Tribhuvan University

Faculty of Humanities and Social Sciences

Ratna Rajya Laxmi Campus

AI VIRTUAL MOUSE

A PROJECT REPORT

Submitted to

Department of Computer Application

Ratna Rajya Laxmi Campus

In partial fulfillment of the requirements for the Bachelors in Computer application

Submitted by

Pradeep Bhatt (T.U. Registration no.:6-2-40-41-2019)

Manish Ghimire (T.U. Registration no.: 6-2-40-29-2019)

2023

Under the supervision of

S.K. Maharjan

Tribhuvan University

Faculty of Humanities and Social Sciences

Ratna Rajya Laxmi Campus

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by PRADEEP BHATT and MANISH GHIMIRE entitled **"AI VIRTUAL MOUSE"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

S.K. Maharjan
SUPERVISOR

Department of Computer Application

Ratna Rajya Laxmi Campus

Putalisadak (Exhibition Road), Kathmandu, Nepal

Tribhuvan University

Faculty of Humanities and Social Sciences

Ratna Rajya Laxmi Campus

LETTER OF APPROVAL

This is to certify that this project prepared by PRADEEP BHATT and MANISH GHIMIRE entitled **"AI VIRTUAL MOUSE"** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

---------- Supervisor Department of Computer Application Putalisadak (Exhibition Road), Kathmandu, Nepal	---------- Program Co-ordinator Ratna Rajya Laxmi Campus Putalisadak (Exhibition Road), Kathmandu, Nepal
---------- Internal Examiner	---------- External Examiner

Acknowledgement

For the partial fulfilment of the project in BCA 6th Semester, we would like to express our sincere gratitude to everyone who have contributed their valuable time in helping us to develop this project. There were times when we thought that this project was too difficult but we are really thankful to our supervisor **Mr. S.K. Maharjan** who continuously inspired us to give our best and work on the project to complete it. Without his vital role and direction, we would not have completed this project. Similarly, we are equally thankful to our program coordinator **Mr. Bhupendra Ram Luhar** who helped us a lot with our project as well. We could have barely done anything without the help of our lecturer who has helped us throughout and helped us understand and solve our errors as well. We would also like to include the continuous support of our classmates and thank them for always inspiring us to continue with the project and encouraging throughout out project development journey here in Ratna Rajya Campus. It is because of all of you that we did not stop and brought the project here today.

Also, our sincere gratitude and appreciation goes to Ratna Rajya Laxmi Campus for providing the platform where we were able to cherish and nurture our compassion, improve our technical skills and for great support and all necessary academic materials that has taught us great helpful academic as well as life lessons. In the end, we would also like to thank the Tribhuvan University for giving us this opportunity via the course of Computer Application to help us understand the project ethics at this early stage and help to evaluate our knowledge and expand it a little more.

Abstract

This project suggests a method for controlling the cursor's position with just your hands, without the use of any electronic devices such as a mouse or a touchpad. Simple functions such as clicking and dragging things on the screen will be performed using a variety of hand gestures. As an input device, the suggested system will just require a webcam. The suggested system will require the use of OpenCV and Python as well as other tools. The camera's output will be presented on the system's screen so that the user can further calibrate it. NumPy, AutoPy, and mouse are Python dependencies that will be utilized to create this system.

Keywords: ***OpenCV, NumPy. Calibrated, Gesture, Virtual Mouse, Hand Gestures, Image capture, Processing, Masking, DnD Frame***

Table of Contents

List of figures

List of tables

Chapter 1: Introduction

1.1 Introduction

It has been a very long time since humans have used hand gestures to communicate with one another in society. Handshaking thumbs up and thumbs down signs have always been present in the surroundings. As a result, why not apply it to the machines to be used? Real gesture is shown in this work. A low-cost USB web camera (or built-in web camera) is included in the initial setup to provide input to the system. Because it employs a camera to track hands, it's also known as hardware.

The main goal of the study work is to describe the data in such a way that it can be simply interpreted and processed by the system. To improve hygiene, reduce human touch. It focuses on extracting features from human hands and then matching those features to recognize hand movement.

Required attributes:

- User-friendly and portable.
- Handle simple operations like as left clicking, minimizing/maximizing, and closing.
- In the case of a laptop, there is no external hardware.

Virtual mouse, remote controls, sign-language recognition, and immersive gaming technologies are all examples of practical applications where fingertip sensing is used. As a result, one of the key goals of vision-based technology in recent decades has been virtual mouse control by picture fingertip detection, especially with classic red-green-blue (RGB) cameras. Gesture-based interface is offered in which users interact with a computer by detecting fingertip detection in RGB inputs in this article. Finally, the fingertip location is transferred to RGB pictures to operate the mouse cursor based on a virtual screen. Our study looks at three computer mouse functions: mouse movement, left-clicking, and right-clicking. The system runs on a single low-cost CPU without the use of a graphics processing unit (GPU), features real-time detection, and can run on a variety of resolution computer screens.

1.2 Problem Statement

The proposed AI virtual mouse system can be used to solve problems in the real world, such as instances where there isn't enough space to use a physical mouse or for people with hand problems who can't use a physical mouse. Furthermore, in the COVID-19 situation, it is not safe to use devices by touching them because this could result in the virus spreading, so the proposed AI virtual mouse can be used to overcome these issues because hand gesture and hand Tip detection is used to control the PC mouse functions by using a webcam or a built-in camera.

1.3 Objective

The major objectives of our project on AI Virtual Mouse can be highlighted below.

- To reduce the physical touches.
- To implement AI Virtual Mouse using transformational algorithm.

1.4 Scope and Limitation

Because deciding which method, procedure, or approach is best is difficult, numerous articles are assessed and classified depending on the methodologies used in each approach. When available, parameters such as speed, accuracy, performance, image size, and platform are presented for each approach. Commercial product surveys are beyond the scope of this study since, for promotional purposes, these items often claim greater accuracy than is achieved. The study finishes with a review of what isn't incorporated in AI Virtual Mouse and what kind of research is possible.

The proposed AI virtual mouse has various flaws, such as a slight loss of accuracy when using the right click mouse function, and the model has some difficulty selecting text by clicking and dragging. These are some of the drawbacks of the proposed AI virtual mouse technology, which will be addressed in future research.

Furthermore, the proposed system can be extended to handle virtual keyboard and mouse functionality, which is another prospective use of Human-Computer Interaction (HCI).

Still there are limitations like:

- While the current application appears to be feasible and user-friendly, the constant upright arm is not appropriate for daily use.

- Designing an independent gesture vocabulary framework could be another important aspect of the related development.
- Working in low light without the use of an expensive low-light camera or light source.

1.5 Development Methodology

This project will be developed by using Waterfall Methodology. This project has ample time, fixed requirements, and well-understood technology so to build this system, waterfall methodology can be used.

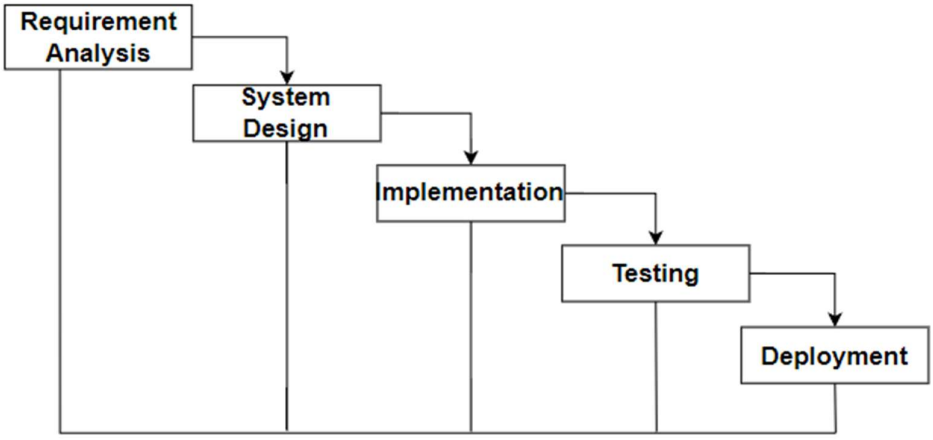


Figure 1.1 Waterfall methodology

1.6 Report Organization

The report document contains five chapters,

Chapter 1 Starting with the introduction with problem statements and objectives of the project.

Chapter 2 Describes background study and literature review.

Chapter 3 Includes the system analysis and design.

Chapter 4 Explains about the tools that are used on our project's front end, back end, and purpose of it. The modules used and testing are also explained in this part.

Chapter 5 Discusses the conclusion of how the project is accomplished, its findings. Also, the recommendation for future enhancements of the project is discussed in this chapter.

In conclusion, this chapter overview's purpose of doing this project including its scopes and objectives.

Chapter 2: Background Study and Literature

2.1 Background Study

As its application varies from movie theaters to presenting huge data on large screens, the AI Virtual Mouse plays a critical role in handling challenges such as ticket booking and display screen control. Most systems have limitations, such as low precision, and differing illumination conditions can have a significant impact on the total identification rate. Different AI Virtual Mouse techniques are discussed, with success rate and processing time as parameters. With the rise in COVID instances, social distancing and limiting touch has become a severe concern for a big number of people throughout the world. The ATM systems in most movie theaters are unsafe since they may be contaminated. And most persons with restricted hand movement have a difficult time.

2.2 Literature review

There have been some analogous virtual mouse works that use hand gesture detection by wearing a glove in the hand and using color tips in the hands for gesture recognition, but they are not as accurate in mouse functionalities. Because of the gloves, the recognition is not as exact; also, the gloves are not suitable for some users, and in some circumstances, the recognition is not as accurate due to the failure of color tip detection. The hand gesture interface has been detected using a camera in some cases.

Quam presented an early hardware-based system in 1990, which required the user to wear a DataGlove. Although Quam's proposed approach produces more accurate, results, it is impossible to conduct certain of the gesture commands with it. [1]

In 2010, Dung-Hua Liou, ChenChiung Hsieh, and David Lee presented a research project on“A Real-Time Hand Gesture Recognition System Using Motion History Image.” The model's biggest flaw is its inability to handle increasingly complex hand gestures. [2]

In 2013, Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil suggested a research project on“Cursor Control System Using Hand Gesture Recognition.” The constraint in this study is that saved frames must be processed for hand segmentation and skin pixel recognition. [3]

Vinay Kr. Pasi, Saurabh Singh, and Pooja Kumari introduced the idea in 2016 “Cursor Control using Hand Gestures” in the Journal of the IJCA. The system suggests that different

bands be used to conduct distinct mouse tasks. The drawback is that mouse functions are dependent on distinct colors. [4]

In the year 2018, Chaithanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS proposed "Virtual Mouse Using Hand Gesture" where color-based model detection is used. However, just a few mouse functions are used. [5]

Chapter 3: System Analysis and Design

3.1 System Analysis

System analysis is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified, and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

During the process of system development, first the webcam must be fully functioning. The gestures are trained in various angles for the dataset and are assigned multiple functions. Then, with the help of various functions, when user shows certain gestures in front of the webcam, several functions are carried out according to the functionalities of the gestures recognized. The user with the help of GUI (Graphical User Interface) can open the virtual mouse or hand gesture detection system from where he/she can control the mouse functionalities on the screen by moving index and middle finger performing various pre-trained gestures.

3.1.1 Requirement analysis

Requirement Analysis is one of the initial tasks performed during designing a program. It is usually composed of studying the existing system, data collection.

i. **Functional requirement**

The functional requirements specify the services that the system should provide, how the system should react to inputs and how the system should behave situations. The functional requirement of this project is:

In functional Requirement we have kept in mind all the functionalities that is offered by system and constraints that needed to be followed.

Functional requirements of our system are listed below:

- The user can open webcam and use hand gestures to control the mouse actions on the system.

Use Case Diagram: A use case diagram is a graphical depiction of a user's possible interactions with a system.

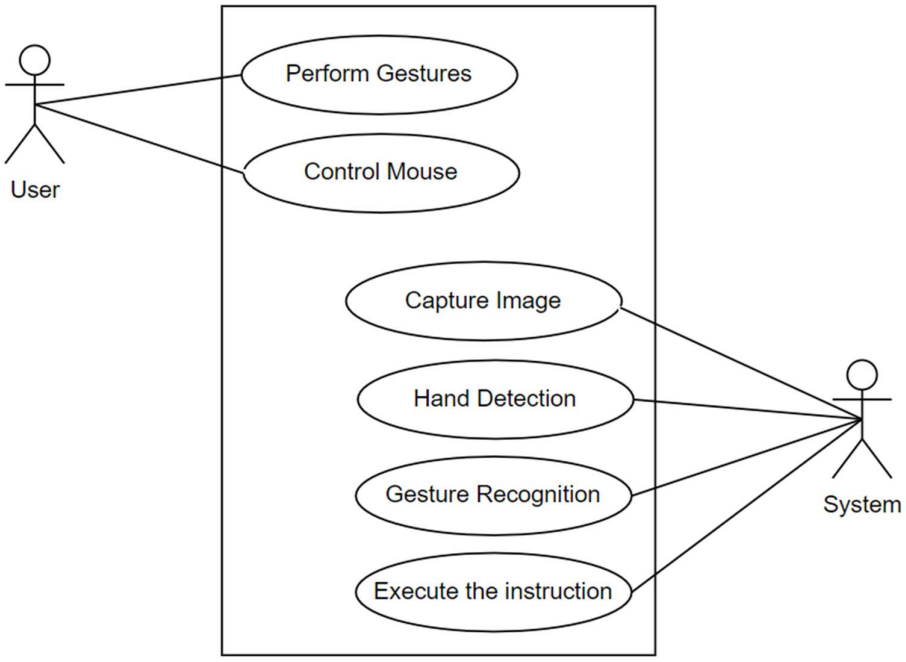


Figure 3.1: Use case diagram

ii. **Non-Functional requirement**

- Requirement: Availability
  Description: The system should be available 24/7.

- This system must be able to deliver all the functional requirements mentioned in the document.
- This system must be developed or built in a user-friendly manner such that even a common person can use it easily.
- This system must provide accurate result so that user can trust our system.
- System should be able to identify hand gestures quickly and accurately.

- System should run smoothly even with high traffic rate.

3.1.2 Feasibility Analysis

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

i. Technical Feasibility

The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it. Install all upgrades framework into the web-based application. This application depends on web, database. Enter their attendance and generate report to excel sheet.

ii. Operational Feasibility

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance includes issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel.

iii. Economic Feasibility

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the paperwork completely. The system is also time effective because the

calculations are automated which are made at the end of the month or as per the user requirement.

iv. Schedule Feasibility

The timelines /Deadlines are analyzed for proposed project, and we have enough time required for completion of this project. Thus, this project is schedule feasible. The required Gantt chart of our project is already shown below.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	16th
Study and Analysis	6 weeks															
Data Collection					3 weeks											
Implementation							8 weeks									
Testing													4 weeks			
Documentation														3 weeks		
Review															2 weeks	
Presentation															2 weeks	

3.1.3 Data Modeling (ER-Diagram)

We do not require any database for this project as it uses live hand co-ordinates.

3.1.4 Process Modeling using DFD

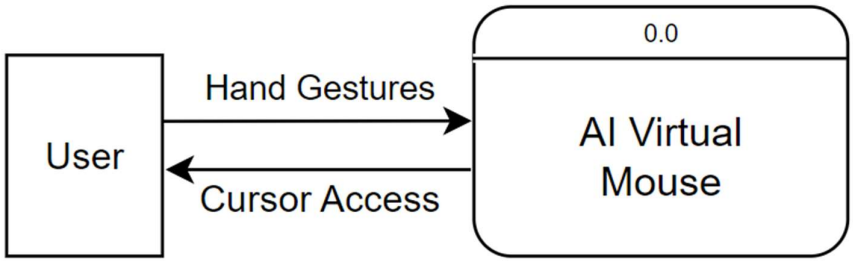


Figure 3.2 Context Level Diagram

This above diagram is the Context level diagram for AI virtual mouse, it detects the user's hand gestures to operate a mouse and enable functioning.

3.2 System Design

3.2.1 Architectural Design

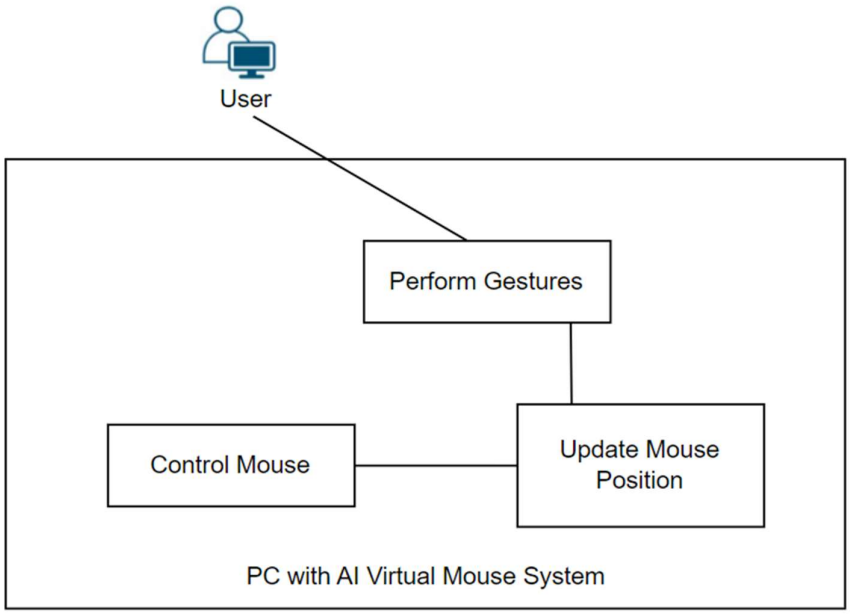


Figure 3.3 Architectural Design

The figure above represents an architectural design of a system where user performs gesture to control mouse and mouse position is updates based on gestures.

3.2.2 Database Schema Design

We do not require database schema as we have no database tables.

3.3 Description of Algorithm

Transformational algorithm:

A transformational algorithm used in an AI virtual mouse is a set of rules or procedures that transform the input data, such as hand movements captured by a camera, into a desired output format, such as the movement of a cursor on a computer screen.

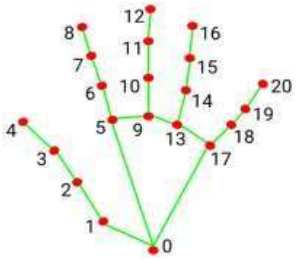
The transformational algorithm used in an AI virtual mouse typically involves several steps. First, the hand movements are captured by a camera and processed to identify the

hand's position and movements. This step might involve image processing techniques such as edge detection or object recognition to identify the hand's location and track its movement.

Once the hand movements have been captured and processed, the transformational algorithm uses mathematical operations to convert the hand movements into cursor movements on the computer screen. For example, the algorithm might map the position of the hand to a corresponding position on the screen, or it might use machine learning techniques to predict the desired cursor movement based on the history of the hand movements.

The transformational algorithm in this AI Virtual Mouse project works by mapping the movement of the finger in the webcam frame to the movement of the computer's mouse cursor on the screen. Here's a breakdown of how it works:

1. **Hand Tracking**: The project uses the MediaPipe library to detect and track the position of your hand, including the landmarks (key points) of your fingers, in real-time using the webcam feed.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

2. **Finger Detection**: It specifically pays attention to the position of your index finger and detects whether it's raised or not.

```
x1, y1 = self.lmList[p1][1:]
```

where, **self.lmList** is a list that contains information about landmarks detected by the hand tracking model.

p1 is a variable representing the index of a landmark in the **self.lmList**.

Check if the fingers are raised based on the tip and lower knuckle positions:

```
self.tipIds = [4, 8, 12, 16, 20];

    for id in range(1, 5):

        if self.lmList[self.tipIds[id]][2] < self.lmList[self.tipIds[id] - 2][2]:

            fingers.append(1)  # Finger is raised

        else:

            fingers.append(0)  # Finger is not raised

    return fingers
```

3. **Coordinate Conversion**: If your index finger is raised and other fingers are not, the algorithm converts the coordinates of your finger's position in the webcam frame to coordinates on your computer screen. This allows it to map your finger's movement to the mouse cursor's movement.

```
x3 = np.interp(x1, (frameR, wCam - frameR), (0, wScr))

y3 = np.interp(y1, (frameR, hCam - frameR), (0, hScr))
```

4. **Smoothening**: To provide a smoother mouse movement experience, the algorithm incorporates smoothening. This means that sudden, jerky hand movements are dampened, resulting in a more controlled mouse cursor motion.

```
clocX = plocX + (x3 - plocX) / smoothening

clocY = plocY + (y3 - plocY) / smoothening
```

5. **Mouse Control**: Finally, it moves the computer's mouse cursor based on your index finger's position on the webcam feed, taking into account the smoothened movement and the screen's dimensions.

```
 autopy.mouse.move(wScr - clocX, clocY)
```

Chapter 4: Implementation and Testing

4.1. Implementation

The process of adopting and integrating a software application into a company workflow is known as software implementation. Depending on the scope of the project, integrating new tools and technologies into an organization can be difficult. The project should be chosen before it is implemented by examining needs, money, potential benefits, barriers, and so on. After you've decided on a solution, you may start putting it into action.

The first step is to turn on the camera so that it can send input to the system. To do so, we must first assign the camera's resources to a variable.

1. Hand position and co-ordinate extraction

The hand position and its coordinates are extracted in the second stage. After that, MediaPipe is used to detect fingers.

Figure 4.1 Hand Position Co-ordinates

Each finger and its character have its own id, and one can tell which finger is up and which is down by looking at the id(s). Now two conditions must be considered:

- Is Index only going up?

Yes:

If only the index finger is up, it extracts the index finger's movement and moves the cursor accordingly.

No:

It expects that movement is not required and catches mouse properties such as right click, left click, and so on if fingers are also up.

2. **Movement of the cursor**

Because the online camera and screen resolution are not the same, we must translate the co-ordinates of the finger collected by the web camera into actual screen size. The mouse cursor is moved in accordance with the index finger movement after the co-ordinates have been computed.

3. **Feature of mouse button**

The left and right mouse buttons are clicked by measuring the distance between two fingers, which could be the index and middle finger or the index and pinky finger. Furthermore, zoom and scroll are performed by measuring the distance between the thumb and index fingers, as well as the distance between the starting and end states of the full palm.

4.1.1. Tools Used (CASE tools, Programming languages, Database platforms)

(Software Requirement Specifications) SRS of Motion Tracker Application:

Application's Goals: Instead of using a conventional mouse, users can use their finger as a mouse by clicking on the TRACK option that appears after running the application. Whatever direction finger is given, the cursor on the system moves and the user can complete their work.

How does the application function: This program is a mix of UI and AI, with code ranging from camera access to finger tracking motions. To begin, the user must first run the.exe file. After opening the file, the user is presented with a user interface, which includes the option of TRACK. When the user selects this option, our backend application is launched, which includes code for accessing the camera and tracking finger gestures.

A need for functionality: After constructing this application, users should be able to utilize the Motion Tracker Application to access their systems. The calculations necessary for this application are all connected to motion detection procedures, and data processing is done with the help of various Python libraries such as NumPy and media-pipe.

Interface specifications: With the use of Python's TKINTER, this application connects with the user interface and Python code. Tkinter is a Python binding for the Tk graphical user interface toolkit. It is Python's de facto standard GUI and the primary Python interface

to the Tk GUI toolkit. Tkinter is standard with Python installations on Linux, Microsoft Windows, and Mac OS X.

Overview: Application is being work by using AI and the web. Implement a code that allows the camera to recognize and respond to each finger movement. After completing this project, users will be able to access their system using only their finger and the device's camera. The system should be able to track hand motion with the help of a web camera. It also comes with certain Python libraries preconfigured, making it simple to use.

Draw.IO: Draw.io was used to draw all the diagrams for the project. Draw.io is used to create diagrams for conveying functional requirements (use-case diagrams), system structure and design (ER diagrams, activity diagrams), and system structure and design (ER diagrams). These diagrams are necessary for a more in-depth understanding of our system.

NumPy: NumPy is the most important Python package for scientific computing. It has, among other things, a powerful N-dimensional array object and advanced (broadcasting) methods. iii. C/C++ and Fortran code integration tools iv. practical knowledge of linear algebra, Fourier transform, and random numbers PURPOSE: To retrieve numerical data.

OpenCV: OpenCV (Open-source computer vision) is a programming library geared mostly toward real-time computer vision. TensorFlow, Torch/PyTorch, and Caffe are among the frameworks it supports. USE: Image manipulation.

AutoPy: AutoPy is a Python GUI automation package that is simple and cross-platform. It has controls for the keyboard and mouse, as well as features for finding colors and bitmaps on the screen and showing alarms. USE: flipping image

Programming Language used:

- **Python:** For works such as Model Creation, Model Training, Model Saving, Model Evaluation and Model Testing.

4.1.2. Implementation Details of Modules (Description of procedures/functions):

I. Interpolation

- **Numpy.interp**

One-dimensional linear interpolation for monotonically increasing sample points.

Returns the one-dimensional piecewise linear interpolant to a function with given discrete data points (*xp*, *fp*), evaluated at *x*.

II. Frame Reduction

The Frame Reduction was done to make the corner of bounding box to the corresponding corner of screen. The problem faced was while the down movement the fingertips where visible but the whole hand was overflowing the camera bounding box. Hence, to make the hand visible within the bounding box all the time Frame Reduction was implemented.

III. Data Collection

The hand tracking data used to support the findings of this study are included within the article. The study uses Google's framework; hence, no new data are needed to train the model.

4.2. Testing

4.2.1. Test Cases for Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff.

Table 1 Test case for Camera and Mouse function

Project Name: Student Attendance Facial Recognition System	
Test Case	
Test Case ID: TC_01	Test Designed by: Pradeep Bhatt
Test Priority (Low/Medium/High): Medium	Test Designed date: 2023-03-01
Module Name: Mourse Functioning	Test Executed by: Pradeep Bhatt
Test Title: Mourse Functioning	Test Execution date: 2023-07-02

Description: Test the Camera and Mouse function
Pre-conditions: A webcam should be installed with supported drivers.
Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1	Detect Finger		Fingers should be detected	As Expected i.e. Fingers detected with outlined frame	Pass
2	Index finger moves cursor		Cursor Should move	As Expected, i.e. Cursor moving	Pass
3	Middle finger Act as click		Clicked	As Expected i.e. Clicking works	Pass

Chapter 5: Conclusion and Future Recommendations

5.1. Conclusion

After analyzing the results of the virtual mouse, we can conclude that the proposed system has worked well. Since the proposed system has great accuracy, this developed AI virtual mouse can be used for places like ticket booking from screens, and, it can be used to reduce the spread of COVID-19, since the AI mouse system can be used using hand gestures without the physical touch denying the traditional physical mouse.

This mouse has some limitations:

- To interpolate the co-ordinates of current position of fingertips
- Down Movement of Hand and corresponding capture of fingertips were uneven.
- Converting co-ordinates of resolution of camera to resolution of screen

5.2. Future Recommendations

The following are the future recommendation that author would like to add in future.

- Adding more hand gesture in accordance with more finger's tips.
- Addition of interactive gestures like Zoom, Volume Control, Pan.
- More smoothening of cursor movement with improved interpolation technique.
- Deployment of Web Based interface for easy access from any kind of device.

References

[1] https://www.ijraset.com/research-paper/gesture-recognition-based-ai-virtual-mouse, "https://www.ijraset.com/research-paper/gesture-recognition-based-ai-virtual-mouse," [Online]. Available: https://www.ijraset.com/research-paper/gesture-recognition-based-ai-virtual-mouse.

[2] ". M. u. H. G. a. C. D. I. J. o. C. A. v. 1. n. 1. p. 6. O. 2. A. M. 2. 2. [. A. A. Ghodichor and B. Chirakattu, "https://www.ijcaonline.org/arc," [Online]. Available: https://www.ijcaonline.org/arc.

[3] [Online].Available: https://www.researchgate.net/publication/239525791_Cursor_Control_System_Using_Hand_Gesture_Recognition.

[4] "https://www.hindawi.com/journals/jhe/2021/8133076/," [Online].

[5] "https://ijrpr.com/uploads/V4ISSUE4/IJRPR11793.pdf," [Online].

Appendices

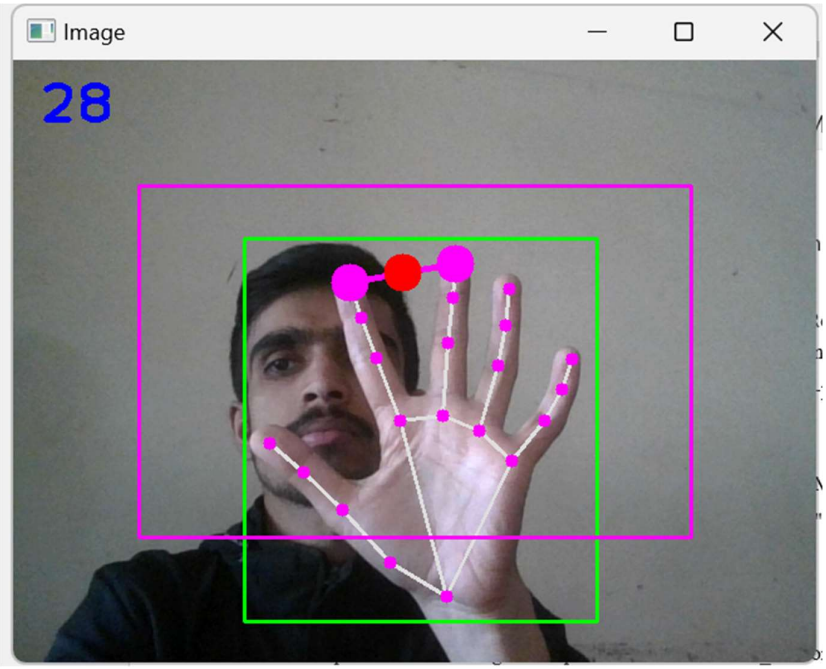


Fig: System detecting hand landmarks

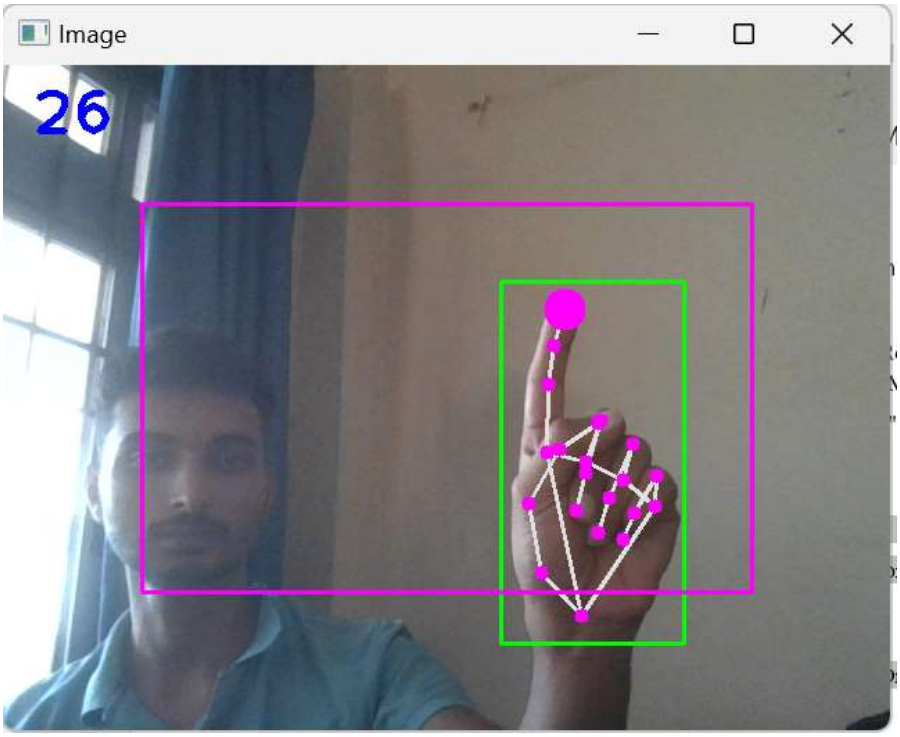


Fig: System detecting index finger for cursor movement

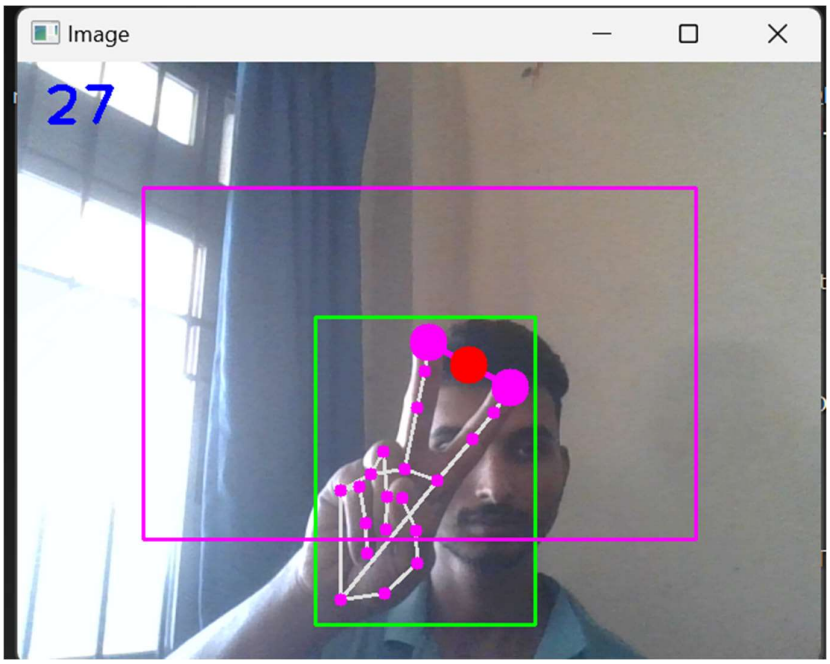


Fig: System detecting finger distance