# Gesture and Voice Controlled Virtual Mouse

A
Synopsis submitted
in the partial fulfillment of the requirements for the award of the degree of

**Bachelor of Technology**
in
**Computer Science and Engineering**

by

Under the guidance of

# Contents

# Declaration

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for requirement of any other degree.

Submitted by:

# Abstract

By controlling cursor movement with a real-time camera and microphone, this project advances the Human Computer Interaction (HCI) paradigm in the field of computer science.
The hand movement and speech is the most effortless and primitive way of communication. It's a replacement for the present ways, which entail manually moving a physical computer mouse or pressing buttons. Instead, the system controls and performs numerous mouse activities using a camera for computer vision technology and a microphone for speech recognition and processing. It can perform all functions that a physical mouse can.

The Virtual Mouse continuously gathers real-time visuals and voice commands, which are then filtered and converted in a number of steps. When the procedure is completed, the programme uses image processing and natural language processing to extract the valid command needed to complete the task.
Specially abled people with hand problems can use this virtual mouse to control the computer's mouse functionalities.

# Introduction

The most efficient and expressive way of human communication is through hand gestures and speech, which is universally accepted for communication. It is expressive enough for a dumb and deaf people to understand it. In this work, a real-world gesture system is proposed. Experimental setup of the system uses fixed position cost-effective web cam for high definition recording feature mounted on the top of the monitor of a computer or a fixed laptop camera. In addition to this, it uses a microphone to capture sound which is later processed to perform various mouse functions. Recognition and the interpretation of sign language or speech is one of the major issues for the communication with dumb and deaf people.

Python computer programming language has been used in the given project for the code, whereas OpenCV is used for computer vision to capture gestures. For hand tracking, the model in the proposed Virtual mouse system uses the MediaPipe package. The Python package Speech Recognition is used for voice instructions.

# Objective

The project's main goal is to create a hands-free virtual Mouse system that focuses on a few key applications in development. This project aims to eliminate the need for a physical mouse while allowing users to interact with the computer system via webcam and speech using various image and audio processing techniques. This project seeks to create a Virtual Mouse programme that can be used in a variety of contexts and on a variety of surfaces.

The following are the objectives of the project:

- Design for mouse operation with the aid of a webcam. The Virtual Mouse technology works with the help of a webcam, which takes real-time photos and photographs. A webcam is required for the application to function.

- The cursor is assigned to a certain screen position when the hand gesture/motion is converted into a mouse operation. The Virtual Mouse application is set up to identify the position of the mouse pointers by detecting the position of the fingertips and knuckles on a defined hand colour and texture.

- Develop a multi user independent speech recognition system that captures voice in real-time with the help of a microphone and is able to retrieve folders, sub-folders, documents, copy, paste, left click, right click and double click by taking voice command and checking its validity.

- Create a voice-activated mouse system that works in tandem with the gesture-activated system.

# Literature Survey

## What is Human-Computer Interaction (HCI)?

The study of how humans (users) interact with computers is known as human-computer interaction (HCI). It is a multidisciplinary field that deals with the design of computer technology. HCI began with computers and has now grown to embrace practically all aspects of information technology design.

## The Meteoric Rise of HCI

When personal computers first became popular in the 1980s, HCI emerged at the same time as machines like the IBM PC 5150, Commodore 64, and Apple Macintosh began to be utilised in homes and offices. For the first time, sophisticated electronic systems such as games units, word processors and accounting aids were available to general consumers for use. As a result, as computers grew in size to the point where they were room-sized, expensive tools created exclusively for professionals in specialised situations, the necessity to research human-computer interaction that was also efficient and simple for less experienced users grew in importance. Design, computer science, psychology, cognitive science, and human-factors engineering are just a few of the fields that have been incorporated into HCI.


Throughout the research on human-computer interaction, several variants of these algorithms have been developed in various fields including Engineering, Design,

social psychology, computer science, cognitive science, information security, sociology, and speech-language pathology are some of the fields covered.

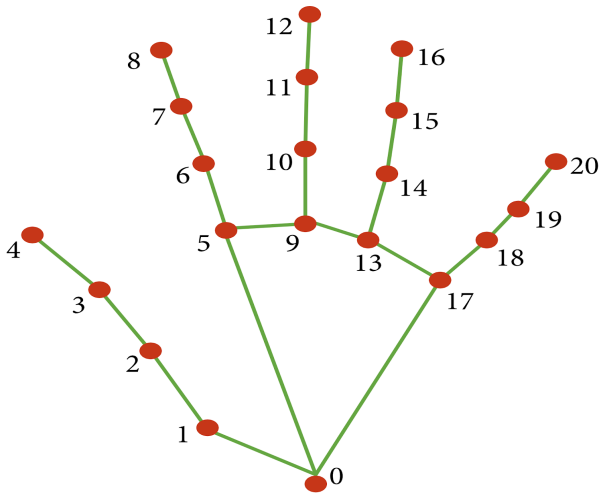
The current research aims to create algorithms that lessen human reliance on hardware and strive for a more natural method of interacting with computers through hand gestures and speech.

The OpenCV library is utilised for computer vision, while the MediaPipe framework is used to recognise and monitor hand motions. The system also employs machine learning techniques to track and recognise hand gestures and tips.

## MediaPipe

MediaPipe is a Google open-source framework that is used to apply in a machine learning pipeline. Because the MediaPipe framework is based on time series data, it is suitable for cross-platform development. The MediaPipe is a multimodal architecture that can be used with a variety of audio and video formats. The MediaPipe framework is used by developers to create and analyse systems using graphs, as well as to create systems for application development. The steps in a MediaPipe-enabled environment are carried out in the pipeline setup. The pipeline is scalable and runs on a range of platforms, including PCs, laptops, and mobile devices.
Performance evaluation, a framework for accessing sensor data, and a reusable collection of components known as calculators are the three primary components of the MediaPipe system.

In order to recognise and detect a hand or palm in real time, a single-shot detector model is used. The single-shot detector is used by MediaPipe. It is first trained for a palm detection model of hands in the hand detection module since palms of hands are easier to train and map. Furthermore, the non-maximum suppression is far more effective on small objects like hands and fists. The location of 21 joint or knuckle co-ordinates in the hand region makes up a model of hand map or landmark.




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

## OpenCV

OpenCV is a real-time computer vision library that focuses on computer vision.
Intel was the first to develop it.
Under the open-source BSD licence, the library is cross-platform and free to use.
OpenCV is written in C++ and has a C++-based user interface. Python, Java, and MATLAB/OCTAVE all have bindings.
Open-source library of computer vision, image analysis, and machine learning. To do this, it has an infinity of algorithms that allow, just by writing a few lines of code, identifying faces, recognizing objects, classifying them, detecting hand movements.

## Past Researches

We have come a long way in the field of human computer interaction. Gesture based mouse control was carried out by wearing gloves initially. Later, colour tips were also used for gesture recognition. Although such systems were not very accurate. The recognition accuracy is less due to use of gloves. Some users may not feel comfortable wearing gloves, and in some situations, recognition is not as accurate as it may be due to colour tip detection failure. Computer-based gesture detection systems have recently received some attention.

In 1990, Quam introduced a hardware-based approach that required the user to wear a DataGlove. Despite the fact that Quam's proposed method generates more precise results, certain of the gesture controls are difficult to execute with the system.

Zhengyou et al. proposed the Visual Panel interface system (2001). A quadrangle-shaped plane is used in this system, allowing the user to perform mouse operations with any tip-pointed interface instrument. Though the system can be operated contact free yet it does not solve the problem of surface area requirement and material handling.

Color tracking mouse stimulation was proposed by Kamran Niyazi et al. (2012). Using computer vision technology, the system tracks two colour tapes on the user's fingertips. One of the tapes will be used to control the cursor's movement, while the other will act as a trigger for the mouse's click events. Despite the fact that the proposed system handled the bulk of the issues, it only has a limited range of capabilities, as it can only perform fundamental actions such as cursor movements, left/right clicks, and double clicks.

To replicate click events, the system requires three fingers with three colour pointers, according to Kazim Sekeroglu (2010). The suggested system can detect pointers using colour information, track their motion, change the cursor according to the position of the pointer, and simulate single and double left or right mouse click events.

Chu-Feng Lien (2015) proposed a way for controlling the mouse cursor and click events using only one's fingertip. To interact with the system, the suggested system does not require hand motions or colour tracking; instead, it uses a feature called Motion History Images (MHI). Because the frame rates can't keep up with quickly moving objects, the proposed system can't detect them. Furthermore, because mouse click events occur when the finger is held in particular positions, this may cause the user to move their fingers constantly to avoid false alarms, which can be inconvenient.

S. Shriram(2021); the model employs the MediaPipe package for tracking the hands and the tips of the hands, as well as the Pynput, Autopy, and PyAutoGUI packages for moving around the computer's window screen and performing actions like left click, right click, and scrolling.

---

## CONCLUSION

The already proposed models made significant improvements in the human control interaction with respect to mouse functions, yet they suffered from some of the drawbacks and limitations. Requirement of gloves, complex gestures, and limited functions are some of them. Through our project we have aimed at solving the above limitations by eliminating gloves and including most of the functions with the help of simple hand gestures. We have further Integrated this virtual system with speech recognition which will input commands like cut, copy, paste, etc and process it to perform the same.

# Limitations of Physical Mouse

Despite the fact that technology has come a long way since its inception in the last decade, it still has a number of drawbacks. The following are some of the identified and generalised limitations of a mouse, or in a larger context, any physical device:

The physical mouse has a number of flaws, including the following:

- The physical mouse experiences wear and tear.
- To operate hardware and software, the physical mouse requires a particular surface.
- The physical mouse is incompatible in a variety of scenarios..
- Depending on the setting, performance varies.
- Even in today's operational conditions, the mouse has limited capabilities.
- Each period of usability for wired and wireless mice is distinct..

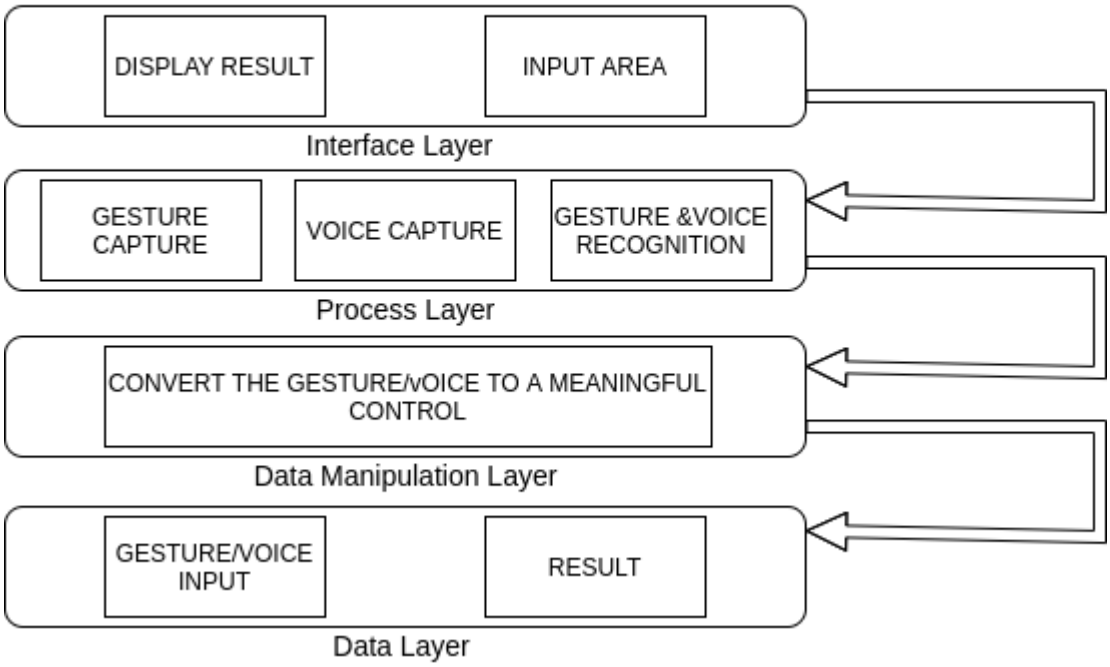
# Motivation of Virtual Mouse

We can say that Virtual Mouse will be replacing the physical mouse soon because we people are targeting towards a lifestyle where everything can be controlled remotely without the involvement of any physical device such as the mouse, keyboards, etc. Not only is using a virtual mouse convenient, but it is also cost-effective.

**Convenient**
To interact with computer we need a physical mouse which requires some additional space to place that mouse. On the other hand, virtual mouse requires only webcam to get hand landmark using mediapipe. Moreover, nowadays people are able to control their monitors and devices remotely through webcam or anyother image taking device thus removing the need to keep the physical mouse and hence controlling systems by being a feet away from device.

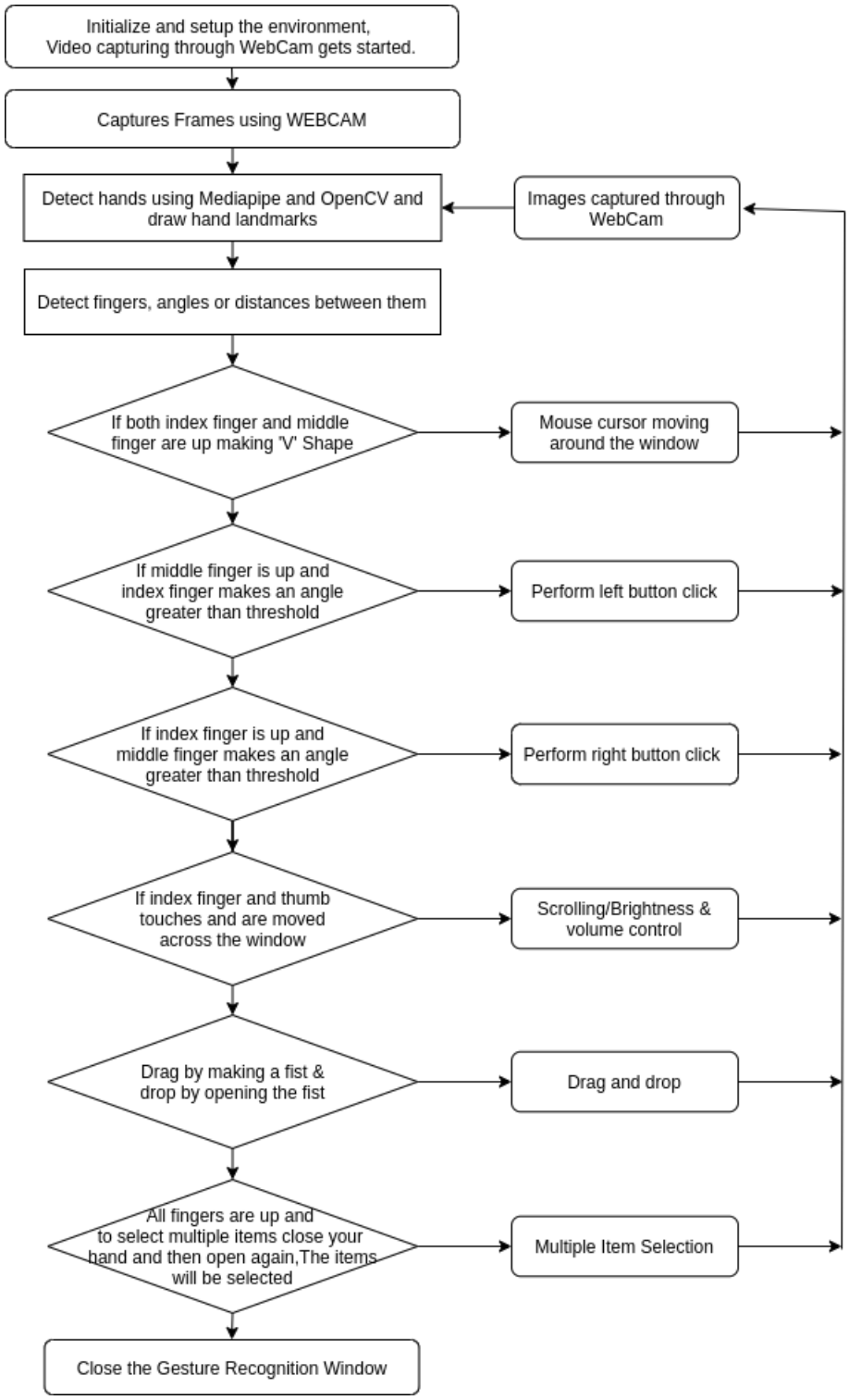
**Cost Effective**
A decent and efficient actual mouse is expensive, but since the Virtual Mouse merely requires a webcam, it is no longer necessary. Hence, nowadays we need not buy any new physical mouse and hence saving costs. In virtual mouse, we only utilise a webcam, which is already installed on many devices such as laptops, and some software that is simple to install.

# ARCHITECTURE



# Methodology

## Gesture Control



**Computer Vision application for object identification**

The frames captured by the webcam on a laptop or PC are used to create our virtual mouse environment. To capture the video object that is being formed, we used OpenCv, a Python computer vision package, and the web camera was used to begin capturing video. The web camera captures the frames and sends them to the virtual environment.

Working of OpenCV

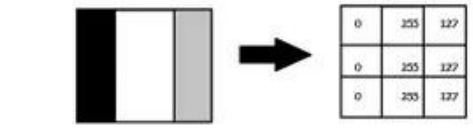
Computer works only with numbers. Everything we save in a computer like video, images, documents, etc is saved in a computer in the form of numbers.

In image processing pixels are converted into numbers.

A pixel is the smallest unit of a digital image.

The numbers can be used to calculate a number's intensity at any given pixel.

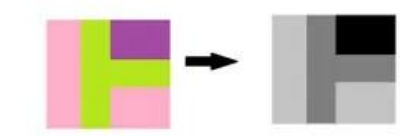
OpenCV can work in a grey scale or BGR(Blue, Green, Red) format.




Images can be classified using:

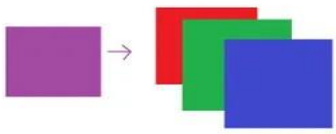
1. Gray Scale Images

Image processing using this method involves converting the image into black and white format, where black is 0 and white is 255.



2. BGR format

The photos feature three colours: blue, green, and red. The computer extracts that value from each pixel and puts the results in an array to be interpreted. Images are represented as three channels blue, green and red.



To identify the hand, the cumulative probability of B G R is employed..

**ML Pipeline(Mediapipe) for Hand Tracking and Gesture Recognition**

Mediapipe is a Machine Learning system built on the collaboration of pipeline models.

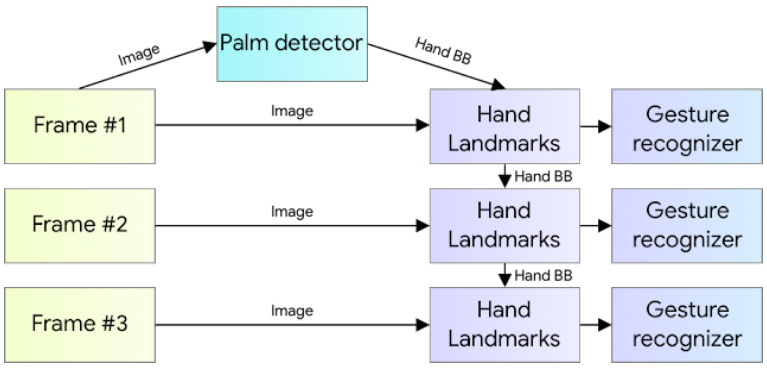
What is ML Pipeline?

A pipeline joins several stages together so that the output of one is used as the input for the next.
Pipeline makes it simple to train and test using the same preprocessing.

The hand tracking method makes use of a machine learning pipeline that consists of two models that work together:

• A palm detector that uses an aligned hand bounding box to locate palms on a whole input image.
• A hand landmark model that uses the palm detector's clipped hand bounding box to produce high-fidelity results. landmarks in 2.5D

The following is a summary of the pipeline:




Palm Detector Model

Hand detection is a tedious process as it requires identifying hands of various sizes, shapes, with deformities, etc. It is more complex than face detection as the contrast in features is far less than that in face.
We use palm detection model first as detecting palm or a fist is much easier than detecting a full hand with articulated fingers. Also palms are smaller therefore non suppression algorithm works better for it.

Hand Landmark Model

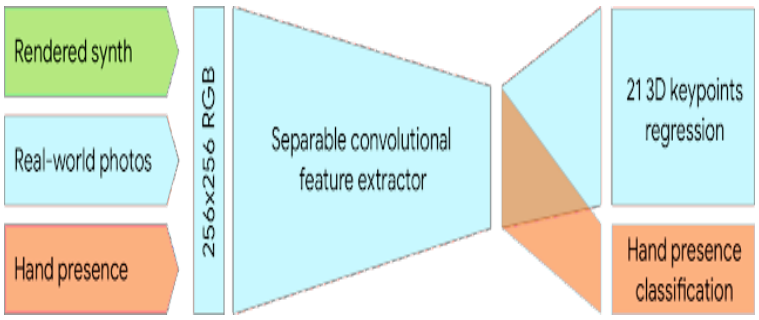
After detecting the palm using a palm detection model next a hand landmark model is used to detect 21 landmark points in 2.5 dimension. The Z depth is analysed using a Image Depth Map. The model recognises both partially and fully acclusioned hands perfectly.

The model has three outputs (see Figure 3):

1. 21 hand landmarks consisting of x, y, and relative depth.
2. A hand flag indicates the existence of a hand in the input image.
3. A binary classification of handedness, e.g. left or right hand.

The topology is the same as for the 21 landmarks. To avoid performing hand detection over and over for the entire frame, the probability of hand presence in a bounded crop is determined. The detector is triggered to reset tracking if the score falls below a threshold. We constructed a binary classification head to predict whether the input hand is left or right. Only the first frame or when the hand prediction shows that the hand is lost is the detector used.

For our project, the fingers are given Ids from 0 to 4.


Rendered synth
Real-world photos
Hand presence
256x256 RGB
Separable convolutional feature extractor
21 3D keypoints regression
Hand presence classification

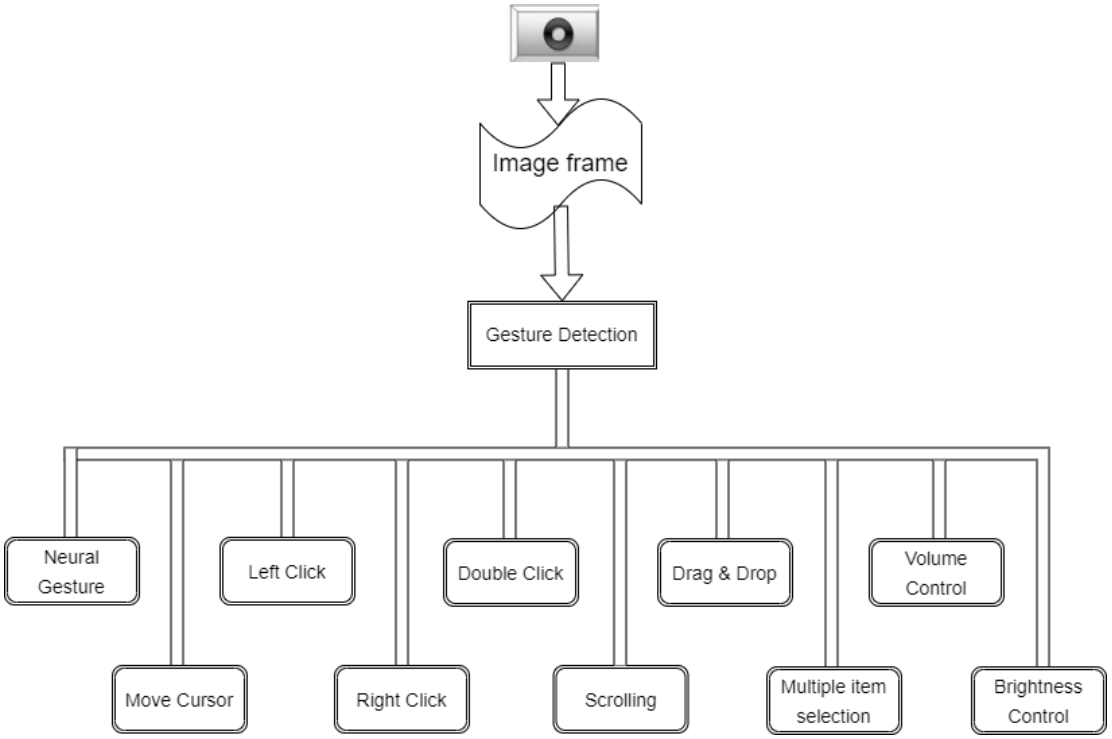
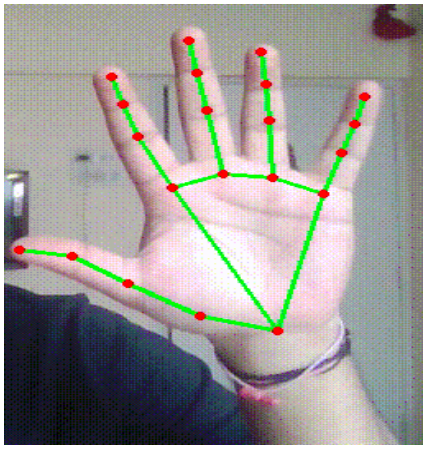


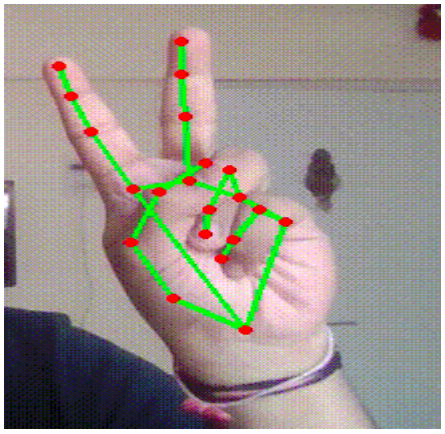
Image frame
Gesture Detection
Neural Gesture
Left Click
Double Click
Drag & Drop
Volume Control
Move Cursor
Right Click
Scrolling
Multiple item selection
Brightness Control

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

- **Neutral Gesture**

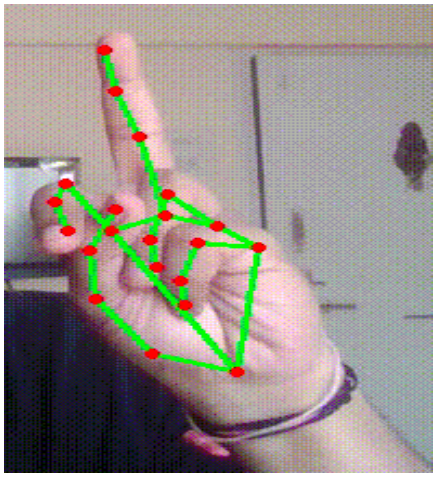


- **Move Cursor**



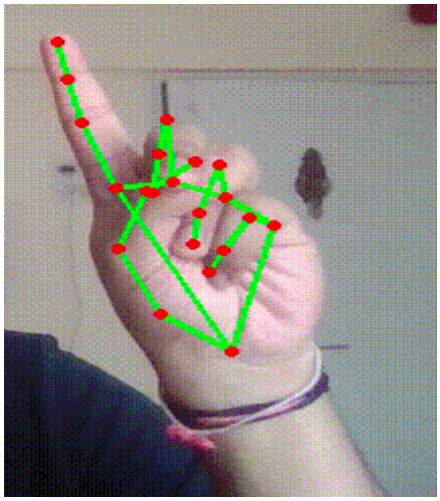
The mouse cursor is used to navigate the computer window.
If the index finger with tip Id1 and the middle finger with tip Id 2 are both up, the mouse cursor is made to move around the computer window.

- **Left Click**

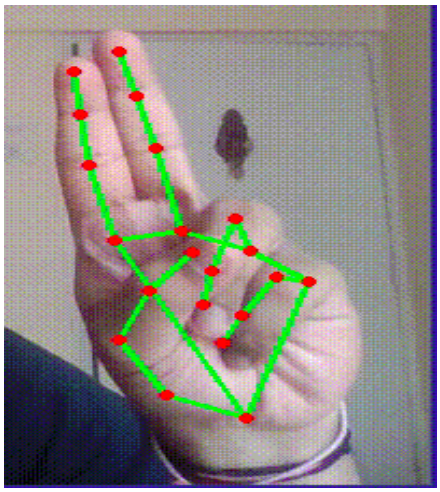


In order for the mouse to perform a left click:
In the first frame, the index finger with Tip ID 1 and the middle finger with Tip ID 2 must be up, followed by Index Finger (Tip Id 1) down and Middle finger (Tip id2) up in the second frame, with both producing an angle greater than 33.5°.

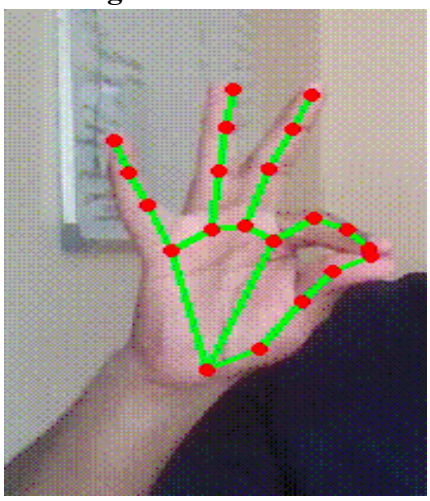
- **Right Click**



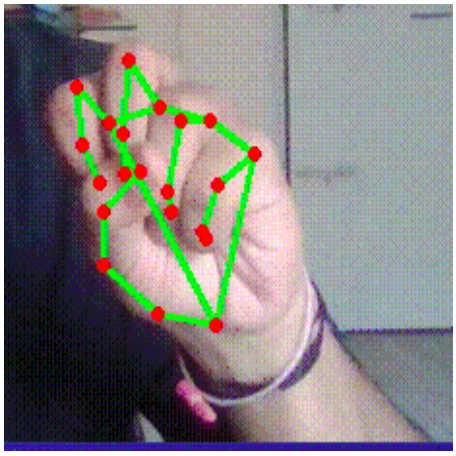
- **Double Click**

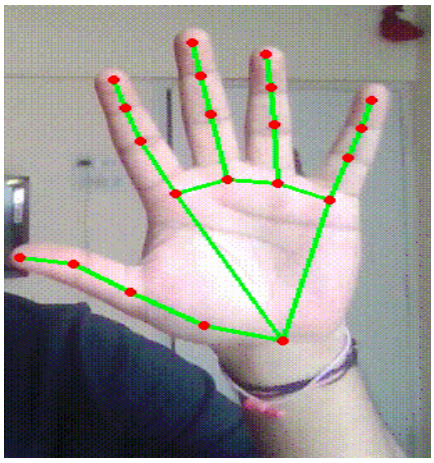


- **Scrolling**

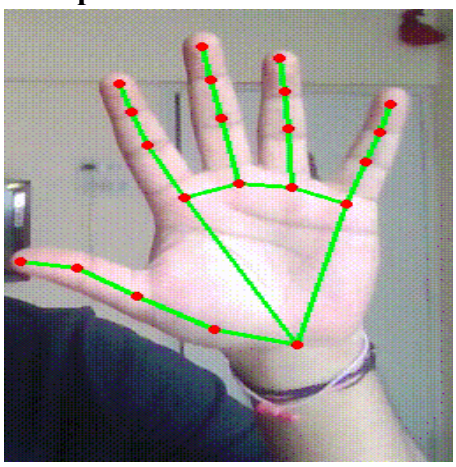


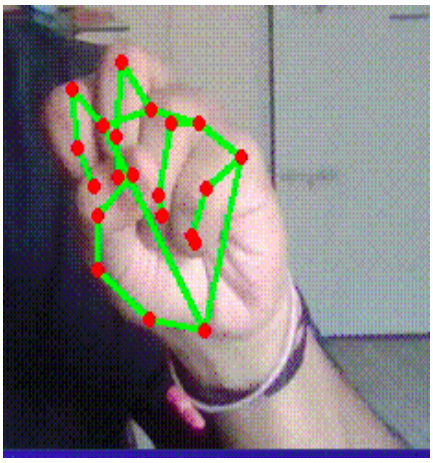
- **Drag and Drop**

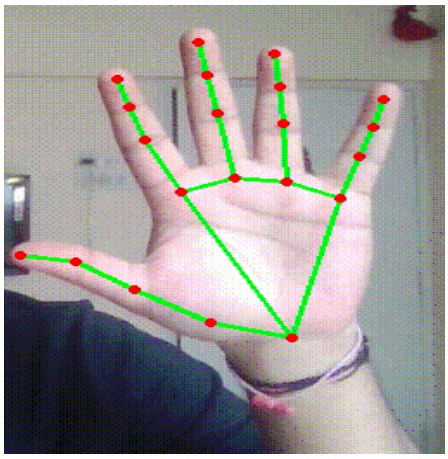




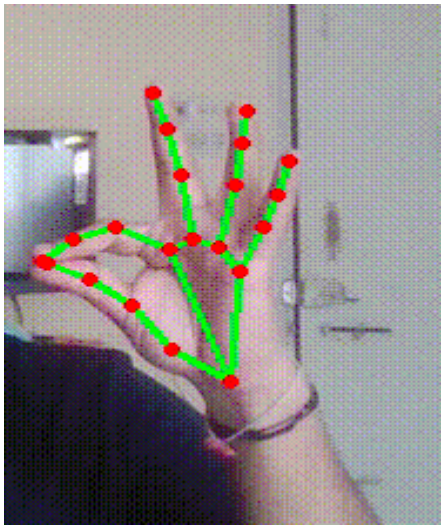
- **Multiple Item Selection**



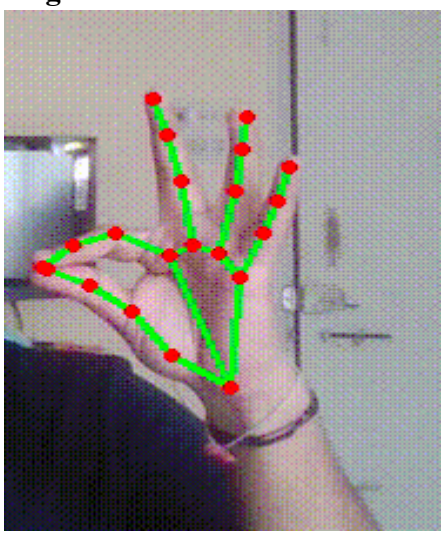




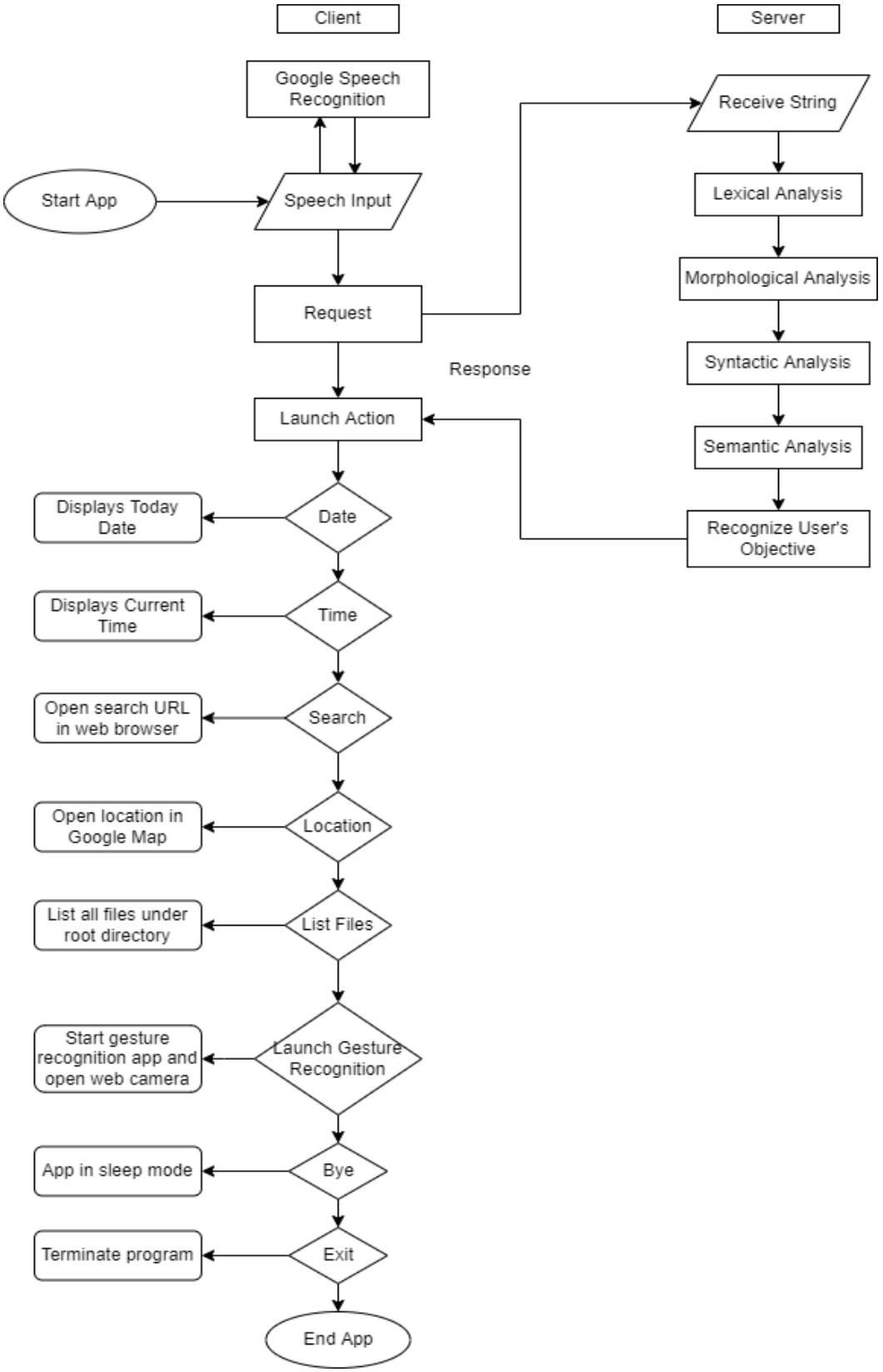
- **Volume Control**



- **Brightness Control**



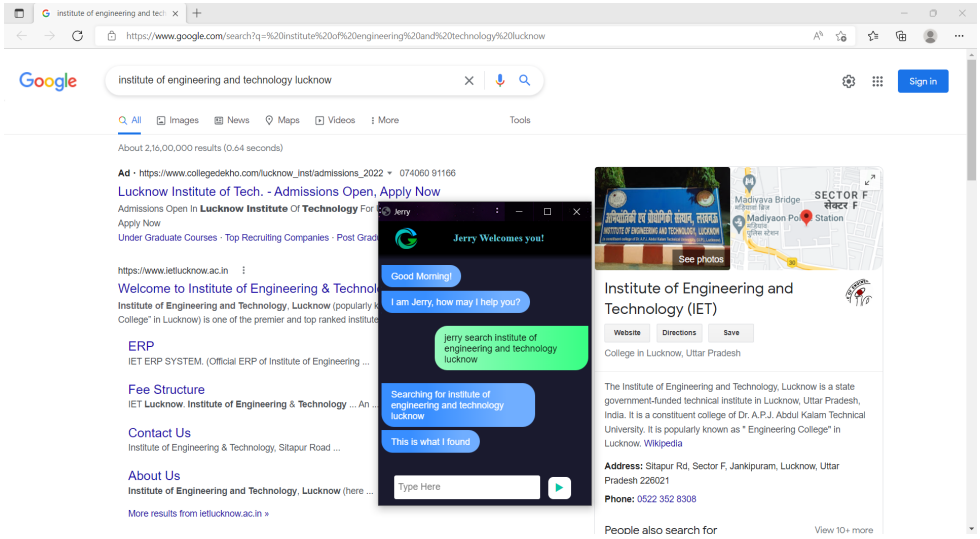
## Voice Control



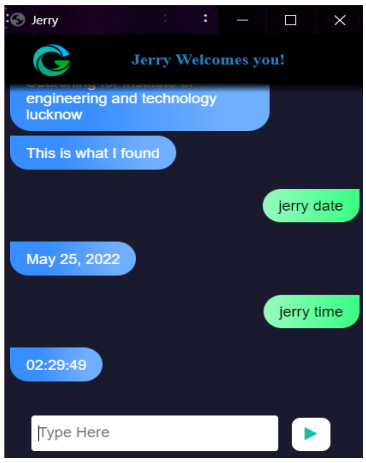
## Voice Assistant Features

- **Launch/Stop Gesture Recognition**

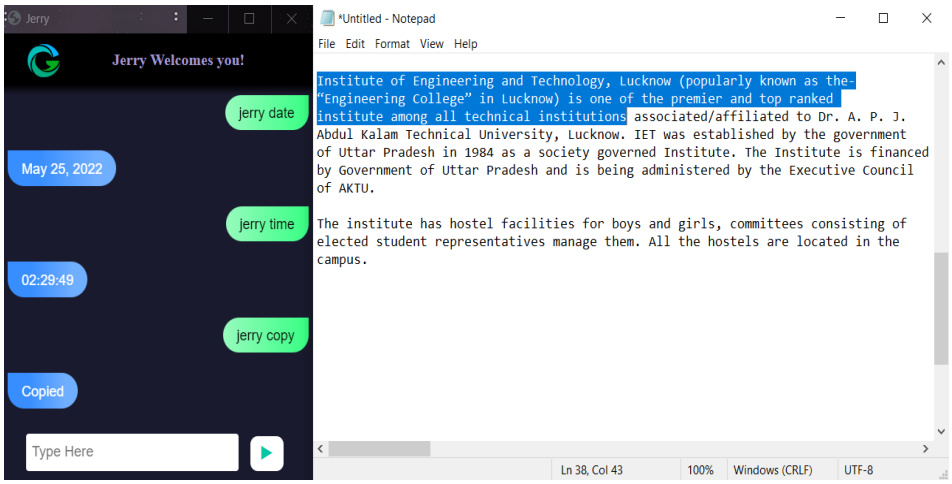

- **Google Search**

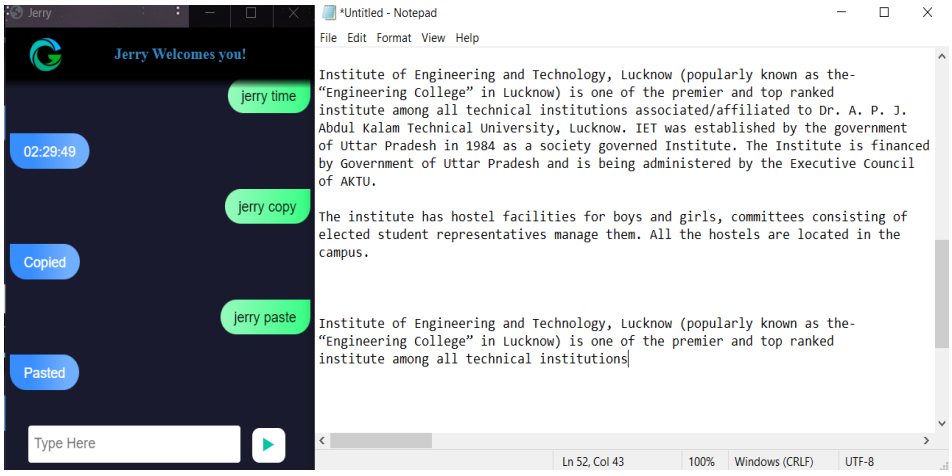


- **Current Date and Time**

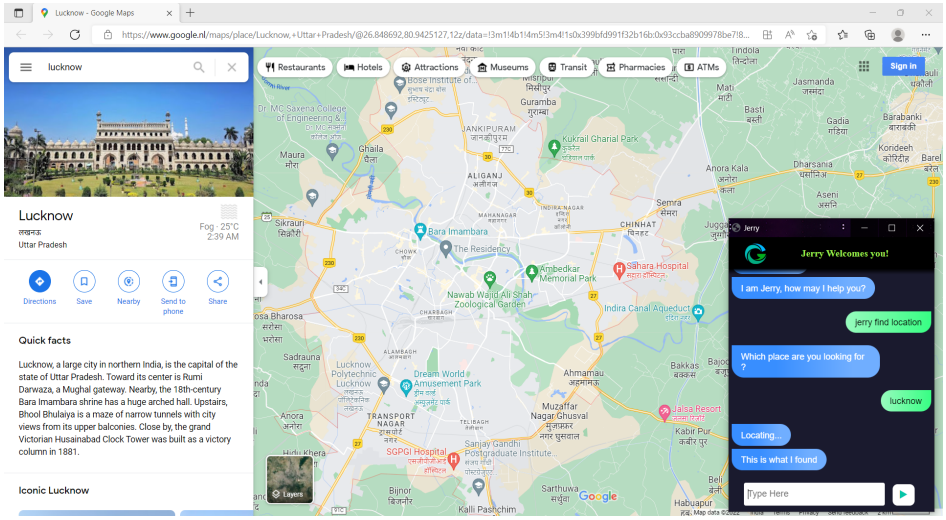



- **Copy and Paste**

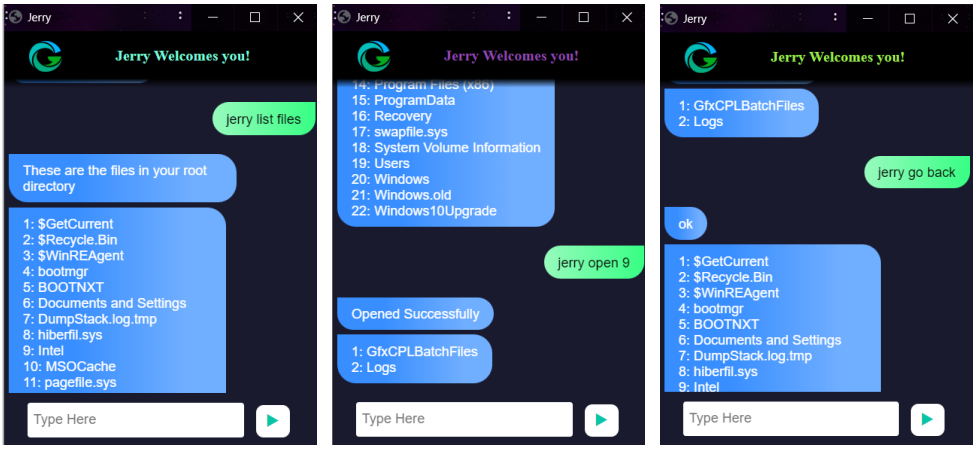


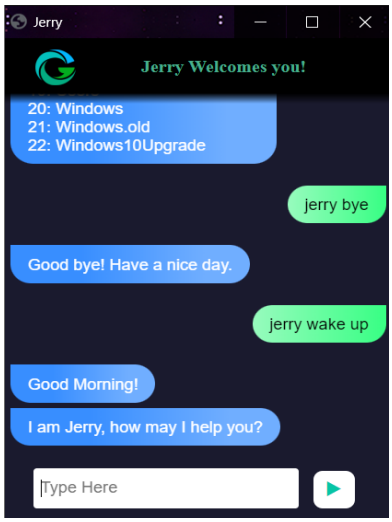
- **Find a location on Google Maps**



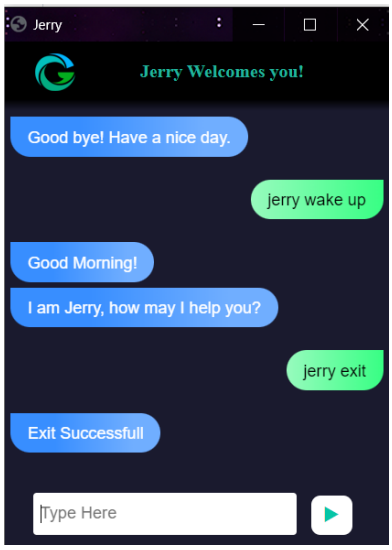

- **File Navigation**

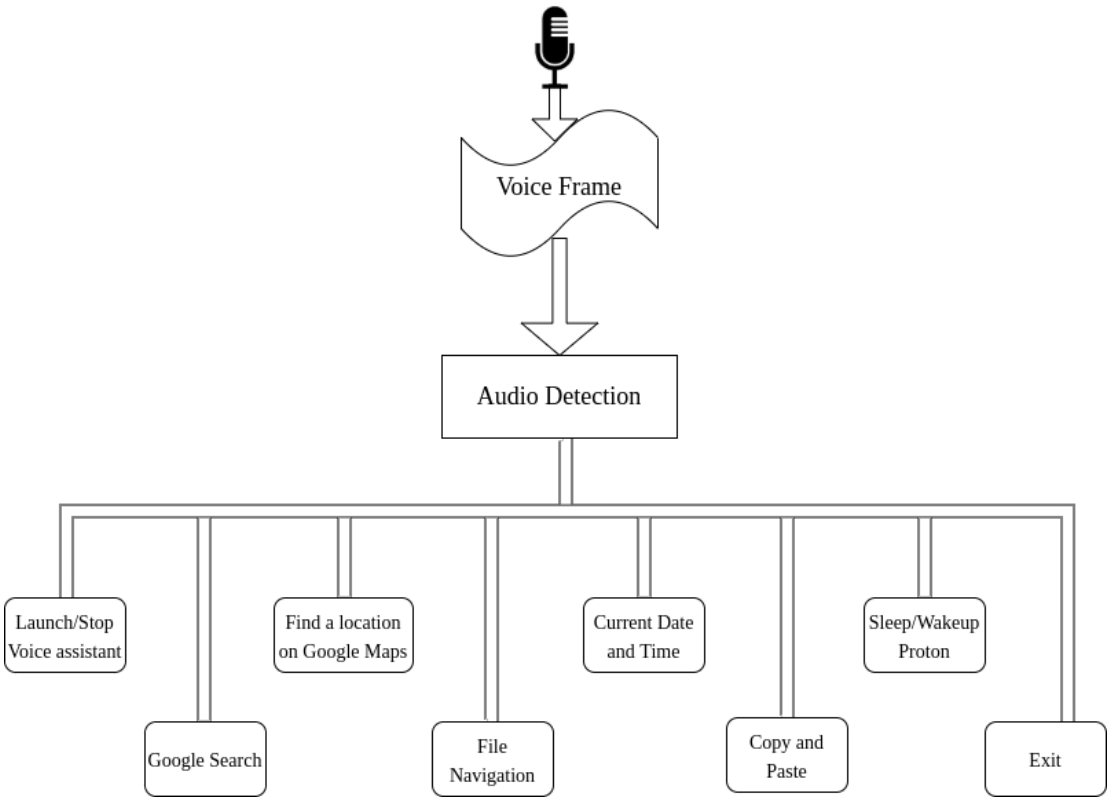


- **Sleep/Wakeup**




- **Exit**




Voice Frame
Audio Detection
Launch/Stop Voice assistant
Google Search
Find a location on Google Maps
File Navigation
Current Date and Time
Copy and Paste
Sleep/Wakeup Proton
Exit

# Hardware and Software Requirement

## Hardware Requirement

The Virtual Mouse application requires the following hardware for development and execution:

- Laptop or Computer Desktop

  To display what the webcam has taken, the virtual software will be started on the laptop or computer desktop.

  The system will make use of (minimum requirements)
  Core2Duo processor (2nd generation)
  2 GB RAM (Main Memory)
  320 GB hard drive
  14-inch LCD monitor

- Webcam

  The image is acquired with a camera that will continue to take photos endlessly so that the application may process the image and calculate pixel position.

  Resolution: 1.3 megapixels is the minimum required.

- Microphone

  Voice commands are recorded via the microphone. Until it is switched off or placed to sleep, it will continue to listen to all commands.

  Microphone should be capable of recognizing frequency range of 40Hz - 16KHz

**Software Requirement**

The following software is required for the development and execution of the Virtual Mouse application:

- Python Language

  The Virtual Mouse application is coded in Python with the help of Microsoft Visual Studio Code, an integrated development environment (IDE) for programming computer applications.
  Basic arithmetic, bit manipulation, indirection, comparisons, logical operations, and more are all available in the Python library.

- Open CV Library

  This software is also created with the help of OpenCV.
  OpenCV (Open Source Computer Vision) is a real-time computer vision library. OpenCV is capable of reading picture pixel values as well as real-time eye tracking and blink detection.

  Software will be using:
  OS : Window 10 Ultimate 64-bit
  Language : Python
  Tool Used : OpenCV and MediaPipe

# Applications

Many applications benefit from the virtual mouse mechanism. It can be utilised to save space by eliminating the need for a physical mouse, as well as in instances where the physical mouse is not available. This technology reduces the need for a hardware device (mouse) and enhances human-computer interaction.

Major applications:

- Can alleviate physical stress on the body, which causes back discomfort, poor vision, and poor posture, among other things.
- Because it is not safe to use equipment by touching them during the COVID-19 outbreak because contacting the gadgets could result in the virus spreading, the proposed virtual mouse can be utilised to manage the computer without using the physical mouse.
- Without the need of gadgets, this system can control automation systems and robots.
- Hand movements can be used to draw 2D images on the virtual system.
- Without the usage of a wireless or cable mouse, a virtual mouse can be utilised to play augmented reality and virtual reality games.
- This virtual mouse can be used by those who have difficulty with their hands to handle computer mouse functionalities.
- Using a combination of gesture and voice control, you can perform the functions of a traditional mouse quickly and efficiently.

# Limitations

There are certain existing environmental issues in this project that may obstruct the outcomes of gesture and voice recognition.

Extreme darkness or brightness can cause the targeted locations on the hand to be overlooked in the captured frames, making the gesture identification process particularly sensitive to light levels. Furthermore, because the current detection region can only handle a radius of 50cm, any display of hand that exceeds this distance will be considered noise and filtered out.

For voice recognition some background noises can hinder the detection of intended command. Identifying different types of accents and performing the exact commands is another tough task.

Furthermore, the program's performance is significantly reliant on the user's hardware, as processing speed and/or resolutions acquired by the webcam/mic may affect the program's load. As a result, the longer it takes to perform a single command, the slower the processing speed and/or the higher the resolutions are.

# Future Scope

Virtual Mouse will be introduced soon to replace the conventional computer mouse, making it easier for users to connect with and administer their computers. In order to correctly track the user's gesture, the software must be fast enough to capture and process every image and speech command.
Other features and improvements could be added to make the application more user-friendly, accurate, and adaptable in different contexts. The following are the enhancements and functionalities that are required:

- **Smart Recognition Algorithm**
  Using the palm and numerous fingers, additional functions such as enlarging and reducing the window, and so on, can be implemented.

- **Better Performance**
  The response time is largely influenced by the machine's hardware, which includes the processor's processing speed, the amount of RAM available, and the webcam's characteristics. As a result, when the software is performed on a respectable machine with a webcam that operates well in various lighting conditions and a better quality microphone that can detect voice instructions correctly and rapidly, the programme may perform better.

# Conclusion

The basic goal of the virtual mouse system is to control the mouse cursor and complete activities without needing a physical mouse by using hand gestures and voice commands. This proposed system is created by using a webcam (or any built-in camera) that recognises hand gestures and hand tip movement and processes these frames to perform the relevant mouse actions using the notion of speech recognition to quickly follow voice commands and perform mouse activities.

The model upon rigorous testing has come out to be highly accurate and sophisticated showing enormous improvements with respect to prior existing models. Since the proposed model has been tested for high sophistication, the virtual mouse can be used for real-time applications. Because the proposed mouse system may be operated digitally utilising hand gestures and voice commands rather than the traditional physical mouse, it will be of more value in combating the propagation of viruses like COVID-19 in the current context.

It functions as a useful user interface and contains all mouse features. Research on advanced mathematical materials for image processing and investigating different hardware solutions has made possible more accurate hand detections. Not only this project shows the different gesture operations and voice commands that can be done by the users but it can also demonstrate the potential in simplifying user interactions with personal computers and hardware systems. Yet a major extension to this work could be to be able to work at a more complex background and compatible with different light conditions.

# References


- Himanshu Bansal, Rijwan Khan, “A review paper on human computer interaction” International Journals of advanced research in Computer Science and Software Engineering, Volume 8, Issue 4, April 2018

- N. Subhash Chandra, T. Venu, P. Srikanth, “A Real-Time Static & Dynamic Hand Gesture Recognition System” International Journal of Engineering Inventions Volume 4, Issue 12, August 2015

- S. Shiriam, B. Nagaraj, J. Jaya, “Deep learning based real time AI Virtual Mouse system using computer vision to avoid COVID-19 spread”, Hindawi Journal of Healthcare Engineering, October 2021.

- Hritik Joshi, Nitin Waybhase, Ratnesh Litoria, “Towards controlling mouse through hand gestures: A novel and efficient approach”, Medi-caps University, May 2022

- Mohhamad Rafi, Khan Sohail, Shaikh Huda, “Control mouse and computer system using voice commands”, International Journal of Research in Engineering and Technology”, Volume 5, Issue 3, March 2016