**AI Virtual Mouse System**

**Mini Project report submitted to**

Text, logo

Description automatically generated with medium confidence

**Group Members:**

* Kartik Mittal (215890116)
* Ananya Gupta (215890030
* Anshuman Mukherjee (215890086)
* Sakcham Gupta (215890050)

Under the supervision of

**Dr. Megha. P. Arakeri**

**Professor**

Department of Computer Science and Engineering

Manipal Institute of Technology

Bengaluru

**Manipal Institute of Technology**

**Bengaluru Campus-560064, Karnataka, India.**

**Abstract**

The mouse is one of the best inventions of human-computer interaction (HCI) technology. Currently the wireless mouse or Bluetooth mouse is still using the device and is not completely removed from the device as it runs on battery power and uses an adapter to connect to the computer. In the AI ​​virtual mouse system concept, this limitation can be overcome by using computer vision to detect hand gestures and gestures using a network camera or on-board display. The algorithms used in the system use machine learning algorithms. The computer can be effectively controlled based on gestures, and left-click, right-click, scrolling functions and computer cursor functions can be performed without using a physical mouse. The algorithm is based on deep learning to recognize hands. Therefore, the proposed system will prevent the spread of COVID-19 by eliminating the influence of people and equipment from the management of computers.

|  |  |  |
| --- | --- | --- |
| **Contents** | | **Page. No.** |
| **1** | Introduction………………………………………………….. | **4** |
| **2** | Problem Statement…………………………………………... | **5** |
| **3** | Objective…………………………………………………….. | **6** |
| **4** | Packages Used………………………………………………... | **7** |
| **5** | Methodology……………………………………………….... | **10** |
| **6** | Implementation Details…………………………………….... | **16** |
| **7** | Results and Accuracy……………………………………….. | **21** |
| **8** | Project Management Timeline……………………………… | **23** |
| **9** | Conclusion…………………………………………………... | **24** |
| References…………………………………………………………... | | **25** |

**Chapter 1**

**Introduction**

With the development of technology in the field of reality and the devices we use in our daily lives, these devices increasingly appear in the form of Bluetooth or wireless technology. This report presents an AI virtual mouse that uses computer vision to perform computer mouse operations using pointing and finger tracking. The main purpose of the system is to perform computer mouse cursor functions and scrolling functions using a network camera or a built-in camera on the computer, rather than using a traditional mouse device.

The usefulness of a webcam can also be greatly extended to other HCI application such as a sign language database or motion controller. Over the past decades there have been significant advancements in HCI technologies for gaming purposes, such as the Microsoft Kinect and Nintendo Wii.

The AI ​​virtual mouse system was designed using the **Python** programming language, and computer vision **OpenCV** was used in the AI ​​virtual mouse system. In the proposed AI virtual mouse system, the model uses the **MediaPipe** package to track hands and arms, and also uses the **PyAutoGUI** package to move the window screen for left-clicking, tapping, and scrolling. The results of the model are quite accurate and the proposed model works well in real applications using CPU instead of GPU.

**Chapter 2**

**Problem statement**

The suggested AI-based virtual mouse system serves as a viable solution to address various real-world challenges. It offers a practical alternative in scenarios where physical space is limited and for individuals facing difficulties using a conventional mouse due to hand-related issues. Moreover, in light of the COVID-19 pandemic, the concern over surface transmission has highlighted the need for touchless interactions with devices. By employing hand gesture and tip detection via a webcam or integrated camera, this AI-powered virtual mouse effectively mitigates these concerns and provides a safer and more hygienic user experience.

**Chapter 3**

**Objectives**

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. Other objectives can be:

1. **Accessibility:** In cramped or on-the-go conditions, like during travel or in a confined workspace, having a more flexible and convenient way to interact with devices is essential.
2. **Convenience and Flexibility**: A need for convenience and flexibility arises in limited physical spaces, such as in small work areas or while traveling, making a more adaptable option for device
3. **Hygiene and safety**: To enable touchless interaction with devices and reduce the risk of surface transmission of pathogens in situations like COVID-19, hygiene and safety measures are essential.
4. **Efficiency**: To increase the efficacy of computer-human interaction, intuitive hand gestures or other non-physical inputs are utilized to allow users to regulate diverse functions and applications on a computer.

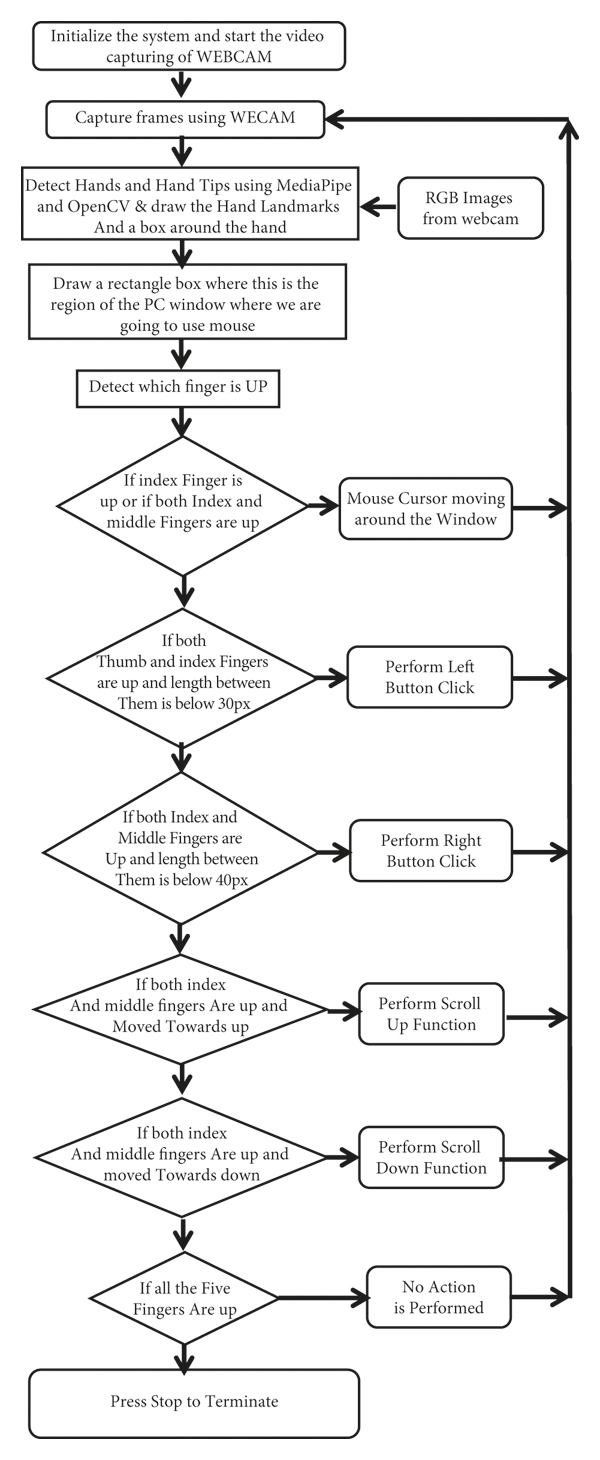
**Chapter 4**

**Packages Used**

1. **OpenCV**: OpenCV, short for Open Source Computer Vision Library, is a widely used open-source computer vision and machine learning software library. It provides a comprehensive suite of tools and functions for real-time image processing and analysis, enabling developers to create applications that can interpret visual data in various forms, such as images and videos. With its extensive collection of algorithms, OpenCV facilitates tasks like object detection, face recognition, image segmentation, and more, making it a fundamental resource for researchers, developers, and enthusiasts in the fields of computer vision, robotics, and artificial intelligence. Its cross-platform compatibility, extensive documentation, and support for multiple programming languages, including C++, Python, and Java, have contributed to its popularity and widespread adoption in diverse domains, from academic research to industrial applications.
2. **MediaPipe**: Mediapipe is an open-source framework developed by Google that facilitates the development of customizable machine learning pipelines for various media processing tasks, including but not limited to video analysis and real-time data processing. It provides a comprehensive set of pre-built, efficient, and easy-to-use building blocks for tasks such as hand tracking, pose estimation, facial recognition, and object detection, utilizing computer vision and machine learning models. With its modular design and cross-platform support, Mediapipe enables developers to create complex pipelines for diverse applications, spanning from simple video effects to advanced augmented reality experiences, thereby fostering innovation in the fields of computer vision, robotics, and multimedia processing. Its flexibility and extensive documentation make it a popular choice among developers for implementing real-time, high-accuracy, and low-latency media processing applications.
3. **PyAutoGUI:** PyAutoGUI is a Python library that enables programmatically controlling the mouse and keyboard. It provides a simple and cross-platform way to automate tasks by simulating mouse and keyboard inputs. With PyAutoGUI, users can automate various tasks such as GUI interaction, automating key presses, and controlling the mouse cursor's position. It also supports taking screenshots and manipulating images. The package is easy to use and works on various operating systems, including Windows, macOS, and Linux, making it a versatile tool for automating tasks and creating simple GUI applications. However, users should exercise caution when using PyAutoGUI for automation, as it can perform actions that may have unintended consequences if not handled carefully.
4. **Math**: The math module is a standard module in Python and is always available. To use mathematical functions under this module, you have to import the module using import math. It gives access to the underlying C library functions.
5. **Enum**: Enum is a class in python for creating enumerations, which are a set of symbolic names (members) bound to unique, constant values. The members of an enumeration can be compared by these symbolic names, and the enumeration itself can be iterated over.
6. **Comtypes**: The Comtypes package is a Python library that facilitates the interaction with the Component Object Model (COM) and ActiveX objects. It enables seamless integration between Python and the Windows-based COM components, allowing for the manipulation and automation of various software applications and services.
7. **PyCaw:** The PyCaw package, also known as Python Core Audio Windows, is a Python library that facilitates easy access to the Windows Core Audio API. It offers a range of functionalities for interacting with the audio subsystem on Windows operating systems, enabling users to control and manipulate audio devices, sessions, and volume settings programmatically.

**Chapter 5**

**Methodology**

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

Figure

**5.1 The Camera Used in the AI Virtual Mouse System**

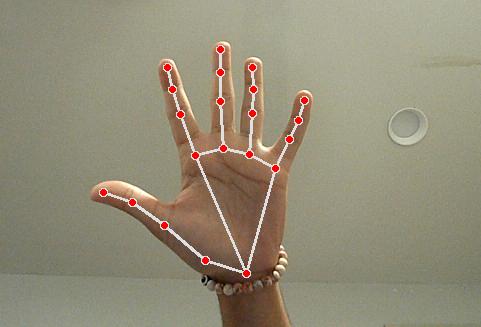


Figure Detection of palm

The AI virtual mouse system under consideration relies on frames captured by the laptop or PC's integrated webcam. Leveraging the capabilities of the Python computer vision library OpenCV, the system initiates video capture via a designated video capture object, as depicted in Figure 2. Through this process, the webcam continually captures frames, subsequently transferring them to the AI virtual system for further processing.

##### 5.2 Capturing the Video and Processing

def findHands(self, img, draw = True):

imgRGB = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

self.results = self.hands.process(imgRGB)

##### 5.3 Detecting Which Finger Is Up and Performing the Particular Mouse Function

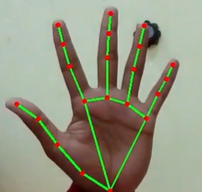
##### 

Figure Finger ID tip enumeration

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, as shown in Figure 3, and according to that, the particular mouse function is performed.

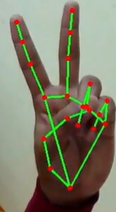
**5.4 Mouse Functions Depending on the Hand Gestures and Hand Tip Detection Using Computer Vision**

**No action:**



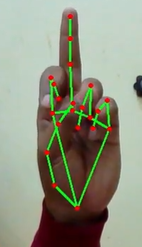
To do nothing, if all five fingers are up, then no action operation is performed.

**Move cursor:**

****

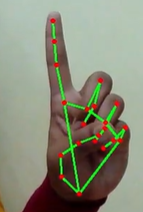
To move the cursor, index and middle finger both should be up and others closed.

**Left Click:**



If the Index finger is closed and the Middle finger is up, then the Left Click operation is performed.

**Right Click:**



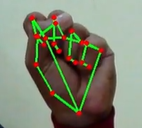
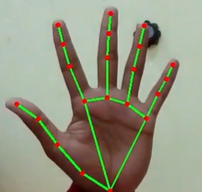
If the Middle finger is closed and the Index finger is up, then the Right Click operation is performed.

**Double Click:**



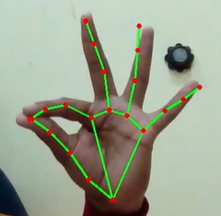
If Middle and Index finger are touching to each other, then the Double Click operation is performed. The distance between both the fingers should be 0px(pixels)

**Drag/Drop:**

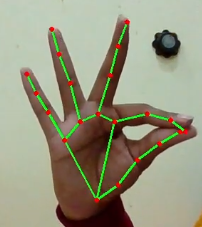
If all five fingers are held/closed together and move the hand, then the Drag Action operation is performed. If all fingers are released/opened after moving the hand in the Drag Action, then the Drop Action is performed.

**Volume Control:**



If Index and Thumb finger are pinched together and other three fingers are up, when the hand (right) is moved up and down, then the Volume Control operation is performed

**Scrolling:**

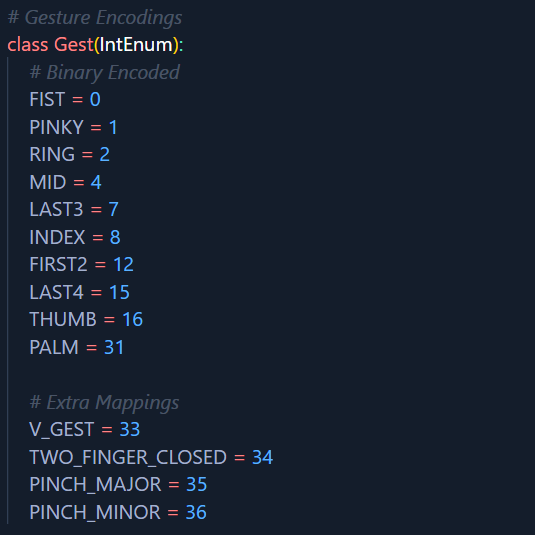
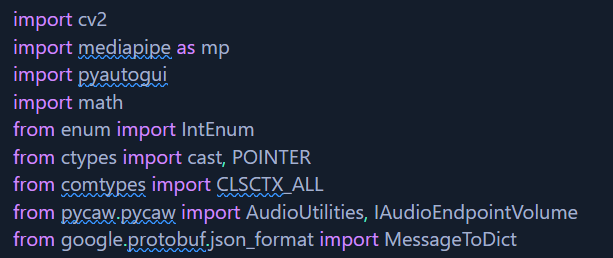


If Index and Thumb finger are pinched together and other three fingers are up, when the hand (left) is moved up and down, then the Scrolling operation is performed.

**Chapter 6**

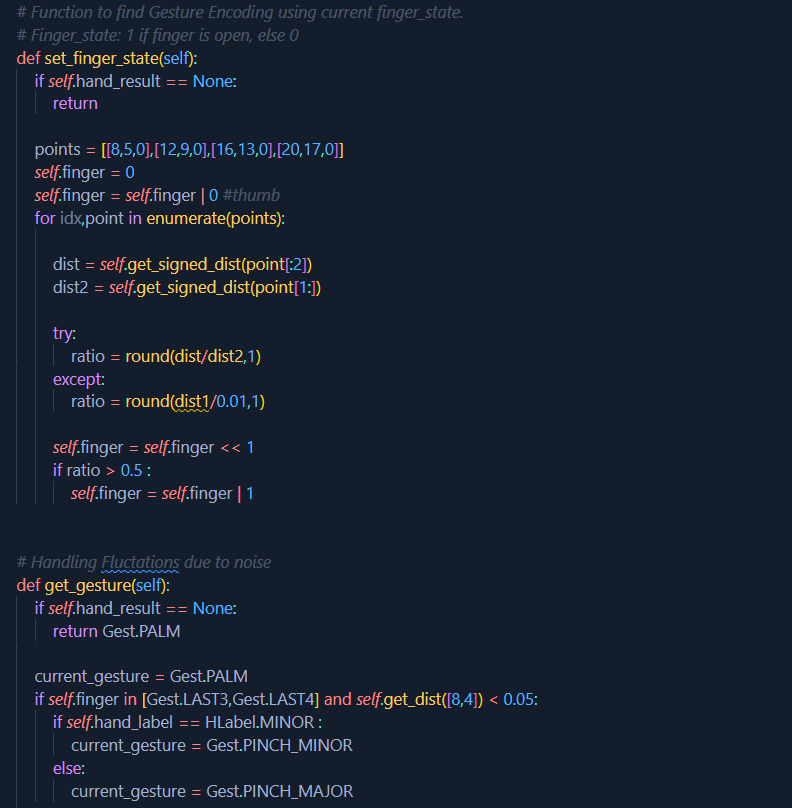
**Implementation Details**

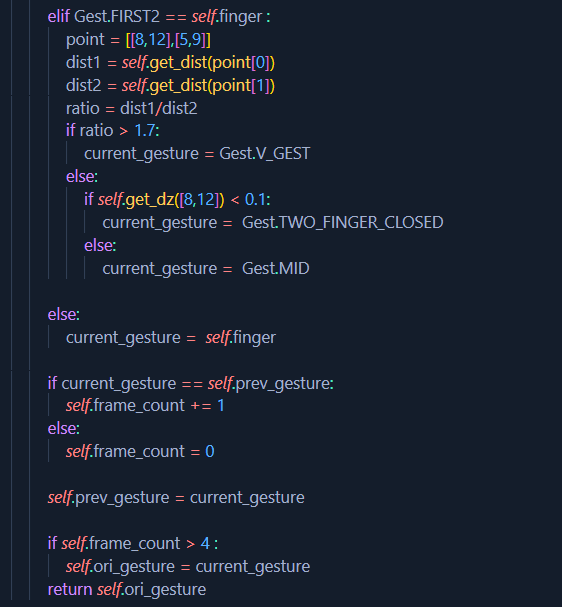
6.1 **Initialization:**

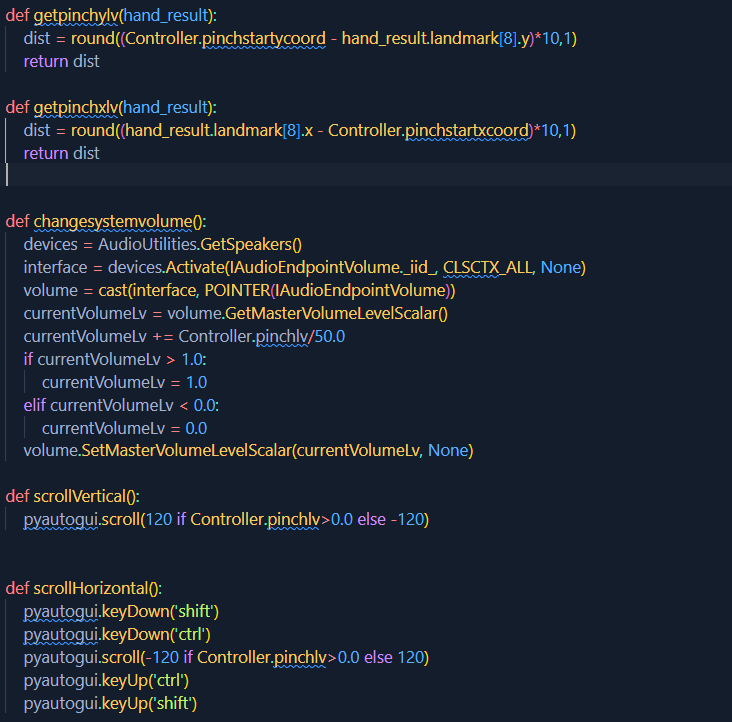


6.2 **Convert Mediapipe Landmarks to recognizable Gestures:**



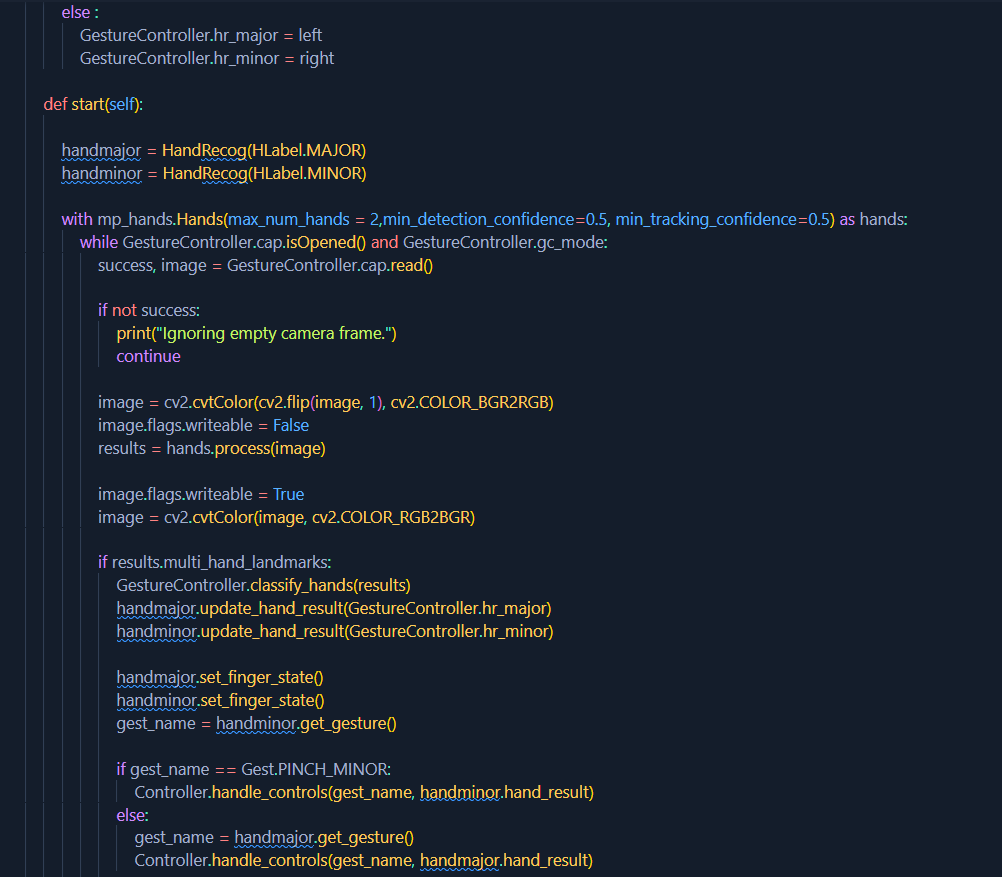


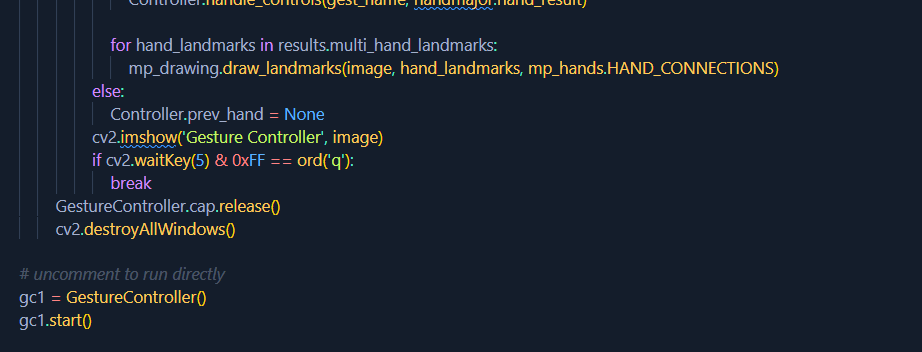




6.3 **Main class:**







**Chapter 7**

**Results and Accuracy**

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 fingertip 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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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%** |

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 Table.

**Chapter 8**

**Project Management**

The project was initiated on **15-Aug-2023** and successfully concluded on 30-**Oct-2023**, spanning a total duration of 11 weeks. The timeline was divided into several key phases, including Selecting, Studying, Implementing, Testing & Debugging, Building report/ppt and Finalisation.

**Chapter 9**

**Conclusion**

Based on the model's outcomes, it can be concluded that the proposed AI virtual mouse system exhibits **superior performance** and **accuracy** in contrast to existing models, effectively addressing many limitations found in current systems. Given its high accuracy, the AI virtual mouse holds promise for real-world applications and can contribute to reducing the spread of COVID-19 by enabling touchless interaction through hand gestures instead of relying on traditional physical mice.

However, the model does encounter certain limitations, such as a slight reduction in accuracy during **Volume functions and scrolling window**. To address these issues, our next focus will involve enhancing the finger-tip detection algorithm to yield more precise and reliable results.

**References**

* S. Shriram, B. Nagaraj, J. Jaya, S. Shankar, P. Ajay, "**Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread**", *Journal of Healthcare Engineering*, vol. 2021, Article ID 8133076, 8 pages, 2021.
* [MediaPipe Framework in Python  |  Google for Developers](https://developers.google.com/mediapipe/framework/getting_started/python_framework)
* [Mouse Control Functions — PyAutoGUI documentation](https://pyautogui.readthedocs.io/en/latest/mouse.html)