC054 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Kiran Patil 2JI20EC055 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Akshay Magadum 2JI20EC013 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ms. Devika Kocheri 2JI20EC039 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Place : Belagavi**

**Date :**

 Jain College of Engineering, Belagavi

Dept. of Electronics and Communication Engineering

**Vision of Institute**

"To be a university as a resource of solution to diverse challenges of society by nurturing innovation, research & entrepreneurship through value based education."

**Mission of Institute**

* To provide work culture that facilitates effective teaching-learning process and lifelong learning skills.
* To promote innovation, collaboration and leadership through best practices.
* To foster industry-institute interaction resulting in entrepreneurship skills and employment opportunities.

**Vision of Department**

*“To achieve excellence in education and research for developing globally competent, ethically sound Electronics & Communication Engineers”.*

**Mission of Department**

1. To provide conductive environment through structured student centric, teaching learning process.
2. To nurture needs of society by infusing scientific temper in students and to grow as a centre of excellence with efficient industry-institute interaction.
3. To inculcate self learning skills, entrepreneurial ability and professional ethics.

**Program Educational Objectives (PEO’s)**

1. Graduates will be able to contemplate real-time social problems and deliver efficient solutions.
2. Graduates will be able to lead and succeed in professional careers.
3. Graduates will contribute through research and enterpreneurship.

**Program Specific Outcomes (PSO’s)**

Graduates in the UG program in Electronics and communication engineering will be able to

1. Design, verify and develop analog and digital systems by using state of art technology to contribute to the societal needs..
2. Apply knowledge in various domain of IoT, real time systems, communication systems, VLSI and embedded systems, image and signal processing using hardware and software tools.

Jain College of Engineering, Belagavi

Dept. of Electronics and Communication Engineering

**PROGRAM OUTCOME’S (PO’S)**

**Engineering Graduates will be able to:**

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineeringfundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complexengineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems anddesign system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and researchmethods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modernengineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assesssocietal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutionsin societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader indiverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineeringcommunity and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the **engineering and management principles and apply these to one’s own work, as a member and** leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage inindependent and life-long learning in the broadest context of technological change.

Jain College of Engineering, Belagavi

Dept. of Electronics and Communication Engineering

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO-PO/PSO Mapping:**  **L1: Remembering L2: Understanding L3: Applying L4: Analyzing L5: Evaluating L6: Creating**   |  |  |  | | --- | --- | --- | | **Course Outcomes** | **Description** | **Bloom’s Cognitive level** | | 18ECP83.01 | **Understand the basic concepts and broad principles of industrial projects.** | **L3** | | 18ECP83.02 | **Apply the theoretical concepts to solve industrial problems with team work and multidisciplinary approach.** | **L3** | | 18ECP83.03 | **Get capable of self education and clearly understand the values of achieving perfection in project implementation and completion.** | **L3** | | 18ECP83.04 | **Demonstrate professionalism with ethics; present effective communication skills relate engineering issues to broader societal context.** | **L3** | | 18ECP83.05 | **Understand concepts of projects and production management.** | **L3** | |
|  |

**Strength of CO Mapping to PO/PSOs with Justification:**

**1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cos** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** |
| **CO1** | **3** | **3** | **3** | **1** |  | **1** | **1** |  |  |  |  |  |  | **3** |
| **CO2** | **3** | **3** | **3** | **1** |  |  |  |  | **3** |  |  |  | **3** | **3** |
| **CO3** | **3** | **3** | **3** | **1** | **1** |  |  | **2** | **3** | **3** | **2** | **2** | **3** | **3** |
| **CO4** |  |  |  |  |  |  |  | **2** |  | **3** |  |  |  |  |
| **CO5** |  |  |  |  |  |  |  |  | **3** | **3** | **2** | **2** | **3** |  |
| **Avg** | **3** | **3** | **3** | **1** | **1** | **1** | **1** | **2** | **3** | **3** | **2** | **2** | **3** | **3** |

|  |  |
| --- | --- |
| CO-PO-PSO | Justification |
| CO1 🡪 PO1 (3)  CO1 🡪 PO2 (3)  CO1 🡪 PO3 (3)  CO1 🡪 PO4 (1)  CO1 🡪PO6(1)  CO1 🡪PO7(1)  CO1 🡪 PSO3(2) | * Strongly mapped as students apply the electronics and communication engineering skills to understand the principles of industrial projects. * Strongly mapped as students identify, formulate, review research literature and analyse complex engineering problems. * Strongly mapped as students design solutions for complex engineering problems with appropriate consideration for the public health and safety. * Slightly mapped as students can use research-based knowledge and methods including design of experiments, analysis and interpretation to provide valid conclusions. * Slightly mapped as students can able to understand the impact of professional engineering project solutions in societal and environmental contexts. * Slightly mapped as students can be able to apply reasoning of contextual project knowledge to assess societal, health and safety issues. * Strongly mapped as apply knowledge in various domain of IoT, real time systems, communication systems, VLSI and embedded systems, image and signal processing using hardware and software tools. |
| CO2 🡪 PO1 (3)  CO2 🡪 PO2 (3)  CO2 🡪 PO3 (3)  CO2 🡪 PO4 (1)  CO2 🡪 PO9 (3)  CO2 🡪 PSO1(3)  CO2 🡪 PSO2(3) | * Strongly mapped as students apply the electronics and communication engineering skills to understand the principles of industrial projects. * Strongly mapped as students identify, formulate, review research literature and analyse complex engineering problems. * Strongly mapped as students design solutions for complex engineering problems with appropriate consideration for the public health and safety. * Slightly mapped as students can use research-based knowledge and methods including design of experiments, analysis and interpretation to provide valid conclusions. * Strongly mapped as students can effectively act as an individual, and as a member or leader and work in a team. * Strongly mapped as students can design, verify and develop analog and digital systems by using state of art technology to contribute to the societal needs and solve industrial problem. * Strongly mapped as students apply knowledge in various domain of IoT, real time systems, communication systems, VLSI and embedded systems, image and signal processing using hardware and software tools and solve industrial problem. |
| CO3 🡪 PO1 (3)  CO3 🡪 PO2 (3)  CO3 🡪 PO3 (3)  CO3 🡪 PO4 (1)  CO3 🡪PO5(1)  CO3 🡪 PO8(2)  CO3 🡪 PO9 (3)  CO3🡪 PO10 (3)  CO3 🡪 PO11 (2)  CO3 🡪 PO12 (2)  CO3 🡪 PSO1(3)  CO3 🡪 PSO2(3) | * Strongly mapped as students apply the electronics and communication engineering skills to understand the principles of industrial projects. * Strongly mapped as students identify, formulate, review research literature and analyse complex engineering problems. * Strongly mapped as students design solutions for complex engineering problems with appropriate consideration for the public health and safety. * Slightly mapped as students can use research-based knowledge and methods including design of experiments, analysis and interpretation to provide valid conclusions. * Slightly mapped as students can Create, select, and apply appropriate techniques, resources, and modernengineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. * Moderately mapped as students can work ethically and professionally in the industry. * Strongly mapped as Students can effectively act as an individual, and as a member or leader and work in a team. * Strongly mapped as Students can comprehend and write effective reports and make documentation with effective presentation. * Moderately mapped as students can learn to management and financial skills required for the execution of project. * Moderately mapped as students can be engaged in life long learning in the broadest context of technological change through project implementation and completion. * Strongly mapped as students can design, verify and develop analog and digital systems by using state of art technology to contribute to the societal needs and solve industrial problem. * Strongly mapped as students apply knowledge in various domain of IoT, real time systems, communication systems, VLSI and embedded systems, image and signal processing using hardware and software tools and solve industrial problem. |
| CO4 🡪 PO8 (3)  CO4🡪 PO10 (3) | * Moderately mapped as students can work ethically and professionally in the industry. * Strongly mapped as Students can comprehend and write effective reports and make documentation with effective presentation. |
| CO5🡪 PO9 (3)  CO5🡪 PO10 (3)  CO5 🡪 PO11 (2)  CO5 🡪 PO12 (2)  CO5 🡪 PSO1(3) | * Strongly mapped as Students can effectively act as an individual, and as a member or leader and work in a team. * Strongly mapped as Students can comprehend and write effective reports and make documentation with effective presentation. * Moderately mapped as students can learn to management and financial skills required for the execution of project. * Moderately mapped as students can be engaged in life long learning in the broadest context of technological change through project implementation and completion. * Strongly mapped as students can design, verify and develop analog and digital systems by using state of art technology to contribute to the societal needs and solve industrial problem. |

**ACKNOWLEDGEMENTS**

Although a single sentence hardly suffices, we would like to thank almighty God for blessing us with his grace and taking our endeavour to a successful culmination.

We express our gratitude to our guide **Prof. S. B. Hugar, Assistant Professor, Dept. of E&CE, JCE, Belagavi,** for his valuable guidance and continual encouragement and assistance throughout the project work**.** We greatly appreciate the freedom and collegial respect. We are grateful to him for discussions about the technical matters and suggestions concerned to our topic.

We extend our sense of gratitude to **Dr. Salma S S, Project Coordinator, Dept. of E&CE, JCE, Belagavi,** for extending support and cooperation which helped us in completion of the project work.

We extend our sense of gratitude to **Prof. V. R. Bagali, Professor & Head, Dept. of E&CE, JCE, Belagavi,** for extending support and cooperation which helped us in completion of the project work.

We would like to express our sincere thanks to **Dr. J Shivakumar., Principal, JCE Belagavi,** for extending support and cooperation which helped us in the completion of the project work.

We would like to extend our gratitude to all staff of **Department of Electronics and Communication Engineering** for the help and support rendered to us.

We would like to extend our gratitude to all our family members and friends especially for their advice and moral support.

**ABSTRACT**

This project promotes an approach for the human computer interaction (HCL). Where we use real time camera for controlling the mouse function.

Our proposed project is on hand gesture-based system that allows users to control desktop mouse movements using hand gesture. To detect hand gesture movements, our system makes use of a desktop webcam. The goal is to control mouse cursor functions with a simple camera or webcam rather than a traditional or standard devices. Using only a camera, the Virtual Mouse provides an infrastructure between the user and the machine. It enables the user to interact with a machine without the need for any mechanical or physical devices, and even allows to control mouse functions.

The domain of the project is AI/ML. The programming language we used in this project is python. This AI virtual mouse project is based on the concept of computer vision.

**TABLE OF CONTENTS**

Chapter 1 introduction 1

[1.1 MOTIVATION 2](#_Toc165316011)

[1.2 PROBLEM STATEMENT 2](#_Toc165316012)

[1.3 OBJECTIVE OF PROJECT 2](#_Toc165316013)

[1.4 LITERATURE SURVEY 3](#_Toc165316014)

[1.5 DESIGN APPROACH 4](#_Toc165316015)

[Chapter 2 hardware design 5](#_Toc165316016)

[2.2 POWER SUPPLY REQUIREMENTS 6](#_Toc165316017)

[Chapter 3 software design 7](#_Toc165316018)

[3.1 PROGRAMMING LANGUAGE 7](#_Toc165316019)

[3.2 DEVELOPMENT TOOLS 7](#_Toc165316020)

[3.3 DESIGN 8](#_Toc165316021)

[3.3.1 MAIN FLOW DIAGRAM 8](#_Toc165316022)

[3.3.2 SYSTEM DEVELOPMENT 10](#_Toc165316023)

[3.3.3 GESTURES 11](#_Toc165316024)

[3.3.4 Algorithm &techniques used 14](#_Toc165316025)

[3.3.5 PERFORMANCE ANALYSIS 16](#_Toc165316026)

[3.3.6 TEST CASES 17](#_Toc165316027)

[Chapter 4 applications, advantages and disadvantages 19](#_Toc165316028)

[conclusion and future scope 20](#_Toc165316029)

[references 21](#_Toc165316030)

**list OF FIGURES**

Figure 1: Laptop 5

Figure 2: Power Adaptor 6

Figure 3: Gesture Capture 10

Figure 4: Gesture A 11

Figure 5: Gesture B 11

Figure 6: Gesture C 12

Figure 7: Gesture D 12

Figure 8: Gesture E 12

Figure 9: Gesture F 13

Figure 10: Gesture G 13

**list OF tables**

Table 1: Performance Analysis 16

Table 2: Test Data 17

Table 3: Abbreviations 22

**list OF flowcharts**

Flowchart 1: Main Module 8

Flowchart 2: Data Flow Diagram 14

# introduction

Creating an AI-based virtual mouse using Python typically involves leveraging computer vision techniques and machine learning algorithms.

Setting Up Environment: First, ensure you have Python installed along with necessary libraries such as OpenCV, NumPy, and perhaps TensorFlow or for machine learning tasks.

Collecting Data: You'll need a dataset of hand movements to train your model. This dataset could consist of images or video recordings of hand movements representing different mouse actions like moving, clicking, and scrolling.

Preprocessing Data: Preprocess the data to extract relevant features. For example, you might apply techniques like background subtraction, thresholding, or skin detection to isolate the hand from the background.

Training the Model: Use machine learning algorithms to train a model that can recognize different hand movements. Convolutional Neural Networks (CNNs) are commonly used for this task due to their effectiveness in image recognition tasks.

Implementing the Virtual Mouse: Once you have a trained model, you can use it to recognize hand movements in real-time video streams from a webcam. Based on the recognized movements, you can then control the position of the virtual mouse cursor on the screen.

Mouse Control Mechanism: Implement the logic to control the mouse cursor based on the recognized hand movements. This might involve mapping hand positions to cursor coordinates and simulating mouse clicks and scrolls.

Testing and Refinement: Test your virtual mouse system extensively to ensure it behaves as expected in different scenarios. You may need to fine-tune your model or adjust parameters to improve its performance.

Deployment: Once you're satisfied with the performance, you can deploy your virtual mouse system for practical use.

## MOTIVATION

With the development technologies in area of devices that we use in our daily life, these devices are becoming compact in the form of Bluetooth or wireless technologies.

The proposal of an AI virtual mouse system that makes use of the hand gestures and hand tip detection for performing mouse functions in the computer using computer vision.

## PROBLEM STATEMENT

The proposed AI virtual mouse system can be used to overcome problems in the real world such as situations where there is no space to use a physical mouse and also for the persons who have problems in their hands and are not able to control a physical mouse. Also, as the system is virtual it can be a prototype for the upcoming field virtual reality, so the proposed AI virtual mouse can be used to overcome these problems of previously used model using hand glove, since Hand Gesture and Hand Tip detection is used to control the PC mouse functions by using a webcam or a built-in camera.

## OBJECTIVE OF PROJECT

The main objective of the proposed AI virtual mouse system is to develop an alternative to the regular and traditional mouse system to perform and control the mouse functions, and this can be achieved with the help of a web camera that captures the hand gestures and hand tip and then processes these frames to perform the particular mouse.

## LITERATURE SURVEY

As modern technology of human computer interactions become important in our everyday lives, varieties of mouse with all kinds of shapes and sizes were invented, from a casual office mouse to a hard-core gaming mouse. However, there are some limitations to these hardware as they are not as environmentally friendly as it seems. For example, the physical mouse requires a flat surface to operate, not to mention that it requires a certain area to fully utilize the functions offered. Furthermore, some of these hardware are completely useless when it comes to interact with the computers remotely due to the cable lengths limitations, rendering it inaccessible.

Multi-point Interactive Whiteboards are available using the Wiimote[4]. The components used are IR pen, computer with Windows XP (installed with Microsoft .NET framework, the Wiimote Connect program and the Wiimote Whiteboard software), wiimote controller, a beamer capable of a 1024 x 786 pixel resolution. Here the wiimote controller tracks the infra-red source on the white board and sends info to PC via Bluetooth. The teaching platform comprises of a Wii-mote-based multi-touch teaching station, a Wii-mote-based interactive whiteboard and a Wii-mote-based stylus input conversion tool[5]. According to the literature survey, most people have used the Wii-mote to configure it as a virtual marker.

The current system is comprised of a generic mouse and trackpad monitor control system, as well as the absence of a hand gesture control system. The use of a hand gesture to access the monitor screen from a distance is not possible. Even though it is primarily attempting to implement, the scope is simply limited in the virtual mouse field. The existing virtual mouse control system consists of simple mouse operations using a hand recognition system, in which we can control the mouse pointer, left click, right click, and drag, and so on. The use of hand recognition in the future will not be used. Even though there are a variety of systems for hand recognition, the system they used is static hand recognition, which is simply a recognition of the shape made by the hand and the definition of action for each shape made, which is limited to a few defined actions and causes a lot of confusion. As technology advances, there are more and more alternatives to using a mouse.

## DESIGN APPROACH

**Firmware Design & Development**

* + Requirement Gathering
  + Software Design Document
  + Firmware Design and Development
  + Coding

**Testing**

* + Module Testing
  + Unit & Integration testing of the software
  + System Testing

# hardware design

**2.1 LAPTOP**



Figure 1: Laptop

To get an image, a webcam is required. Mouse sensitivity is constant with digicam clarity. The excellent consumer information is demonstrated whilst the digicam configuration is high enough. The camera is used for real-time images at any time the pc is on. The system will pick out the suitable movement primarily based on the contact and finger movement.

**Camera Used in the AI Virtual Mouse System:** The proposed AI virtual mouse system is based on the frames that have been captured by the webcam in a laptop or PC. By using the Python computer vision library OpenCV, the video capture object is created, and the web camera will start capturing video, as shown in Figure. The web camera captures and passes the frames to the AI virtual system.

**Capturing the Video and Processing:**The AI virtual mouse system uses the webcam where each frame is captured till the termination of the program.

## POWER SUPPLY REQUIREMENTS



Figure 2: 12VDC – 1.5A Power Adaptor

**Specifications:**

* Input Voltage : 100~240V
* Output Power : ~30W
* Output Voltage : DC 19V ±5%
* Output Current : DC 1.5A

# software design

## PROGRAMMING LANGUAGE

* Python

Python is a popular programming language known for its simplicity and versatility. It's widely used in various domains such as web development, data science, machine learning, and more. Its readability and extensive library support make it a go-to choice for many developers.

## DEVELOPMENT TOOLS

* PyCharm

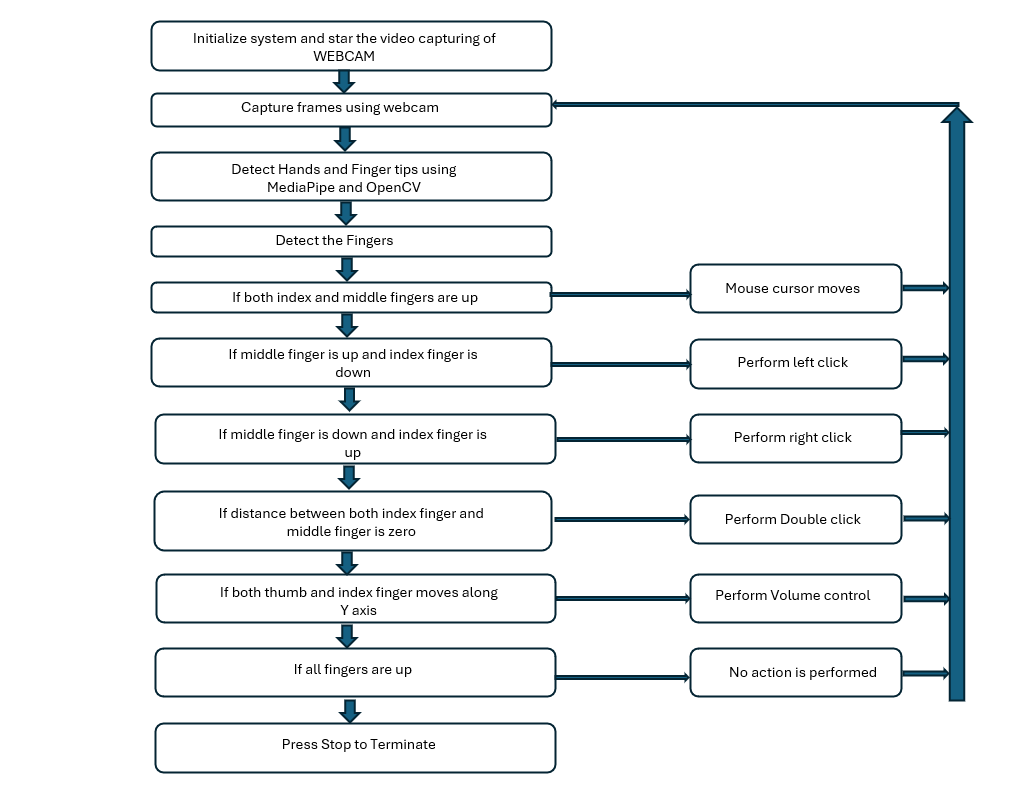
PyCharm is a powerful Integrated Development Environment (IDE) specifically designed for Python development. It offers features like code completion, syntax highlighting, debugging tools, version control integration, and more, to enhance the productivity of Python developers.

* OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. It provides tools and algorithms to help developers build applications that process and analyze images and videos. OpenCV is widely used in fields like robotics, augmented reality, facial recognition, and object detection.

## DESIGN

### MAIN FLOW DIAGRAM



Flowchart : Main Module

The various functions and conditions used in the system are explained in the flowchart 1 of the real-time AI virtual mouse system.

Initialize system and start video capturing of Webcam: The system first initializes and starts capturing video frames from the webcam.

Capture frames using webcam: The captured video frames are fed into the next block.

Detect Hands and Finger tips using MediaPipe and OpenCV: OpenCV is a library commonly used for real-time computer vision [1]. MediaPipe is a framework built on top of OpenCV that provides pre-built machine learning models for tasks like hand pose estimation [2]. This block uses these tools to detect the hand and finger tips in the captured frame.

Detect the Fingers: After identifying the hand and finger tips, this block specifically detects which fingers are up.

Conditional blocks based on finger detection: The system then enters a series of conditional blocks that determine what action, if any, to perform based on the detected finger positions:

* If both index and middle fingers are up: The mouse cursor moves.
* If middle finger is up and index finger is down: A left click is performed.
* If middle finger is down and index finger is up: A right click is performed.
* If distance between both index finger and middle finger is zero: A double click is performed.
* If both thumb and index finger moves along Y axis: The system performs volume control.
* If all fingers are up: No action is performed.
* Press Stop to Terminate: The user can stop the system by pressing a stop button.

### SYSTEM DEVELOPMENT

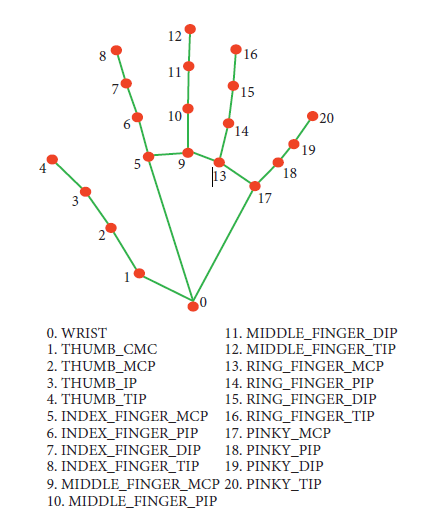


Figure 3: Gesture Capture

**Rectangular Region for Moving through the Window:**The AI virtual mouse system makes use of the transformational algorithm, and it converts the coordinates of fingertip from the webcam screen to the computer window full screen for controlling the mouse. When the hands are detected and when we find which finger is up for performing the specific mouse function, a rectangular box is drawn with respect to the computer window in the webcam region where we move throughout the window using the mouse cursor.

**Detecting Which Finger Is Up and Performing the Particular Mouse Function:.** In this stage, we are detecting which finger is up using the tip Id of the respective finger that we found using the MediaPipe and the respective co-ordinates of the fingers that are up , and according to that, the particular mouse function is performed.

**Mouse Functions Depending on the Hand Gestures and Hand Tip Detection Using Computer Vision For the Mouse Cursor Moving around the Computer Window:**If the index finger is up with tip Id = 1 or both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up, the mouse cursor is made to move around the  window of the computer using the AutoPy package of Python, as shown in Figure.

**For the Mouse to Perform Left Button Click:**If both the index finger with tip Id = 1 and the thumb finger with tip Id = 0 are up and the distance between the two fingers is lesser than 30px, the computer is made to perform the left mouse button click using the pynput.

### GESTURES

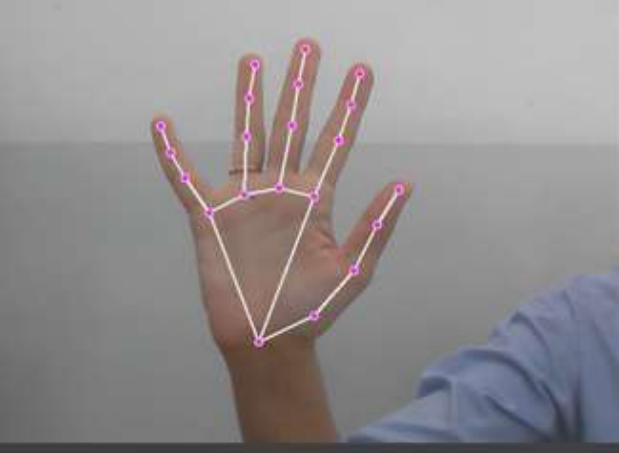


Figure 4: Gesture A: Camera Used in the Virtual Gesture Mouse project

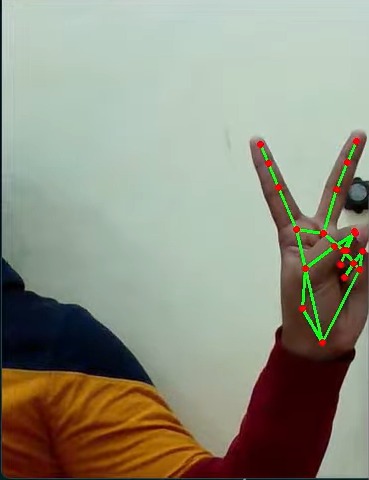
****

Figure 5: Gesture B: Moving Hand through the Window using Rectangular Area



Figure 6: Gesture C: For the Mouse to Perform Left Button Click



Figure 7: Gesture D: For the Mouse to Perform Double Click



Figure 8: Gesture E: For the Mouse to Perform Right Button Click



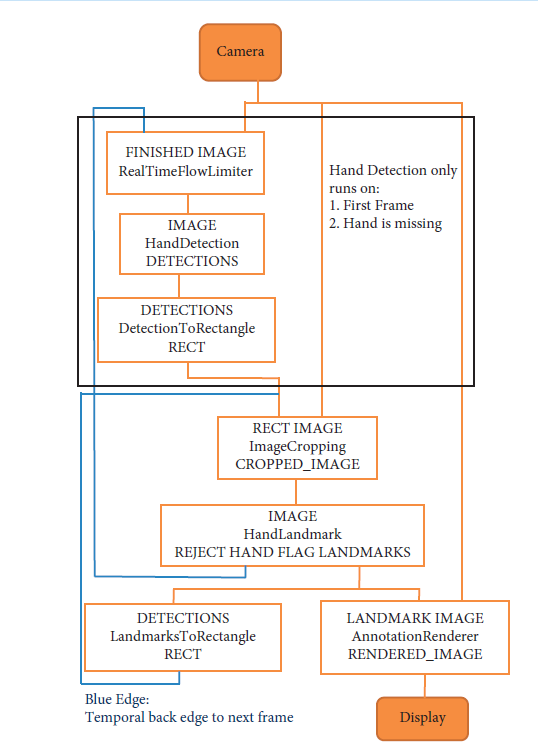
Figure 9: Gesture F: For the Mouse to Perform Scroll Function and Control Volume

A person holding a drawing

Description automatically generated

Figure 10: Gesture G: For the Mouse to Perform Drag & Drop Function

### Algorithm & techniques used



Flowchart 2: Data Flow Diagram

For the purpose of detection of hand gestures and hand tracking, the MediaPipe framework is used, and OpenCV library is used for computer vision. The algorithm makes use of the machine learning concepts to track and recognize the hand gestures and hand tip*.*

**MediaPipe** is a framework which is used for applying in a machine learning pipeline, and it is an opensource framework of Google. The MediaPipe framework is useful for cross platform development since the framework is built using the time series data. The MediaPipe framework is multimodal, where this framework can be applied to various audios and videos. The MediaPipe framework is used by the developer for building and analyzing the systems through graphs, and it also been used for developing the systems for the application purpose. The steps involved in the system that uses MediaPipe are carried out in the pipeline configuration. The pipeline created can run in various platforms allowing scalability in mobile and desktops. The MediaPipe framework is based on three fundamental parts; they are performance evaluation, framework for retrieving sensor data, and a collection of components which are called calculators, and they are reusable. A pipeline is a graph which consists of components called calculators, where each calculator is connected by streams in which the packets of data flow through. Developers can replace or define custom calculators anywhere in the graph creating their own application. The calculators and streams combined create a data-flow diagram; the graph is created with MediaPipe where each node is a calculator, and the nodes are connected by streams.

Single-shot detector model is used for detecting and recognizing a hand or palm in real time. The single-shot detector model is used by the MediaPipe. First, in the hand detection module, it is first trained for a palm detection model because it is easier to train palms. Furthermore, the non-maximum suppression works significantly better on small objects such as palms or fists. A model of hand landmark consists of locating joint or knuckle co-ordinates in the hand region,

**OpenCV** is a computer vision library which contains image-processing algorithms for object detection. OpenCV is a library of python programming language, and real-time computer vision applications can be developed by using the computer vision library. The OpenCV library is used in image and video processing and analysis such as face detection and object detection.

### PERFORMANCE ANALYSIS

**Table** **1: Performance Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mouse function performed** | **Success** | **Failure** | **Accuracy (%)** |
| Mouse movement | 100 | 0 | 100% |
| Left button click | 98 | 2 | 98% |
| Right button click | 99 | 1 | 99% |
| Scroll function | 93 | 7 | 93% |
| Volume control | 96 | 4 | 96% |
| No action performed | 100 | 0 | 100% |
| Result | 681 | 19 | 97.28% |

In the proposed AI virtual mouse system, the concept of advancing the human-computer interaction using computer vision is given.

Cross comparison of the testing of the AI virtual mouse system is difficult because only limited numbers of datasets are available. The hand gestures and finger tip detection have been tested in various illumination conditions and also been tested with different distances from the webcam for tracking of the hand gesture and hand tip detection. An experimental test has been conducted to summarize the results shown in Table 1.

The test was performed 25 times by 4 persons resulting in 600 gestures with manual labelling, and this test has been made in different light conditions and at different distances from the screen, and each person tested the AI virtual mouse system 10 times in normal light conditions, 5 times in faint light conditions, 5 times in close distance from the webcam, and 5 times in long distance from the webcam, and the experimental results are tabulated in Table1.

From Table 1 , it can be seen that the proposed AI virtual mouse system had achieved an accuracy of about 97%. From this 97% accuracy of the proposed AI virtual mouse system, we come to know that the system has performed well. As seen in Table, the accuracy is low for “Scroll function” as this is the hardest gesture for the computer to understand. The accuracy for scroll function is low because the gesture used for performing the particular mouse function is harder. Also, the accuracy is very good and high for all the other gestures. Compared to previous approaches for virtual mouse, our model worked very well with 97% accuracy.

### TEST CASES

Table : Test data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case id** | **Scenario** | **Boundary**  **Value** | **Expected Result** | **Actual Result** | **Status** |
| **1** | Used in normal environment. | >90% | In normal environment hand gestures can be recognized easily. | Hand gestures got easily recognized and work properly. | **Passed** |
| **2** | Used in bright environment. | >60% | In brighter environment, software should work fine as it easily detects the hand movements but in a more brighter conditions it may not detect the hand gestures as expected. | In bright conditions the software works very well. | **Passed** |
| **3** | Used in dark environment | <30% | In dark environment, It should work properly. | In dark environment software didn’t work properly in detecting hand gestures. | **Failed** |
| **4** | Used at a near distance  (15cm) from the web cam. | >80% | At this distance, this software should perform perfectly. | It works fine and all features works properly. | **Passed** |
| **5** | Used at a far distance  (35cm) from the web cam. | >95% | At this distance, this software should work fine. | At this distance, it is working properly. | **Passed** |
| **6** | Used at a farther distance  (60cm) from the web cam. | >60% | At this distance, their will be some problem in detecting hand gestures but it should work fine. | At this distance, The  functions of this software works properly. | **Passed** |

# applications, advantages and disadvantages

**APPLICATIONS**

The AI virtual mouse system is useful for many applications; it can be used to reduce the space for using the physical mouse, and it can be used in situations where we cannot use the physical mouse. The system eliminates the usage of devices, and it improves the human-computer interaction.

* The proposed model has a greater accuracy of 99% which is far greater than the that of other proposed models for virtual mouse, and it has many applications.
* Persons with problems in their hands can use this system to control the mouse functions in the computer.
* As the proposed model does not need any type of hand glove it can be used as a prototype for virtual reality system.

**ADVANTAGES**

* It eliminates the need for a physical mouse device, which can save space, reduce cost, and avoid battery or connection issues.
* It can enhance the user experience by providing a natural and intuitive way of interacting with the computer.
* AI-based virtual mice can potentially be implemented using affordable hardware components, such as webcams or depth sensors, combined with open-source software libraries and frameworks available in Python.

**DISADVANTAGES**

* It may require a high-quality webcam and a good lighting condition to capture the hand gestures accurately.
* It may not be compatible with some applications or platforms that do not support mouse input.

# conclusion and future scope

**CONCLUSION**

The main objective of the AI virtual mouse system is to control the mouse cursor functions by using the hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a built-in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions. From the results of the model, we can come to a conclusion that the proposed AI virtual mouse system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems. Since the proposed model has greater accuracy, the AI virtual mouse can be used for real-world applications, and also, it can be used as a prototype for virtual reality system, since the proposed mouse system can be used virtually using hand gestures without using the traditional physical mouse. The model has some limitations such as small decrease in accuracy in right click mouse function and some difficulties in clicking and dragging to select the text. Hence, we will work next to overcome these limitations by improving the finger tip detection algorithm to produce more accurate results.

**FUTURE SCOPE**

Due to accuracy and efficiency plays an important role in making the program as useful as an actual physical mouse, a few techniques had to be implemented. After implanting such type of application there is big replacement of physical mouse i.e., there is no need of any physical mouse. Each & every movement of physical mouse is done with this motion tracking mouse (virtual mouse).There are several features and improvements needed in order for the program to be more user friendly, accurate, and flexible in various environments. The following describes the improvements and the features required:

* Smart Movement: Due to the current recognition process are limited within 25cm radius, an adaptive zoom in/out functions are required to improve the covered distance, where it can automatically adjust the focus rate based on the distance between the users and the webcam.
* Better Accuracy & Performance: The response time are heavily relying on the hardware of the machine, this includes the processing speed of the processor.

# references

[1] OpenCV Website – [www.opencv.org](http://www.opencv.org) [1999]

[2] MSDN Microsoft developers network – [www.msdn.microsoft.com](http://www.msdn.microsoft.com) [1992]

[3] Code project – www.codeproject.com/Articles/498193/Mouse-Control-via-

Webcam [2012]

[4] Aniket Tatipamula’s Blog - http://anikettatipamula.blogspot.in/2012/02/hand-

gesture-using-opencv.html

[5] Microsoft Research Paper- http://research.microsoft.com/en-

us/um/people/awf/bmvc02/project.pdf

**APPENDIX A**

**ABBREVIATIONS USED**

Table 3: Abbreviations

|  |  |
| --- | --- |
| GUI | Graphical User Interface |
| HCI | Human Computer Interaction |
| MHI | Motion History Images |
| IDE | Integrated Development Environment |
| OpenCV | Open Source Computer Vision |
| NUI | Natural User Interface |

**APPENDIX** B

**CODE**

# Imports

import cv2

import mediapipe as mp

import pyautogui

import math

from enum import IntEnum

from ctypes import cast, POINTER

from comtypes import CLSCTX\_ALL

from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume

from google.protobuf.json\_format import MessageToDict

import screen\_brightness\_control as sbcontrol

pyautogui.FAILSAFE = False

mp\_drawing = mp.solutions.drawing\_utils

mp\_hands = mp.solutions.hands

# Gesture Encodings

class Gest(IntEnum):

# Binary Encoded

FIST = 0

PINKY = 1

RING = 2

MID = 4

LAST3 = 7

INDEX = 8

FIRST2 = 12

LAST4 = 15

THUMB = 16

PALM = 31

# Extra Mappings

V\_GEST = 33

TWO\_FINGER\_CLOSED = 34

PINCH\_MAJOR = 35

PINCH\_MINOR = 36

# Multi-handedness Labels

class HLabel(IntEnum):

MINOR = 0

MAJOR = 1

# Convert Mediapipe Landmarks to recognizable Gestures

class HandRecog:

def \_\_init\_\_(self, hand\_label):

self.finger = 0

self.ori\_gesture = Gest.PALM

self.prev\_gesture = Gest.PALM

self.frame\_count = 0

self.hand\_result = None

self.hand\_label = hand\_label

def update\_hand\_result(self, hand\_result):

self.hand\_result = hand\_result

def get\_signed\_dist(self, point):

sign = -1

if self.hand\_result.landmark[point[0]].y < self.hand\_result.landmark[point[1]].y:

sign = 1

dist = (self.hand\_result.landmark[point[0]].x - self.hand\_result.landmark[point[1]].x)\*\*2

dist += (self.hand\_result.landmark[point[0]].y - self.hand\_result.landmark[point[1]].y)\*\*2

dist = math.sqrt(dist)

return dist\*sign

def get\_dist(self, point):

dist = (self.hand\_result.landmark[point[0]].x - self.hand\_result.landmark[point[1]].x)\*\*2

dist += (self.hand\_result.landmark[point[0]].y - self.hand\_result.landmark[point[1]].y)\*\*2

dist = math.sqrt(dist)

return dist

def get\_dz(self,point):

return abs(self.hand\_result.landmark[point[0]].z - self.hand\_result.landmark[point[1]].z)

# Function to find Gesture Encoding using current finger\_state.

# Finger\_state: 1 if finger is open, else 0

def set\_finger\_state(self):

if self.hand\_result == None:

return

points = [[8,5,0],[12,9,0],[16,13,0],[20,17,0]]

self.finger = 0

self.finger = self.finger | 0 #thumb

for idx,point in enumerate(points):

dist = self.get\_signed\_dist(point[:2])

dist2 = self.get\_signed\_dist(point[1:])

try:

ratio = round(dist/dist2,1)

except:

ratio = round(dist1/0.01,1)

self.finger = self.finger << 1

if ratio > 0.5 :

self.finger = self.finger | 1

# Handling Fluctations due to noise

def get\_gesture(self):

if self.hand\_result == None:

return Gest.PALM

current\_gesture = Gest.PALM

if self.finger in [Gest.LAST3,Gest.LAST4] and self.get\_dist([8,4]) < 0.05:

if self.hand\_label == HLabel.MINOR :

current\_gesture = Gest.PINCH\_MINOR

else:

current\_gesture = Gest.PINCH\_MAJOR

elif Gest.FIRST2 == self.finger :

point = [[8,12],[5,9]]

dist1 = self.get\_dist(point[0])

dist2 = self.get\_dist(point[1])

ratio = dist1/dist2

if ratio > 1.7:

current\_gesture = Gest.V\_GEST

else:

if self.get\_dz([8,12]) < 0.1:

current\_gesture = Gest.TWO\_FINGER\_CLOSED

else:

current\_gesture = Gest.MID

else:

current\_gesture = self.finger

if current\_gesture == self.prev\_gesture:

self.frame\_count += 1

else:

self.frame\_count = 0

self.prev\_gesture = current\_gesture

if self.frame\_count > 4 :

self.ori\_gesture = current\_gesture

return self.ori\_gesture

# Executes commands according to detected gestures

class Controller:

tx\_old = 0

ty\_old = 0

trial = True

flag = False

grabflag = False

pinchmajorflag = False

pinchminorflag = False

pinchstartxcoord = None

pinchstartycoord = None

pinchdirectionflag = None

prevpinchlv = 0

pinchlv = 0

framecount = 0

prev\_hand = None

pinch\_threshold = 0.3

def getpinchylv(hand\_result):

dist = round((Controller.pinchstartycoord - hand\_result.landmark[8].y)\*10,1)

return dist

def getpinchxlv(hand\_result):

dist = round((hand\_result.landmark[8].x - Controller.pinchstartxcoord)\*10,1)

return dist

def changesystembrightness():

currentBrightnessLv = sbcontrol.get\_brightness()/100.0

currentBrightnessLv += Controller.pinchlv/50.0

if currentBrightnessLv > 1.0:

currentBrightnessLv = 1.0

elif currentBrightnessLv < 0.0:

currentBrightnessLv = 0.0

sbcontrol.fade\_brightness(int(100\*currentBrightnessLv) , start = sbcontrol.get\_brightness())

def changesystemvolume():

devices = AudioUtilities.GetSpeakers()

interface = devices.Activate(IAudioEndpointVolume.\_iid\_, CLSCTX\_ALL, None)

volume = cast(interface, POINTER(IAudioEndpointVolume))

currentVolumeLv = volume.GetMasterVolumeLevelScalar()

currentVolumeLv += Controller.pinchlv/50.0

if currentVolumeLv > 1.0:

currentVolumeLv = 1.0

elif currentVolumeLv < 0.0:

currentVolumeLv = 0.0

volume.SetMasterVolumeLevelScalar(currentVolumeLv, None)

def scrollVertical():

pyautogui.scroll(120 if Controller.pinchlv>0.0 else -120)

def scrollHorizontal():

pyautogui.keyDown('shift')

pyautogui.keyDown('ctrl')

pyautogui.scroll(-120 if Controller.pinchlv>0.0 else 120)

pyautogui.keyUp('ctrl')

pyautogui.keyUp('shift')

# Locate Hand to get Cursor Position

# Stabilize cursor by Dampening

def get\_position(hand\_result):

point = 9

position = [hand\_result.landmark[point].x ,hand\_result.landmark[point].y]

sx,sy = pyautogui.size()

x\_old,y\_old = pyautogui.position()

x = int(position[0]\*sx)

y = int(position[1]\*sy)

if Controller.prev\_hand is None:

Controller.prev\_hand = x,y

delta\_x = x - Controller.prev\_hand[0]

delta\_y = y - Controller.prev\_hand[1]

distsq = delta\_x\*\*2 + delta\_y\*\*2

ratio = 1

Controller.prev\_hand = [x,y]

if distsq <= 25:

ratio = 0

elif distsq <= 900:

ratio = 0.07 \* (distsq \*\* (1/2))

else:

ratio = 2.1

x , y = x\_old + delta\_x\*ratio , y\_old + delta\_y\*ratio

return (x,y)

def pinch\_control\_init(hand\_result):

Controller.pinchstartxcoord = hand\_result.landmark[8].x

Controller.pinchstartycoord = hand\_result.landmark[8].y

Controller.pinchlv = 0

Controller.prevpinchlv = 0

Controller.framecount = 0

# Hold final position for 5 frames to change status

def pinch\_control(hand\_result, controlHorizontal, controlVertical):

if Controller.framecount == 5:

Controller.framecount = 0

Controller.pinchlv = Controller.prevpinchlv

if Controller.pinchdirectionflag == True:

controlHorizontal() #x

elif Controller.pinchdirectionflag == False:

controlVertical() #y

lvx = Controller.getpinchxlv(hand\_result)

lvy = Controller.getpinchylv(hand\_result)

if abs(lvy) > abs(lvx) and abs(lvy) > Controller.pinch\_threshold:

Controller.pinchdirectionflag = False

if abs(Controller.prevpinchlv - lvy) < Controller.pinch\_threshold:

Controller.framecount += 1

else:

Controller.prevpinchlv = lvy

Controller.framecount = 0

elif abs(lvx) > Controller.pinch\_threshold:

Controller.pinchdirectionflag = True

if abs(Controller.prevpinchlv - lvx) < Controller.pinch\_threshold:

Controller.framecount += 1

else:

Controller.prevpinchlv = lvx

Controller.framecount = 0

def handle\_controls(gesture, hand\_result):

x,y = None,None

if gesture != Gest.PALM :

x,y = Controller.get\_position(hand\_result)

# flag reset

if gesture != Gest.FIST and Controller.grabflag:

Controller.grabflag = False

pyautogui.mouseUp(button = "left")

if gesture != Gest.PINCH\_MAJOR and Controller.pinchmajorflag:

Controller.pinchmajorflag = False

if gesture != Gest.PINCH\_MINOR and Controller.pinchminorflag:

Controller.pinchminorflag = False

# implementation

if gesture == Gest.V\_GEST:

Controller.flag = True

pyautogui.moveTo(x, y, duration = 0.1)

elif gesture == Gest.FIST:

if not Controller.grabflag :

Controller.grabflag = True

pyautogui.mouseDown(button = "left")

pyautogui.moveTo(x, y, duration = 0.1)

elif gesture == Gest.MID and Controller.flag:

pyautogui.click()

Controller.flag = False

elif gesture == Gest.INDEX and Controller.flag:

pyautogui.click(button='right')

Controller.flag = False

elif gesture == Gest.TWO\_FINGER\_CLOSED and Controller.flag:

pyautogui.doubleClick()

Controller.flag = False

elif gesture == Gest.PINCH\_MINOR:

if Controller.pinchminorflag == False:

Controller.pinch\_control\_init(hand\_result)

Controller.pinchminorflag = True

Controller.pinch\_control(hand\_result,Controller.scrollHorizontal, Controller.scrollVertical)

elif gesture == Gest.PINCH\_MAJOR:

if Controller.pinchmajorflag == False:

Controller.pinch\_control\_init(hand\_result)

Controller.pinchmajorflag = True

Controller.pinch\_control(hand\_result,Controller.changesystembrightness, Controller.changesystemvolume)

'''

---------------------------------------- Main Class ----------------------------------------

Entry point of Gesture Controller

'''

class GestureController:

gc\_mode = 0

cap = None

CAM\_HEIGHT = None

CAM\_WIDTH = None

hr\_major = None # Right Hand by default

hr\_minor = None # Left hand by default

dom\_hand = True

def \_\_init\_\_(self):

GestureController.gc\_mode = 1

GestureController.cap = cv2.VideoCapture(0)

GestureController.CAM\_HEIGHT = GestureController.cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT)

GestureController.CAM\_WIDTH = GestureController.cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH)

def classify\_hands(results):

left , right = None,None

try:

handedness\_dict = MessageToDict(results.multi\_handedness[0])

if handedness\_dict['classification'][0]['label'] == 'Right':

right = results.multi\_hand\_landmarks[0]

else :

left = results.multi\_hand\_landmarks[0]

except:

pass

try:

handedness\_dict = MessageToDict(results.multi\_handedness[1])

if handedness\_dict['classification'][0]['label'] == 'Right':

right = results.multi\_hand\_landmarks[1]

else :

left = results.multi\_hand\_landmarks[1]

except:

pass

if GestureController.dom\_hand == True:

GestureController.hr\_major = right

GestureController.hr\_minor = left

else :

GestureController.hr\_major = left

GestureController.hr\_minor = right

def start(self):

handmajor = HandRecog(HLabel.MAJOR)

handminor = HandRecog(HLabel.MINOR)

with mp\_hands.Hands(max\_num\_hands = 2,min\_detection\_confidence=0.5, min\_tracking\_confidence=0.5) as hands:

while GestureController.cap.isOpened() and GestureController.gc\_mode:

success, image = GestureController.cap.read()

if not success:

print("Ignoring empty camera frame.")

continue

image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR\_BGR2RGB)

image.flags.writeable = False

results = hands.process(image)

image.flags.writeable = True

image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR)

if results.multi\_hand\_landmarks:

GestureController.classify\_hands(results)

handmajor.update\_hand\_result(GestureController.hr\_major)

handminor.update\_hand\_result(GestureController.hr\_minor)

handmajor.set\_finger\_state()

handminor.set\_finger\_state()

gest\_name = handminor.get\_gesture()

if gest\_name == Gest.PINCH\_MINOR:

Controller.handle\_controls(gest\_name, handminor.hand\_result)

else:

gest\_name = handmajor.get\_gesture()

Controller.handle\_controls(gest\_name, handmajor.hand\_result)

for hand\_landmarks in results.multi\_hand\_landmarks:

mp\_drawing.draw\_landmarks(image, hand\_landmarks, mp\_hands.HAND\_CONNECTIONS)

else:

Controller.prev\_hand = None

cv2.imshow('Gesture Controller', image)

if cv2.waitKey(5) & 0xFF == 13:

break

GestureController.cap.release()

cv2.destroyAllWindows()

# uncomment to run directly

gc1 = GestureController()

gc1.start()