TABLE OF CONTENT

1. INTRODUCTION

In the new present day progresses at extends the areas of exaggerated reality and contraptions that we will generally will as a rule use in our existence, these devices have gotten diminished at ranges the sort of Bluetooth or far off advancements. This paper proposes Associate in Nursing AI virtual mouse system that produces use of the hand signals and hand tip acknowledgment for performing articulations mouse limits at ranges the advantageous PC cheating adaptable PC vision. The most impartial of the projected system is to perform device pointer works and material performs using a web camera or a characteristic camera at extends the smaller PC rather than using an obsolete mouse contraption. Hand signal

Also, hand tip area by misuse helpful PC vision is used as a HCI with the PC. With the usage of the AI virtual mouse system, we will follow the tip of the hand signal by using an intrinsic camera or net camera and play out the mouse pointer assignments and investigating work and together move the pointer with it.

While utilizing a remote or a Bluetooth mouse, a few gadgets particularly like the mouse, the contraption to interface with the pc, and also, battery to drive the mouse to control a utilized, yet all through this paper, the client utilizes his/her inborn camera or visual camera and utilizations his/her hand signs to deal with the PC mouse works out. Inside the projected construction, data interstate camera hinders this cycle the edges that square measure got accordingly see the different hand signals and hand tip developments shapes the specific mouse work.

Python programming language is utilized for empowering the AI virtual mouse structure, what's more, Open CV that can't avoid being that the library for versatile PC vision is utilized at ranges the AI virtual mouse framework. Inside the projected AI virtual mouse utilizing hand signal, the model purposes the python Media-pipe bunch for the journey for the hands and for pursue of the tip of the hands, what's more, Numpy, Autopy, and PyAuto GUI packs were utilized for propelling the screen of the PC for performing verbalizations limits like left click, right snap, and examining limits. There the projected model show in a perfect world high accuracy level, and in this way the projected model can work respectably in clear application with the use of a cycle or where as not the utilization of PC GPU.

1.1 Introduction to Tools

1.1.1Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.Python can be used on a server to create web applications.Python can be used alongside software to create workflows.Python can connect to database systems. It can also read and modify files.Python can be used to handle big data and perform complex mathematics.Python can be used for rapid prototyping, or for production-ready software development.Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).Python has a simple syntax similar to the English language.Python has syntax that allows developers to write programs with fewer lines than some other programming languages.Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.Python can be treated in a procedural way, an object- oriented way or a functional way.Python can be used for rapid prototyping, or for production-ready software development.Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).Python has a simple syntax similar to the English language.Python has syntax that allows developers to write programs with fewer lines than some other programming languages.Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.Python can be treated in a procedural way, an object- oriented way or a functional way.

1.1.1.1 PYthon 2 V/S Python 3

Python has two major versions, Python 2 and Python 3, with some significant differences between them. Python 3 offers several improvements over Python 2, including better Unicode support, simplified print function, and improved error handling. Python 2 has reached its end-of-life and is no longer receiving updates, so it is recommended to use Python 3 for your project report. However, if you are working with legacy code or libraries that are only compatible with Python 2, you may need to use Python 2.Python 3 is more in-demand and includes a typing system. Python 2 is outdated and uses an older syntax for the print function. While Python 2 is still in use for configuration management in DevOps, Python 3 is the current standard. Python (the code, not the snake) is a popular coding language to learn for beginners.

1.1.1.2 Python 3

Python 3 is an improved version. It supports Unicode characters. If we dive some integer, it gives exact output. Earlier in python 2, we get a rounded value for the same. Most of the IT companies switching towards python3. It gets faster with every newer version. Also, Python has big library support for python3. Companies like Facebook, Instagram are using Python 3 as their language. The most important thing to make a note is Python 2.7 will not be supported after 2020.

1.1.2.Computer Vision

Computer vision is a field of study which encompasses on how computer see and understand digital images and videos. Computer vision involves seeing or sensing a visual stimulus, make sense of what it has seen and also extract complex information that could be used for other machine learning activities. Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information.Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos.

1.1.2.1.Applications of Computer Vision

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.

Major applications:

- Object Detection: OpenCV and MediaPipe can be used to detect objects in images and videos. This can be used in various applications such as surveillance, autonomous driving, and robotics.

- Facial Recognition: Facial recognition is another application of computer vision that can be implemented using OpenCV and MediaPipe. This technology can be used for security purposes, access control, and personalized marketing.

- Gesture Recognition: OpenCV and MediaPipe can be used to recognize human gestures such as hand gestures, facial expressions, and body movements. This technology can be used in applications such as sign language recognition, virtual reality, and gaming.

- Image and Video Analysis: OpenCV and MediaPipe can be used to analyze images and videos for various purposes such as object tracking, motion analysis, and scene understanding. This technology can be used in applications such as sports analytics, medical

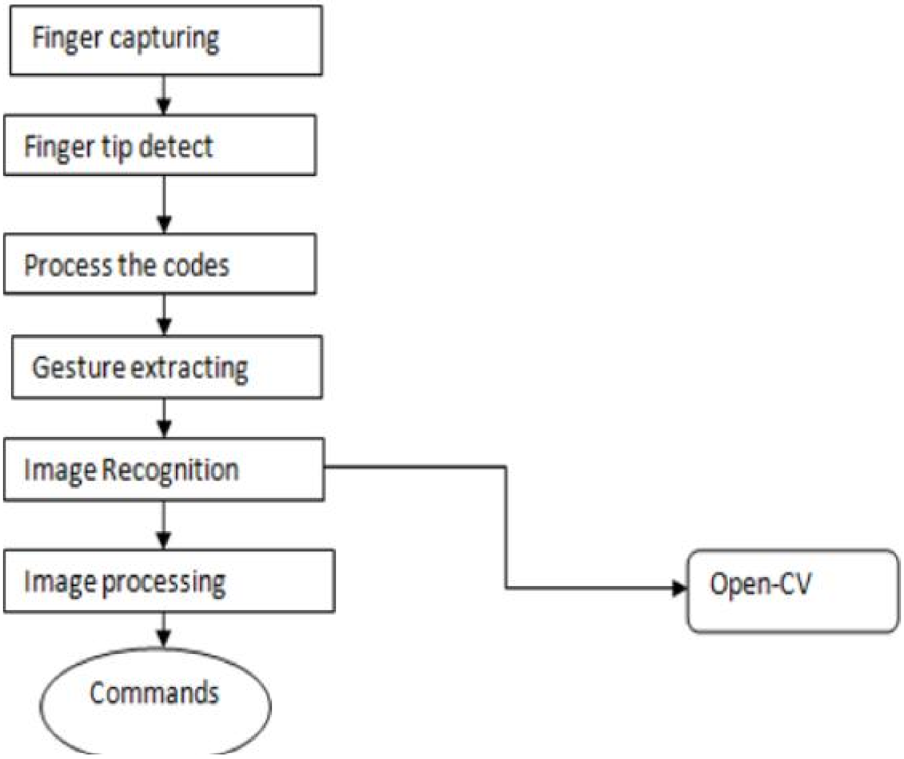
1.1.2.2.Opencv

Opencv is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV. OpenCV was started at Intel in 1999 by **Gary Bradsky**, and the first release came out in 2000. **Vadim Pisarevsky** joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge. Later, its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. OpenCV now supports a multitude of algorithms related to Computer Vision and Machine Learning and is expanding day by day. OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc.,and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

1.1.2.3 openCV-module

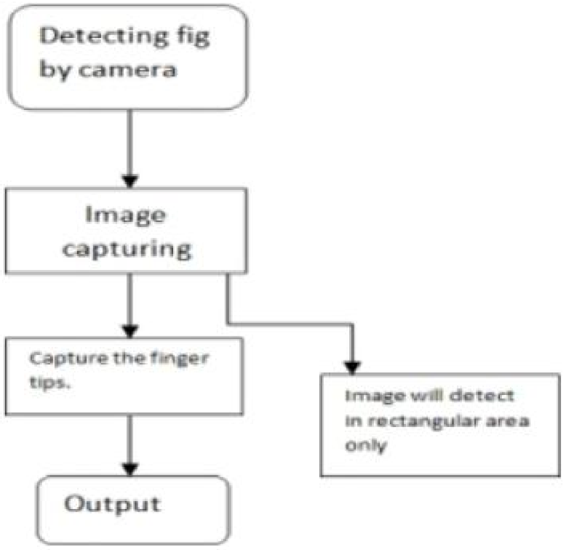
Open CV - Python is a library of Python bindings designed to solve computer vision problems.Python is a general purpose programming language started by **Guido van Rossum** that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.Compared to languages like C/C++, Python is slower.

That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.OpenCV-Python makes use of **Numpy**, which is a highly optimized library for numerical operations with a MATLAB-style syntax.All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as MediaPip.



1.1.2.4.Mediapipe

Media-Pipe is a system that is utilized for applying in a different AI pipeline, partner with an open source structure of Google. The Media-Pipe system is useful for across stage improvement since the edge work is made abuse the measurement data. The Media-Pipe structure is multi-modular, any place this system is frequently applied to differed sounds and recordings .The Media-Pipe structure is utilized by the engineer for building and breaking down the frameworks through diagrams, and it conjointly been utilized for fostering the frameworks for the machine reason.The means worried inside the framework that utilizes Media- Pipe square measure administrated inside the line setup. The The pipeline made will run in various stages allowing quantity friability in portable and work areas. The Media-Pipe structure is predicated on three rudimentary parts, they're execution investigation, system for recovering identifier data, and a gathering of parts that square measure known as mini- computers and those they square measure reusable. A pipeline might be a chart that comprises of parts known as number cruncher any place each mini-computer is associated by streams during which the parcels of information course through.The number cruncher and streams joined produce an information stream outline; the diagram is made with Media- Pipe any place each hub might be an adding machine and thus the hubs square measure associated by stream.



Single-shot is utilized for location and perceiving a finger and palm progressively exploitation journal PC net cam. Finder framework is utilized by the Media Pipe, in the Hand discovery module of python, its style for a finger and hand recognition model because of it's easy to mentor hand. The planed model of hand reason mark comprises of 21 joint reason and co-ordinates inside the hand.

1.1.2.5.Numpy

NumPy is a Python library for numerical computing. It provides support for arrays, which are multi-dimensional data structures for efficient storage and manipulation of large datasets. It includes functions for linear algebra, random number generation, and Fourier transforms, as well as tools for integrating with other libraries like MediaPipe. NumPy is an essential library for many data science, scientific computing, and machine learning tasks, as it allows for vectorized operations and provides high performance due to its use of C backend.

In this project OpenCV represents an image as a NumPy array comprising integers that represent thepixels and intensity- hence, by indexing and slicing portions of the NumPy array, we are essentially isolating specific pixels thereby isolating specific portions of the imageitself, thus allowing us to effectively crop the image

.

1.1.3.Methodology

Pre-processing or to be specific picture handling is an earlier advance in PC vision, where the objective is to change over a picture into a structure reasonable for additional investigation. Instances of tasks, for example, openness rectification, shading adjusting, picture sound decrease, or expanding picture sharpness are exceptionally significant and very consideration requesting to accomplish adequate outcomes.For this article, I propose to introduce a part of the typically used picture taking care of methodology using an outstandingly notable Computer Vision library, Open-CV. I'll endeavor to portray immediately the manner by which each movement works and spotlight more on dealing with the point even more basically, giving you all the code you truly need so you have a functioning experience of the material.

Camera Used In The Virtual Gesture Mouse Project

Open-CV is python vision library that contains Associate in the organized AI virtual mouse system depends upon the edges that are gotten by the camera in Associate in nursing passing PC.

Pictures can be conveyed in concealing layered with 3 channels (Blue, Green, and Red), Grayscale with pixel values fluctuating from 0 (dull) to 255 (white), and twofold portraying dim or white characteristics (0 or 1) specifically.

Moving Hand Through The Window Using Rectangular Area

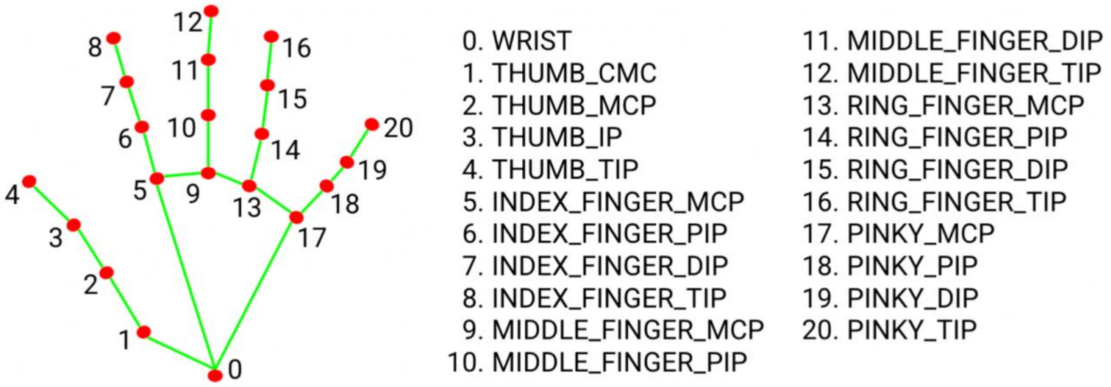
The AI virtual mouse framework utilizes the instructive algorithmic rule, and it changes over the co-ordinates of tip from the camera screen to the pc window full screen for the mouse. whenever the hands unit saw and keeping in mind that we've missing to see that finger is up for topic the specific mouse perform, Associate in Nursing rectangular box is attracted concerning the pc window at ranges the camera locale any spot we've a penchant to will every now and again move all through the window plan the mouse pointer.

Pose Landmarks Detection Task

The MediaPipe Pose Landmarker task lets you detect the landmarks of human bodies in an image. You can use this task to identify key body locations and render visual effects on them. This task uses machine learning (ML) models that can work with single images or a continuous stream of images. The task outputs body pose landmarks in image coordinates and in 3-dimensional (x,y,z) world coordinates.

Hand Landmarks Detection

The MediaPipe Hand Landmarker task lets you detect the landmarks of the hands in an image. You can use this Task to localize key points of the hands and render visual effects over the hands. This task operates on image data with a machine learning (ML) model as static data or a continuous stream and outputs hand landmarks in image coordinates, hand landmarks in world coordinates and handedness(left/right hand) of multiple detected hands.The hand landmark model bundle detects the keypoint localization of 21 hand-knuckle coordinates within the detected hand regions. The model was trained on approximately 30K real-world images, as well as several rendered synthetic hand models imposed over various backgrounds.



The hand landmarker model bundle contains a palm detection model and a hand landmarks detection model. The Palm detection model locates hands within the input image, and the hand landmarks detection model identifies specific hand landmarks on the cropped hand image defined by the palm detection model. Since running the palm detection model is time consuming, when in video or live stream running mode, Hand Landmarker uses the bounding box defined by the hand landmarks model in one frame to localize the region of hands for subsequent frames. Hand Landmarker only re-triggers the palm detection model if the hand landmarks model no longer identifies the presence of hands or fails to track the hands within the frame. This reduces the number of times Hand Landmarker tiggers the palm detection model.

Gesture Recognition

The MediaPipe Gesture Recognizer task lets you recognize hand gestures in real time, and provides the recognized hand gesture results along with the landmarks of the detected hands. You can use this task to recognize specific hand gestures from a user, and invoke application features that correspond to those gestures. This task operates on image data with a machine learning (ML) model, and accepts either static data or a continuous stream. The task outputs hand landmarks in image coordinates, hand landmarks in world coordinates, handedness (left/right hand), and the hand gesture categories of multiple hands.

Detect The Finger Tips & Doing The Mouse Cursor Movements

In this framework, AI mouse is police evaluation that finger is up misleading the spot co-ordinate of the particular finger that it'll found abuse the Media-Pipe and hence the singular bits of the fingers that region unit up, and according to that, the authentic mouse perform is played out its assignments.

2.SYSTEM STUDY

2.1 Introduction

Computer technology has become an integral part of our lives, and the use of a computer mouse has become ubiquitous. However, there are certain situations where the use of a physical mouse is not feasible or practical. For example, individuals with disabilities may find it difficult to use a traditional mouse. This is where the virtual mouse system comes into play, which enables users to control the mouse pointer using hand gestures and movements.This can be customized according to the users,the virtual mouse system is an innovative solution that provides a new way of controlling a computer without a physical mouse. It uses computer vision techniques to track hand gestures and movements, which are then translated into mouse pointer movements. In this system study, we will examine how we can implement the virtual mouse system using OpenCV and Mediapipe. OpenCV is a popular computer vision library that provides a range of functions for image and video processing. It has been used for a variety of applications, including object detection, facial recognition, and augmented reality. Mediapipe is a newer computer vision library developed by Google that focuses on real-time, cross-platform computer vision applications. It provides efficient hand, face, and pose detection in real-time, making it an ideal choice for the virtual mouse system.The virtual mouse system implemented using OpenCV and Mediapipe will enable users to control the mouse pointer using hand gestures and movements. By detecting hand movements in real-time, the system will provide an intuitive and innovative way of controlling the computer, which can be especially useful for individuals with disabilities. The system study outlined in this report provides an overview of how OpenCV and Mediapipe can be used to implement the virtual mouse system.

The virtual mouse system is an innovative solution that provides a new way of controlling a computer without a physical mouse. It uses computer vision techniques to track hand gestures and movements, which are then translated into mouse pointer movements. OpenCV and Mediapipe are two popular computer vision libraries that can be used to implement the virtual mouse system. OpenCV provides functions for image and video processing, while Mediapipe focuses on real-time, cross-platform computer vision applications. The combination of these libraries can provide an efficient and intuitive way of controlling a computer using hand gestures and movements.

2.2.Economic Feasibility

The economic feasibility of a virtual mouse system using OpenCV and Mediapipe will depend on the cost of development and implementation, as well as the potential benefits and savings that the system can provide. Some of the economic factors to consider include the cost of software development, hardware requirements, training and support, and ongoing maintenance and updates.However, the benefits of a virtual mouse system can be significant. For individuals with disabilities or physical impairments, the virtual mouse system can provide a much-needed alternative to a traditional mouse, improving their accessibility and productivity. Additionally, the system can be useful in situations where a physical mouse is not practical, such as in a hospital or laboratory setting, where the use of a traditional mouse could pose a risk of contamination.

The cost-benefit analysis of a virtual mouse system should take into account the potential savings and benefits that the system can provide. For example, the system could reduce the need for expensive specialized input devices, or improve productivity by enabling users to control the computer more efficiently. Additionally, the system could have a positive impact on the health and wellbeing of users with physical impairments, reducing the risk of repetitive strain injuries or other health issues.Overall, the economic feasibility of a virtual mouse system using OpenCV and Mediapipe will depend on the specific use case and the potential benefits and savings that the system can provide. By carefully considering the costs and benefits, it may be possible to develop and implement a system that provides significant value to users while remaining cost-effective and economically feasible.

2.3.Operational Feasibility

Moving Hand Through The Window Using Rectangular Area

The AI virtual mouse framework utilizes the instructive algorithmic rule, and it changes over the co-ordinates of tip from the camera screen to the pc window full screen for the mouse. whenever the hands unit saw and keeping in mind that we've missing to see that finger is up for topic the specific mouse perform, Associate in Nursing rectangular box is attracted concerning the pc window at ranges the camera locale any spot we've a penchant to will every now and again move all through the window plan the mouse pointer.

Detect The Finger Tips & Doing The Mouse Cursor Movements

In this framework, AI mouse is police evaluation that finger is up misleading the spot co-ordinate of the particular finger that it'll found abuse the Media-Pipe and hence the singular bits of the fingers that region unit up, and according to that, the authentic mouse perform is played out its assignments.

2.4.Existing System

The existing system of a computer mouse typically includes a pointing device that is used to control the cursor on a computer screen. The device typically has one or more buttons that can be clicked to initiate actions on the computer, such as opening a file or selecting text. The mouse usually connects to the computer via a wired or wireless connection, and the movement of the device is tracked by a sensor, either a mechanical ball or an optical sensor.In recent years, various improvements have been made to computer mice, including the addition of scroll wheels, touchpads, and trackballs. Some mice now feature programmable buttons and adjustable DPI settings, allowing for greater customization and precision. Additionally, ergonomic designs have been developed to help reduce discomfort and strain during prolonged use.

2.5.Proposed System

the proposed system of virtual mouse technology using OpenCV and Mediapipe include improved accessibility for people with physical disabilities, greater convenience for users who prefer a touchless interface, and increased flexibility and customization options for developers.
this potential to increase productivity and efficiency for users, as it allows for faster and more precise control of the mouse cursor. It also has applications in areas such as gaming, virtual reality, and augmented reality, where touchless interfaces are becoming increasingly important.

3.SYSTEM ANALYSIS

3.1 Introduction

Artificial intelligence (AI) has revolutionized the field of computer science, enabling machines to perform tasks that once required human intervention. One such task is the use of a physical mouse to interact with a computer. In recent years, the development of virtual mice using AI and computer vision technology has gained popularity due to its potential to overcome the limitations of physical mice. In this project, we have designed an AI virtual mouse using the Python programming language and the OpenCV library for computer vision. The virtual mouse operates through hand gestures, allowing users to interact with their computer in a natural and intuitive way, without the need for a physical device. We have implemented this project on Anaconda distribution, a popular platform for scientific computing and data science tasks. By using a webcam or built-in camera for hand gesture capture and hand tip detection using computer vision, the proposed AI virtual mouse system overcomes the limitations of traditional physical mice that rely on batteries or dongles to connect to the computer. The machine learning algorithm used in the system detects hand gestures, allowing users to perform various functions such as left click, right click, scrolling, and cursor movement. The algorithm is based on deep learning, ensuring accurate and reliable results.

3.2 Hardware Specification

PROCESSOR	: INTEL I5 / AMD RYZEN 5
HARD DISK	: 50GB
CACHE MEMORY	: 8 MB
RAM	: 8 GB
MONITOR	: L C D MONITOR
CAMERA	: 5 MEGA PIXEL

3.3 Software Specification

OPERATING SYSTEM	: WINDOWS 10
PROGRAMMING LANGUAGE	: PYTHON
COMPUTER VISION	: OPENCV,MEDIAPIPE
TERMINAL	: ANACONDA PROMPT

4.SYSTEM DESIGN

4.Introduction

This paper proposes 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. The main objective of the proposed system is to perform computer mouse cursor functions and scroll function using a web camera or a built-in camera in the computer instead of using a traditional mouse device. Hand gesture and hand tip detection by using computer vision is used as a HCI with the computer. With the use of the AI virtual mouse system, we can track the fingertip of the hand gesture by using a built-in camera or web camera and perform the mouse cursor operations and scrolling function and also move the cursor with it. While using a wireless or a Bluetooth mouse, some devices such as the mouse, the dongle to connect to the PC, and also, a battery to power the mouse to operate are used, but in this paper, the user uses his/her built-in camera or a webcam and uses his/her hand gestures to control the computer mouse operations. In the proposed system, the web camera captures and then processes the frames that have been captured and then recognizes the various hand gestures and hand tip gestures and then performs the particular mouse function.

Python programming language is used for developing the AI virtual mouse system, and also, OpenCV which is the library for computer vision is used in the AI virtual mouse system. In the proposed AI virtual mouse system, the model makes use of the MediaPipe package for the tracking of the hands and for tracking of the tip of the hands, and also, Pynput, Autopy, and PyAutoGUI packages were used for moving around the window screen of the computer for performing functions such as left click, right click, and Hindawi Journal of Healthcare Engineering Volume 2021, Article ID 8133076, 8 pages https://doi.org/10.1155/2021/8133076 scrolling functions. the results of the proposed model showed very high accuracy level, and the proposed model can work very well in real-world application with the use of a CPU without the use of a GPU

4.2.Integrated Camera System for AI Virtual Mouse Control

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, the web camera captures and passes the frames to the AI virtual system. the AI virtual mouse system uses the webcam where each frame is captured till the termination of the program. $e video frames are processed from BGR to RGB color space to find the hands in the video frame by frame as shown in the following code:

```
def findHands(self, img, draw = True):
imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
self.results = self.hands.process(imgRGB)
```

4.3.Computer Vision-based Hand Gesture Mouse Control System

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 6, and according to that, the particular mouse function is performed.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.

5.SYSTEM TESTING

5.1 Types Of Testing

Testing is a critical aspect of software development, and it helps ensure that the software is working as intended and meets the needs of its users. In the case of virtual mouse technology using OpenCV and Mediapipe, testing is particularly important to ensure that the software is accurate, responsive, and accessible to users with different levels of physical ability. By conducting a variety of testing types, developers can identify and fix issues before the software is released, which can help improve the user experience and reduce the risk of errors or security vulnerabilities.The various levels at which testing are conducted are:

- Unit testing
- Integration testing
- Sequential testing
- System testing
- Validation testing

Unit Testing

In unit testing each program unit is tested individually. so any errors in a unit are debugged. Sample data is given for unit testing. The unit test results are recorded for future references. Unit testing focus verification efforts on the smallest unit of software design, the module. This is known as “module testing”. It comprises of the set test performed by an individual programmer prior to the integration of unit into the large system. The modules are tested separately, this testing is carried out programming stage itself.

In this step each module is found to be working satisfactory as regard to the expected out from module. The unit testing was done for every module in the software for various inputs, such they each line of code is at least once executed. This testing was carried out during the unit to a large system.

Integration Testing

Integration testing is a software testing technique that focuses on verifying the interactions and interfaces between different components of a software system. The goal of integration testing is to identify and resolve issues that may arise when the components of a system are combined and interact with each other.During the integration testing process, individual software components are combined into larger groups and tested as a whole. This allows developers to identify and resolve compatibility issues, data flow problems, and other types of errors that may occur when the components of a system interact.

Program Testing

Program testing checks for two types of errors; syntax and logic. A syntax error is a program statement that violates one or more rules of the language in which it is written. A logic error deals with incorrect data fields. When a program is tested, the actual output is compared with the expected output. All the modules are combined and tested as a whole. Here correction is difficult because the vastexpenses of all errors uncovered are correct for the next testing steps. We follow bottom-up integration. Bottom up integration testing as its name implies begin construction andsling with atomic modules. Because components are integrated from the bottom up, accessing required for the components subordinate to a given level is always available and need for stubs iseliminated.

Sequential Testing

Sequential or series testing is checking the logic of one or more programs in the candidate system, where the output of one program will affect the processing done by another program.

System Testing

System testing executing a program to check logic changes made in it and with the intension of finding errors-making the program fails. Effective testing does not guaranties reliability is a design consideration. This testing actually consistsof a series of different test whose primary purpose is to fully exercise the computer based system. It begins where integration testing is completed and finally software is completely assembled as package ,interfacing errors are uncovered and corrected.

Acceptance Testing

Acceptance testing is running the system with live data by the actual user. An acceptance test has the objective of selling the user in the validity and reliabilityof the system. A comprehensive test report is prepared. The report indicates the system's tolerance, performance range, error rate and accuracy. It verifies the system procedures operate to system specification and the integrity of important data is maintained, performance of an acceptance test is actually the users show. User motivation is very important for the successful performanceof the system. After that a comprehensive test report is prepared. This report shows the systems tolerance, performance range, error rate and accuracy.

performance testing

Performance testing is a type of software testing that involves testing the software's performance under different conditions to ensure that it meets the desired performance criteria. In the case of virtual mouse development using OpenCV and Mediapipe, performance testing would involve testing the software under different lighting conditions and with different hand movements to ensure that it is accurate and responsive. This could involve conducting stress tests, load tests, or other types of tests to simulate different user scenarios and verify that the software is performing as expected. By conducting performance testing, developers can identify and fix issues related to the software's speed, responsiveness, and accuracy, which can help improve the user experience and ensure that the software meets the needs of its users.

Input Testing

input testing would involve testing the system's ability to recognize different types of inputs and respond appropriately. This would include testing the system with various input types, such as mouse movements, clicks, and gestures, to ensure that the system can accurately detect and respond to each input.To test the system's ability to recognize inputs accurately, testers would need to create a test plan that includes different scenarios that simulate real-world usage. For example, testers could create test cases that involve moving the virtual mouse in different directions and at different speeds, clicking on different parts of the screen, and using gestures such as swipes and pinches.In addition to testing the system's ability to recognize and respond to inputs accurately, testers would also need to test the system's performance under different conditions.

Output Testing

Output testing would involve verifying the accuracy and correctness of the system's responses to different inputs. This would include testing the system's ability to accurately move the virtual mouse on the screen, click on the right targets, and recognize different gestures. To conduct output testing for a virtual mouse using computer vision, testers would need to create test cases that simulate various scenarios and input types. For example, testers could create test cases that involve moving the virtual mouse to different parts of the screen and clicking on specific targets, such as buttons or links. Testers could also test the system's ability to recognize different types of gestures, such as swipes or pinches, and respond appropriately. Testers would need to verify that the system's responses to different inputs are accurate and meet the specified requirements. This would include checking that the virtual mouse moves to the correct target, clicks on the right button, and performs the expected actions. In addition to verifying the accuracy of the system's responses, testers would also need to test the system's performance under different conditions. This could include testing the system's response time, accuracy, and efficiency when there are multiple virtual mice on the screen, or when the screen resolution changes.

Security Testing

Security testing is a type of software testing that involves testing the software for vulnerabilities and ensuring that it is secure against potential attacks. In the case of virtual mouse development using OpenCV and Mediapipe, security testing would involve testing the software for potential security issues such as unauthorized access to the user's computer or malicious code injection. This could involve conducting penetration tests or vulnerability scans to identify any potential vulnerabilities and verify that the software is secure. By conducting security testing, developers can identify and fix any security issues before the software is released to the public, which can help ensure that the software is secure and protects the user's sensitive information.

5.2.System Implementation

A crucial phase in the system life cycle is the successful implementation of the new system design. Implementation involves creating computer compatible files, training the operating staff, installing hardware, terminals. In the system implementation, user training is crucial for minimizing resistance to change and giving the new system a chance to prove its worth. The objectives of the system implementation is to put the system into operation while holding costs, risks and personal irritation to minimum. Once the physical system has been designed in details, the next stage is to run the design into a working systemand then to monitor the operation of the system to ensure that is continue to work efficiently and the operation of the system to ensure that is continues to work efficiently and effectively. The implementation stage of a is often very complex and time consuming because many more people are involved than in the earlier stages.

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. To achieve the objectives and benefits from computer based system, it is essential for the people who will be confident of their role in the new jobs. After software is developed to meet user’s requirements, users test it for acceptance. The change over phase is used to provide adaptability for the new system.

5.3.System Maintenance

Software maintenance is the process of modifying a software system or component after its delivery in order to correct faults,improve the performance and other attributes ,or to adapt to the changed environment .maintenance covers a wide range of activities including correcting the error and design coding, updating the documentation and test data , and upgrading the user hardware and software .maintenance is always necessary to keep the software usable and useful. Hardware also requires periodic maintenance to keep the system into its standards. After installation is completed and user start is adjust to the changes created by the candidate system.evaluation and maintenance begin. If new information is consistent with design specification the changes have to be made. Hardware also requires periodic maintenance to keep in tune with design specifications. User priorities changes in organizational requirements or environmental factors also called for system enhancements. Maintenance covers wide range of activities, including correcting, coding and design errors, updating documentation and test data ,and upgrading user support.any activities classified as maintenance are actually enhancements.

Maintenance means restoring something to do its original condition. Unlike hardware, software does not wear outfit is corrected. In contrast, enhancement means adding, modifying or redeveloping the code to support changes in the specifications. It is necessary to keep up with changing user needs the operational environment. Maintenance means repairing processing or performance failures or making changes because of previously uncorrected problems or false assumptions .adaptive maintenance means changing the program function. Perfective maintenance means enhancing the performance or modifying the program to respond to the user's additional or changing needs of these types more time and money are spend on perfective than on corrective and adaptive maintenance together. Maintenance activities begin where conversion leaves off. Maintenance is handled by the same planning and control used in a formal system project .a major problem with Software maintenance is its labor intensive nature. documentation is as much a part ofmaintenance as it is of system development .to put maintenance in its proper perspective requires considerable skill and experience and is an important and is an important and ongoing aspect of system development is an additional factor in the success of the maintenance programmer is the work environment.

5.3.1.Software Maintenance Activities can be Classified into:

- Corrective maintenance.
- Adaptive maintenance.
- Perceptive maintenance.

Corrective maintenance removes software faults. Perfective maintenance improves the system without changing its functionality.the objective of perfective maintenance should be to prevent failures and optimize the software.Adaptive maintenance modifies the software to keep it up to date with its operative environment.it may be needed because of changes in the user requirements ,changes in target platform, or changes in external interfaces. Minor adaptive changes should be handled by normal maintenance process major adaptive changes should be carried out as, a separate development project.If you want to changes the software to improve future maintainability or reliability or to provide better basis for future enhancement then perceptive maintenance is perform.

6. CONCLUSION

Conclusion

In conclusion, the development of an AI virtual mouse using Python and OpenCV has shown promising potential for revolutionizing the way users interact with computers. By using computer vision and deep learning algorithms, the system can accurately detect hand gestures, allowing users to perform various functions without the need for a physical device. This technology not only offers greater accessibility for people with physical disabilities but also provides a touchless interface, which is particularly relevant in the current COVID-19 pandemic. Furthermore, the AI virtual mouse system has various potential applications beyond traditional computer usage, such as in the fields of robotics, automation, and virtual reality. Overall, this project demonstrates the power of AI and computer vision technology in overcoming the limitations of physical devices and enhancing user experience.

In addition to the potential applications, the AI virtual mouse system also offers several advantages over traditional physical mice. For example, it eliminates the need for batteries or dongles, which can be costly and environmentally unfriendly. It also reduces clutter and improves mobility, as the user can control their computer without being tethered to a physical device. Additionally, the system is highly customizable and can be adjusted to suit the user's preferences, such as sensitivity, hand gesture recognition, and button mapping.The implementation of the AI virtual mouse system on Anaconda distribution provides a powerful platform for scientific computing and data science tasks, making it easier for developers to integrate the system into their existing workflows. Furthermore, the deep learning algorithm used in the system allows for ongoing improvement and optimization, ensuring that the system remains accurate and reliable over time.Overall, the development of an AI virtual mouse using Python and OpenCV represents a significant step forward in computer-human interaction, offering a more intuitive, accessible, and customizable user experience. With continued development and improvement, this technology has the potential to transform the way we interact with computers and other devices in the future.

7.FUTURE WORK

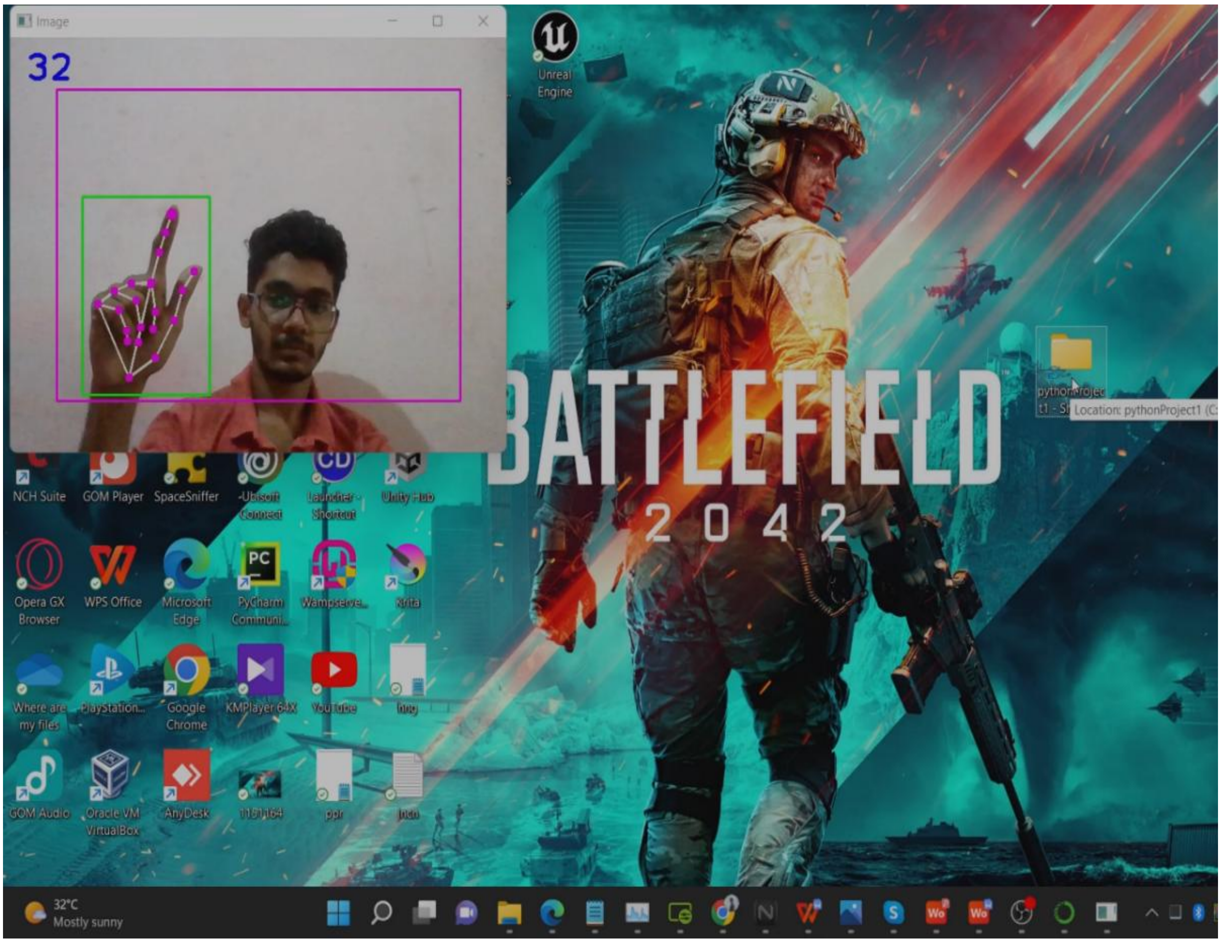
Future Work

The work is to make PC understanding projects that can deal with continuous issues and to achieve targets of the affiliations and regular daily existence as well as individuals. There is a degree in encouraging the machines games, talk affirmation machine, language revelation, PC vision, ace systems, progressed mechanics, etc The more you learn about AI sciences, for instance material science or science, the better. For the normal ways of managing Artificial Intelligence, find out about mind science and the tangible framework. Get to know some Machine vernaculars. It is savvy to focus on one crucial machine language. Occupations are commonly to depend after getting the programming vernaculars. Calling decisions in AI where student can land positions at Occupation will be offered like: Game Programmer, Robotic, Scientist, Computer Scientist and data analyst. Man-made consciousness has different applications in the present society. It is becoming fundamental for the present time since it can take care of complicated issues with an effective way in various ventures, like Health care, diversion, finance, schooling, and so forth Computer based intelligence is making our everyday existence more agreeable and quick.

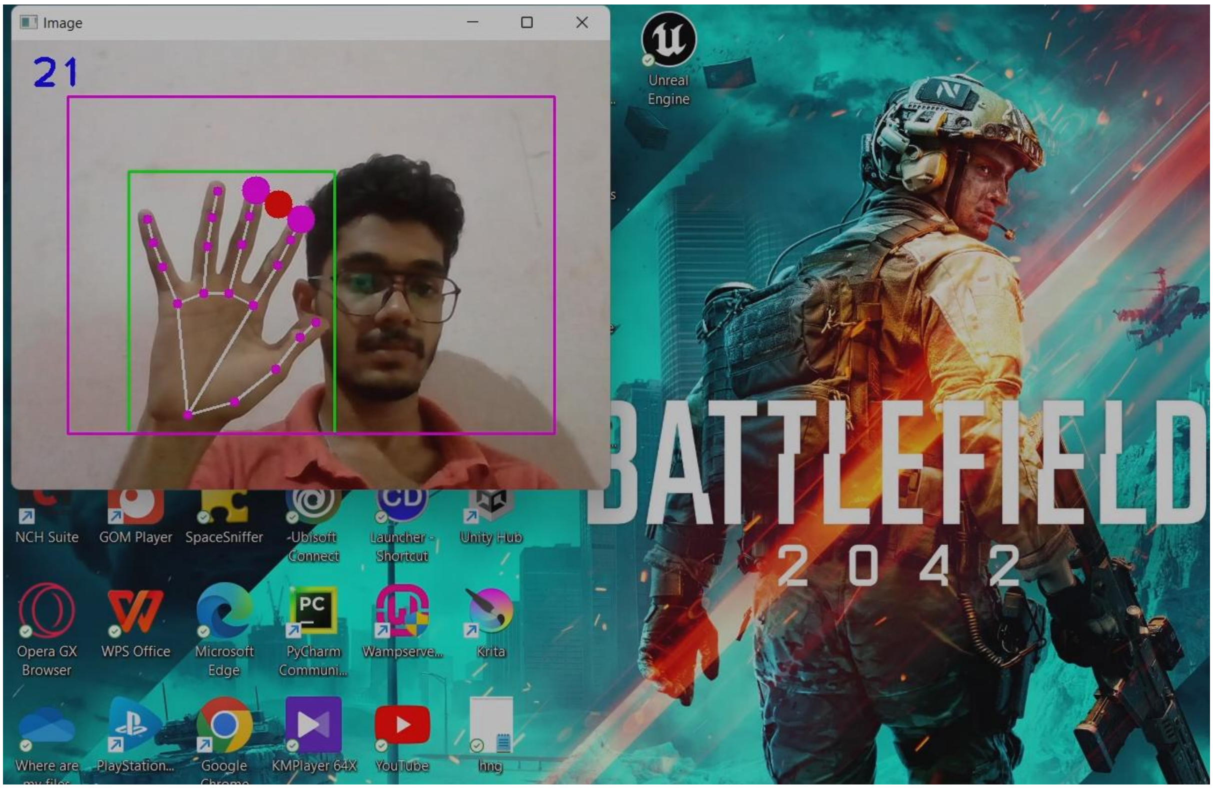
8.APPENDIX

8.1 Screenshot

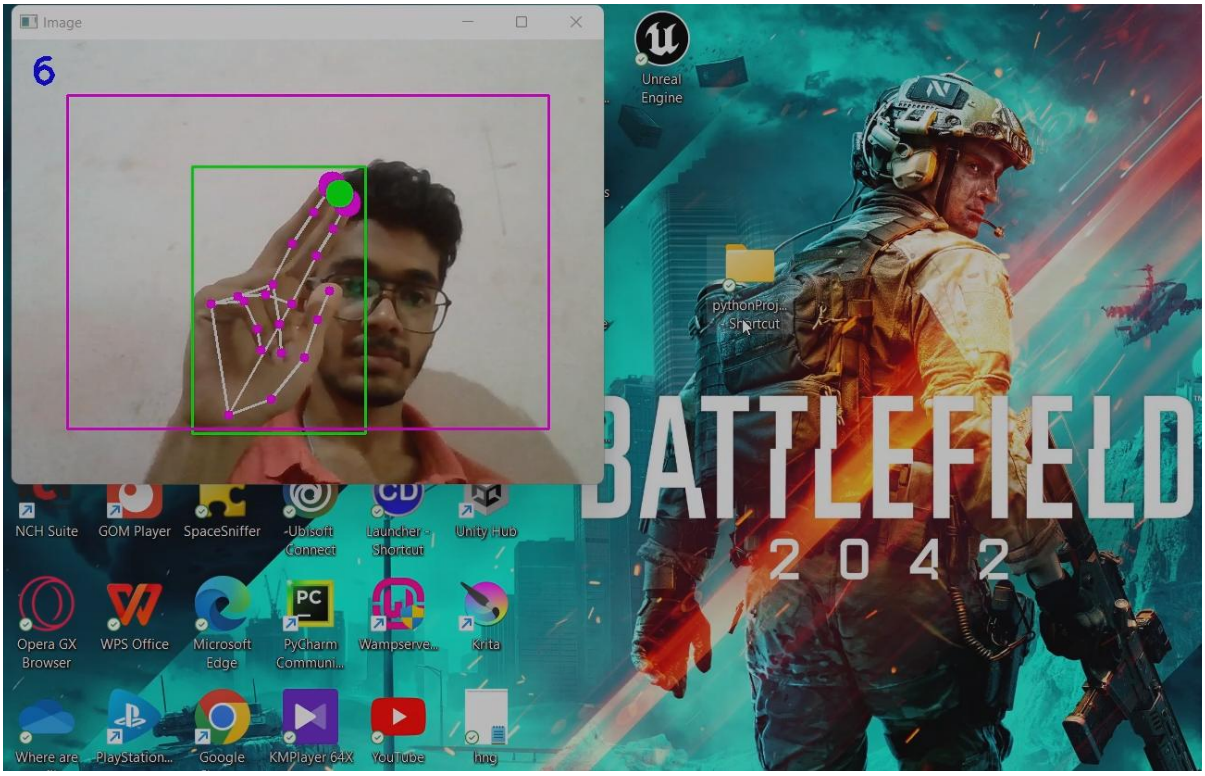
Live Captured Images:



Hand Landmark Detection:



Gester Recognition:



8.2.SAMPLE CODE:

HandTracking.py

```
import cv2
import mediapipe as mp
import time
import math
import numpy as np



class handDetector():
    def __init__(self, mode=False, maxHands=2, detectionCon=False, trackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands,
                                        self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils
        self.tipIds = [4, 8, 12, 16, 20]

    def findHands(self, img, draw=True):    # Finds all hands in a frame
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.hands.process(imgRGB)

        if self.results.multi_hand_landmarks:
            for handLms in self.results.multi_hand_landmarks:
                if draw:
                    self.mpDraw.draw_landmarks(img, handLms,
                                               self.mpHands.HAND_CONNECTIONS)

        return img

    def findPosition(self, img, handNo=0, draw=True):   # Fetches the position of hands
        xList = []
        yList = []
        bbox = []
        self.lmList = []
        if self.results.multi_hand_landmarks:
            myHand = self.results.multi_hand_landmarks[handNo]
            for id, lm in enumerate(myHand.landmark):
                h, w, c = img.shape
                cx, cy = int(lm.x * w), int(lm.y * h)
                xList.append(cx)
                yList.append(cy)
                self.lmList.append([id, cx, cy])
                if draw:
                    cv2.circle(img, (cx, cy), 5, (255, 0, 255), cv2.FILLED)

            xmin, xmax = min(xList), max(xList)
            ymin, ymax = min(yList), max(yList)
            bbox = xmin, ymin, xmax, ymax

            if draw:
                cv2.rectangle(img, (xmin - 20, ymin - 20), (xmax + 20, ymax + 20),
                              (0, 255, 0), 2)

        return self.lmList, bbox

    def fingersUp(self):    # Checks which fingers are up
        fingers = []
        # Thumb
        if self.lmList[self.tipIds[0]][1] > self.lmList[self.tipIds[0] - 1][1]:
            fingers.append(1)
        else:
            fingers.append(0)

        # Fingers
        for id in range(1, 5):

            if self.lmList[self.tipIds[id]][2] < self.lmList[self.tipIds[id] - 2][2]:
                fingers.append(1)
            else:
                fingers.append(0)

        # totalFingers = fingers.count(1)

        return fingers

    def findDistance(self, p1, p2, img, draw=True,r=15, t=3):   # Finds distance between
two fingers
        x1, y1 = self.lmList[p1][1:]
        x2, y2 = self.lmList[p2][1:]
        cx, cy = (x1 + x2) // 2, (y1 + y2) // 2

        if draw:
            cv2.line(img, (x1, y1), (x2, y2), (255, 0, 255), t)
            cv2.circle(img, (x1, y1), r, (255, 0, 255), cv2.FILLED)
            cv2.circle(img, (x2, y2), r, (255, 0, 255), cv2.FILLED)
            cv2.circle(img, (cx, cy), r, (0, 0, 255), cv2.FILLED)
        length = math.hypot(x2 - x1, y2 - y1)

        return length, img, [x1, y1, x2, y2, cx, cy]



def main():
    pTime = 0
    cTime = 0
    cap = cv2.VideoCapture(1)
    detector = handDetector()
    while True:
        success, img = cap.read()
        img = detector.findHands(img)
        lmList, bbox = detector.findPosition(img)
        if len(lmList) != 0:
            print(lmList[4])
        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime

        cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 255), 3)

        cv2.imshow("Image", img)
        cv2.waitKey(1)


if __name__ == "__main__":
    main()
```

Main.ipynb

```
import cv2
import numpy as np
import time
import HandTracking as ht
import autopy   # Install using "pip install autopy"

### Variables Declaration
pTime = 0               # Used to calculate frame rate
width = 640             # Width of Camera
height = 480            # Height of
frameR = 60             # Frame Rate
smoothening = 8         # Smoothening Factor
prev_x, prev_y = 0, 0   # Previous coordinates Camera
curr_x, curr_y = 0, 0   # Current coordinates

cap = cv2.VideoCapture(0)   # Getting video feed from the webcam
cap.set(3, width)           # Adjusting size
cap.set(4, height)
detector = ht.handDetector(maxHands=1)                  # Detecting one hand at max
screen_width, screen_height = autopy.screen.size()      # Getting the screen size
while True:
    success, img = cap.read()
    img = detector.findHands(img)                    # Finding the hand
    lmlist, bbox = detector.findPosition(img)          # Getting position of hand

    if len(lmlist)!=0:
        x1, y1 = lmlist[8][1:]
        x2, y2 = lmlist[12][1:]

        fingers = detector.fingersUp()      # Checking if fingers are upwards
        cv2.rectangle(img, (frameR, frameR), (width - frameR, height - frameR), (255, 0, 255), 2)   # Creating boundary box
        if fingers[1] == 1 and fingers[2] == 0:     # If fore finger is up and middle finger is down
            x3 = np.interp(x1, (frameR,width-frameR), (0,screen_width))
            y3 = np.interp(y1, (frameR, height-frameR), (0, screen_height))

            curr_x = prev_x + (x3 - prev_x)/smoothening
            curr_y = prev_y + (y3 - prev_y) / smoothening

            autopy.mouse.move(screen_width - curr_x, curr_y)    # Moving the cursor
            cv2.circle(img, (x1, y1), 7, (255, 0, 255), cv2.FILLED)
            prev_x, prev_y = curr_x, curr_y

        if fingers[1] == 1 and fingers[2] == 1:     # If fore finger & middle finger both are up
            length, img, lineInfo = detector.findDistance(8, 12, img)

            if length < 40:     # If both fingers are really close to each other
                cv2.circle(img, (lineInfo[4], lineInfo[5]), 15, (0, 255, 0), cv2.FILLED)
                autopy.mouse.click()    # Perform Click

    cTime = time.time()
    fps = 1/(cTime-pTime)
    pTime = cTime
    cv2.putText(img, str(int(fps)), (20, 50), cv2.FONT_HERSHEY_PLAIN, 3, (255, 0, 0), 3)
    cv2.imshow("Image", img)
    cv2.waitKey(1)
```