A VISION BASE APPLICATION FOR VIRTUAL MOUSE INTERFACE USING HAND GESTURE

Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Technology in the field of Computer Science and Engineering

##### **BY**

Sourav Sahoo (123180703089)

Sayan Ghosh(123180703078)

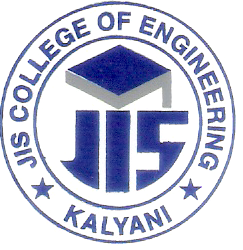
Indrani Naskar(123180703037)

Sankha Sarkar (380117011033)

Under the supervision

of

**Sumanta Chatterjee**



##### Department of Computer Science and Engineering

**JIS College of Engineering**

Block-A, Phase-III, Kalyani, Nadia, Pin-741235

West Bengal, India

May, 2022



**CERTIFICATE**

This is to certify that **Sourav Sahoo(123180703089),Sayan Ghosh(123180703078),Indrani Naskar(123180703037),Sankha sarkar(380117011033)** have completed their project entitled “A VISION BASE APPLICATION FOR VIRTUAL MOUSE INTERFACE USING HAND GESTURE”**,** under the guidance of **Sumanta Chatterjee** in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** from JIS college of Engineering (An Autonomous Institute)is an authentic record of their own work carried out during the academic year 2021-22 and to the best of our knowledge, this work has not been submitted elsewhere as part of the process of obtaining a degree, diploma, fellowship or any other similar title.

**--------------------------------- ------------------------------- ------------------------------**

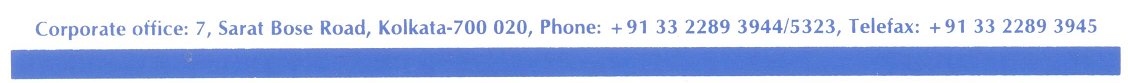
**Signature of theSupervisor Signature of the HOD Signature of the Principal**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature of the External Expert**

**Place:**

**Date:**



**ACKNOWLEDGEMENTS**

The analysis of the project work wishes to express our gratitude to Sumanta Chatterjee for allowing the degree attitude and providing effective guidance in development of this project work. His conscription of the topic and all the helpful hints, he provided, contributed greatly to successful development of this work, without being pedagogic and overbearing influence.

We also express our sincere gratitude to Dr. Dharmpal Singh, Head of the Department of Computer Science and Engineering of JIS College of Engineering and all the respected faculty members of Department of CSE for giving the scope of successfully carrying out the project work.

Finally, we take this opportunity to thank to Prof. **(Dr.) Partha Sarkar**, Principal of JIS College of Engineering for giving us the scope of carrying out the project work.

**…………………………………….………………………………….**

Sayan Ghosh

B.TECH in Computer Science and Engineering

4thYEAR/8th SEMESTER

Univ Roll--123180703078

**…………………………………….………………………………….**

Sourav Sahoo

B.TECH in Computer Science and Engineering

4thYEAR/8th SEMESTER

Univ Roll--123180703089

**…………………………………….………………………………….**

IndraniNaskar

B.TECH in Computer Science and Engineering

4thYEAR/8th SEMESTER

Univ Roll--123180703033

**…………………………………….………………………………….**

Sankha Sarkar

B.TECH in Computer Science and Engineering

4thYEAR/8th SEMESTER

Univ Roll--38011701103

# TABLE OF CONTENT

|  |  |
| --- | --- |
| **Title Page** | **1** |
| **Certificate** | **2** |
| **Acknowledgement** | **3** |
| **List of Figures & Tables** | **4** |
|  |  |
| **Abstract** | **7** |
| **1.Introduction** | **8** |
| **2.literature Survey** | **10** |
| **3.Methodology** | **26** |
| **4.Result and Discussion** | **47** |
| **5.Conclusion** | **63** |
| **6.Limitation** | **64** |
| **7.Futurework** | **64** |
| **8. Application** | **65** |
| **9.Reference** | **66** |
| **Publication from the work** | **52** |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **Fig.No** | **Description** |
| 1 | Mechanical mouse, with top cover removed |
| 2 | Optical Mouse, with top cover removed |
| 3 | MediaPipe hand recognition graph |
| 4 | Co-ordinates or land marks in the hand |
| 5 | The Flow Chart of Portable Vision-Based Human Computer Interaction |
| 6 | Agile method overview |
| 7 | Gesture for the computer to perform left button click. |
| 8 | Gesture for the computer to perform right button click. |
| 9 | Gesture for the computer to perform scroll up function. |
| 10 | Virtual mouse stabilization modulated by placing the actual cursor plane |
| 11 | Graph of Accuracy for different performed actions |
| 12 | Accuracy graph for different available mouse system |

**LIST OF TABLES**

|  |  |
| --- | --- |
| **Fig.No** | **Description** |
| 1 | Advantage and disadvantage of the Mechanical Mouse |
| 2 | Advantage and disadvantage of the Optical and Laser Mouse |
| 3 | Specification of pycharm |
| 4 | Gesture for specific mouse functions |
| 5 | Verification Plan P1 |
| 6 | Different brightness testing results |
| 7 | Verification Plan P2 |
| 8 | Verification Plan P3 |
| 9 | Distance testing results |
| 10 | Experimental results & accuracy |

**ABSTRACT**

This paper proposes a way of controlling the position of the mouse cursor with the help of a fingertip without using any electronic device. We can be performing the operations like clicking and dragging objects easily with help of different hand gestures. The proposed system will require only a webcam as an input device. Python and OpenCV will be required to implement this software. The output of the webcam will be displayed on the system’s screen so that it can be further calibrated by the user. The python dependencies that will be used for implementing this system are NumPy, Autopy and Mediapipe. The algorithm is trained so that hands can be detected using the camera. Hence, the proposed system will prevent the spread of Covid-19 by eliminating the need for human intervention and dependency on physical devices to control the computer system.

# **1. INTRODUCTION**

A mouse, in computing terms is a pointing device that detects two-dimensional movements relative to a surface. This movement is converted into the movement of a pointer on a display that allows to control the Graphical User Interface (GUI) on a computer platform. There are a lot of different types of mouse that have already existed in the modern days technology, there's the mechanical mouse that determines the movements by a hard rubber ball that rolls around as the mouse is moved. Years later, the optical mouse was introduced that replace the hard rubber ball to a LED sensor to detects table top movement and then sends off the information to the computer for processing. On the year 2004, the laser mouse was then introduced to improve the accuracy movement with the slightest hand movement, it overcome the limitations of the optical mouse which is the difficulties to track high-gloss surfaces. However, no matter how accurate can it be, there are still limitations exist within the mouse itself in both physical and technical terms. For example, a computer mouse is a consumable hardware device as it requires replacement in the long run, either the mouse buttons were degraded that causes inappropriate clicks, or the whole mouse was no longer detected by the computer itself.Since the computer was developed, it has become an indispensable part of our lives. With the growing importance of computers in day-to-day lives, peoples’ thoughts have naturally focused on making their use easy and convenient. For this reason, studies on human-computer interaction (HCI) have been actively conducted. HCI aims at creating or improving computing systems, including their functionality, reliability, usability, and efficiency. From this viewpoint, the user interface (UI) is an important part of HCI, leading to several studies on the same. UI has gradually evolved from a command-line interface (CLI) used to communicate with a computer using simple commands, to a graphical user interface (GUI). Recently, studies on a natural user interface/natural user experience (NUI/NUX) have been carried out. GUI is currently the most widely-used type for communication between the user and computer, with input devices such as a mouse and keyboard. On the other hand, NUI/NUX is characterized by the use of a user’s natural gestures, voice, etc., to communicate with the computer without any special input devices. To accomplish this, machine learning, image processing, and signal processing algorithms are used for gesture and voice recognition. However, because these algorithms are very difficult to implement and require a large amount of time for training to achieve recognition, it is difficult to realize an NUI/NUX. Therefore, as an alternative, it is necessary to develop a UI with a simple implementation and excellent performance. Recently, Microsoft’s Kinect has gained great popularity among the general public and developers as a development tool for NUI/NUX. Kinect can easily obtain depth information that cannot be accessed using a regular camera through a sensor. This can assist in the recognition of a gesture or voice, through the software development kit (SDK) provided, which can handle image processing and speech recognition without additional algorithms. Kinect also implements a skeleton image, which has the advantage of making it easy to obtain the location and depth information for each joint. Examples of studies using Kinect include a system that selects the objects in a virtual world through a Kinect camera and software for mobility-impaired seniors . In addition, a study on a hand mouse, which is the basis of this research, was conducted in 2014 . In this study, the proposed system is a hand-mouse interface that introduces a new concept called a ‘‘virtual monitor’’ to extract a user’s physical features in real time. We can easily obtain a user’s physical features using Kinect. A virtual monitor is generated based on a user’s physical features. It is possible for a user to accurately control a mouse pointer using their hands. Moreover, the virtual monitor is intuitive, because it is used like a touch screen. Through this, we propose an intuitive hand-mouse interface with better accuracy than previous such interfaces. In section 2, we review the previous work on an NUI/NUX- related hand mouse. Section 3 describes the hand-mouse interface design and implementation. In this section, we explain the total flow of the hand-mouse interface. We then explain how to create the virtual monitor, coordinate the transformation algorithm, etc. Section 4 presents an experimental evaluation of the accuracy and intuitiveness. To verify the accuracy, we compare the results with those of another hand mouse, using OpenCV to demonstrate the superiority of the proposed system.

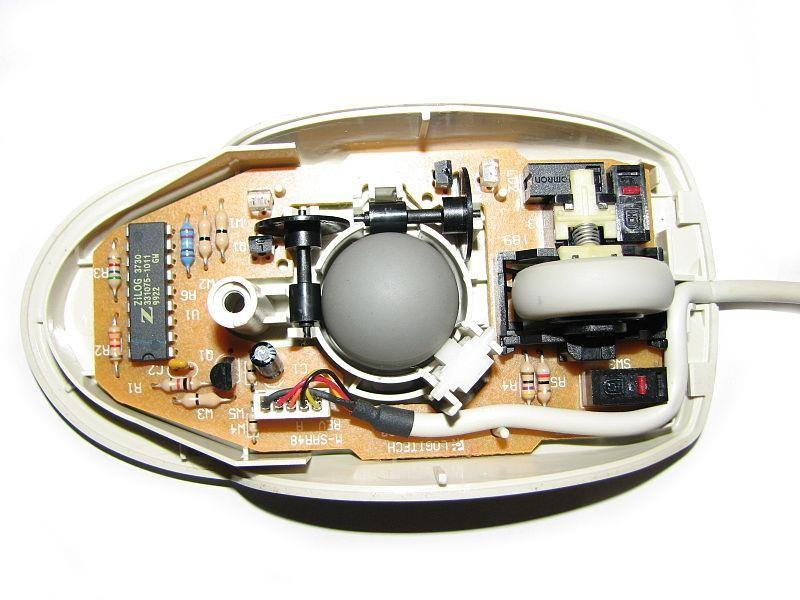
Despite the limitations, the computer technology continues to grow, so does the importance of the human computer interactions. Ever since the introduction of a mobile device that can be interact with touch screen technology, the world is starting to demand the same technology to be applied on every technological devices, this includes the desktop system. However, even though the touch screen technology for the desktop system is already exists, the price can be very steep. Therefore, a virtual human computer interaction device that replaces the physical mouse or keyboard by using a webcam or any other image capturing devices can be an alternative way for the touch screen. This device, which is the webcam, will be constantly utilized by a software that monitors the gestures given by the user in order to process it and translate to motion of pointes, as similar to a physical mouse.

## 1.1 Review of the Physical Mouse

It is known that there are various types of physical computer mouse in the modern technology, the following will discuss about the types and differences about the physical mouse.  Computer mouse is a hand-held [pointing device](https://en.wikipedia.org/wiki/Pointing_device) that detects [two-dimensional](https://en.wikipedia.org/wiki/Two-dimensional_space) motion relative to a surface. This motion is typically translated into the motion of a [pointer](https://en.wikipedia.org/wiki/Pointer_(user_interface)) on a [display](https://en.wikipedia.org/wiki/Computer_monitor), which allows a smooth control of the [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) of a [computer](https://en.wikipedia.org/wiki/Computer).

## 1.2 Mechanical Mouse

The [trackball](https://en.wikipedia.org/wiki/Trackball), a related pointing device, was invented in 1946 by [Ralph Benjamin](https://en.wikipedia.org/wiki/Ralph_Benjamin) as part of a post-[World War II](https://en.wikipedia.org/wiki/World_War_II)-era [fire-control](https://en.wikipedia.org/wiki/Fire-control_system) [radar](https://en.wikipedia.org/wiki/Radar) plotting system called the [Comprehensive Display System](https://en.wikipedia.org/wiki/Comprehensive_Display_System) (CDS). Benjamin was then working for the British [Royal Navy](https://en.wikipedia.org/wiki/Royal_Navy) Scientific Service. Benjamin's project used [analog computers](https://en.wikipedia.org/wiki/Analog_computer) to calculate the future position of target aircraft based on several initial input points provided by a user with a [joystick](https://en.wikipedia.org/wiki/Joystick). Benjamin felt that a more elegant [input device](https://en.wikipedia.org/wiki/Input_device) was needed and invented what they called a "roller ball" for this purpose.The device was patented in 1947,but only a prototype using a metal ball rolling on two rubber-coated wheels was ever built, and the device was kept as a military secret.Another early trackball was built by [Kenyon Taylor](https://en.wikipedia.org/wiki/Kenyon_Taylor), a British [electrical engineer](https://en.wikipedia.org/wiki/Electrical_engineering) working in collaboration with Tom Cranston and Fred Longstaff. Taylor was part of the original [Ferranti Canada](https://en.wikipedia.org/wiki/Ferranti_Canada), working on the [Royal Canadian Navy](https://en.wikipedia.org/wiki/Royal_Canadian_Navy)'s [DATAR](https://en.wikipedia.org/wiki/DATAR) (Digital Automated Tracking and Resolving) system in 1952.DATAR was similar in concept to Benjamin's display. The trackball used four disks to pick up motion, two each for the X and Y directions. Several rollers provided mechanical support. When the ball was rolled, the pickup discs spun and contacts on their outer rim made periodic contact with wires, producing pulses of output with each movement of the ball. By counting the pulses, the physical movement of the ball could be determined. A [digital computer](https://en.wikipedia.org/wiki/Computer#Vacuum_tubes_and_digital_electronic_circuits) calculated the tracks and sent the resulting data to other ships in a task force using [pulse-code modulation](https://en.wikipedia.org/wiki/Pulse-code_modulation) radio signals. This trackball used a standard Canadian [five-pin bowling](https://en.wikipedia.org/wiki/Five-pin_bowling) ball. It was not patented, since it was a secret military project.Known as the trackball mouse that is commonly used in the 1990s, the ball within the mouse are supported by two rotating rollers in order to detect the movement made by the ball itself. One roller detects the forward/backward motion while the other detects the left/right motion. The ball within the mouse are steel made that was covered with a layer of hard rubber, so that the detection are more precise. The common functions included are the left/right buttons and a scroll-wheel. However, due to the constant friction made between the mouse ball and the rollers itself, the mouse are prone to degradation, as overtime usage may cause the rollers to degrade, thus causing it to unable to detect the motion properly, rendering it useless. Furthermore, the switches in the mouse buttons are no different as well, as long term usage may cause the mechanics within to be loosed and will no longer detect any mouse clicks till it was disassembled and repaired.[Douglas Engelbart](https://en.wikipedia.org/wiki/Douglas_Engelbart) of the Stanford Research Institute (now [SRI International](https://en.wikipedia.org/wiki/SRI_International)) has been credited in published books by [Thierry Bardini](https://en.wikipedia.org/wiki/Thierry_Bardini),[Paul Ceruzzi](https://en.wikipedia.org/wiki/Paul_Ceruzzi),[Howard Rheingold](https://en.wikipedia.org/wiki/Howard_Rheingold),and several others as the inventor of the computer mouse. Engelbart was also recognized as such in various obituary titles after his death in July 2013.By 1963, Engelbart had already established a research lab at SRI, the [Augmentation Research Center](https://en.wikipedia.org/wiki/Augmentation_Research_Center) (ARC), to pursue his objective of developing both hardware and software computer technology to "augment" human intelligence. That November, while attending a conference on computer graphics in [Reno, Nevada](https://en.wikipedia.org/wiki/Reno,_Nevada), Engelbart began to ponder how to adapt the underlying principles of the [planimeter](https://en.wikipedia.org/wiki/Planimeter) to inputting X- and Y-coordinate data.On 14 November 1963, he first recorded his thoughts in his personal notebook about something he initially called a "[bug](https://en.wikipedia.org/wiki/Bug_(computer_mouse))", which in a "3-point" form could have a "drop point and 2 orthogonal wheels".He wrote that the "bug" would be "easier" and "more natural" to use, and unlike a stylus, it would stay still when let go, which meant it would be "much better for coordination with the keyboard".In 1964, [Bill English](https://en.wikipedia.org/wiki/Bill_English_(computer_engineer)) joined ARC, where he helped Engelbart build the first mouse prototype.[[2]](https://en.wikipedia.org/wiki/Computer_mouse#cite_note-Bardini_2000-3)[[24]](https://en.wikipedia.org/wiki/Computer_mouse#cite_note-25) They christened the device the *mouse* as early models had a cord attached to the rear part of the device which looked like a tail, and in turn resembled the common [mouse](https://en.wikipedia.org/wiki/Mouse).[[25]](https://en.wikipedia.org/wiki/Computer_mouse#cite_note-26) According to Roger Bates, a hardware designer under English, another reason for choosing this name was because the cursor on the screen was also referred to as "CAT" at this time.As noted above, this "mouse" was first mentioned in print in a July 1965 report, on which English was the lead author.On 9 December 1968, Engelbart publicly demonstrated the mouse at what would come to be known as [The Mother of All Demos](https://en.wikipedia.org/wiki/The_Mother_of_All_Demos). Engelbart never received any royalties for it, as his employer SRI held the patent, which expired before the mouse became widely used in personal computers.[[26]](https://en.wikipedia.org/wiki/Computer_mouse#cite_note-27) In any event, the invention of the mouse was just a small part of Engelbart's much larger project of augmenting human intellect.Several other experimental pointing-devices developed for Engelbart's oN-Line System ([NLS](https://en.wikipedia.org/wiki/NLS_(computer_system))) exploited different body movements – for example, head-mounted devices attached to the chin or nose – but ultimately the mouse won out because of its speed and convenience. The first mouse, a bulky device (pictured) used two [potentiometers](https://en.wikipedia.org/wiki/Potentiometer) perpendicular to each other and connected to wheels: the rotation of each wheel translated into motion along one [axis](https://en.wikipedia.org/wiki/Coordinate_system).[[30]](https://en.wikipedia.org/wiki/Computer_mouse#cite_note-31) At the time of the "Mother of All Demos", Engelbart's group had been using their second generation, 3-button mouse for about a year.



|  |  |
| --- | --- |
| Advantage | Disadvantage |
| * Allows the users to control the computer system by moving the mouse. * Provides precise mouse tracking   movements | * Prone to degradation of the mouse rollers and button switches, causing to be faulty. * Requires a flat surface to operate. |

**Figure 1:** Mechanical mouse, with top cover remo

**Advantages and disvantages of using Mechanical Mouse :**

## 1.3 Optical And Laser Mouse

A mouse that commonly used in these days, the motions of optical mouse rely on the Light Emitting Diodes (LEDs) to detect movements relative to the underlying surface, while the laser mouse is an optical mouse that uses coherent laser lights. Comparing to its predecessor, which is the mechanical mouse, the optical mouse no longer rely on the rollers to determine its movement, instead it uses an imaging array of photodiodes. The purpose of implementing this is to eliminate the limitations of degradation that plagues the current predecessor, giving it more durability while offers better resolution and precision. However, there's still some downside, even-though the optical mouse are functional on most opaque diffuse surface, it's unable to detect motions on the polished surface. Furthermore, long term usage without a proper cleaning or maintenance may leads to dust particles trap between the LEDs, which will cause both optical and laser mouse having surface detection difficulties. Other than that, it's still prone to degradation of the button switches, which again will cause the mouse to function improperly unless it was disassembled and repaired.Modern surface-independent optical mice work by using an [optoelectronic](https://en.wikipedia.org/wiki/Optoelectronic) [sensor](https://en.wikipedia.org/wiki/Sensor) (essentially, a tiny low-resolution video camera) to take successive images of the surface on which the mouse operates. As computing power grew cheaper, it became possible to embed more powerful special-purpose [image-processing](https://en.wikipedia.org/wiki/Image_processing) [chips](https://en.wikipedia.org/wiki/Integrated_circuit) in the mouse itself. This advance enabled the mouse to detect relative motion on a wide variety of surfaces, translating the movement of the mouse into the movement of the cursor and eliminating the need for a special mouse-pad. A surface-independent coherent light optical mouse design was patented by Stephen B. Jackson at Xerox in 1988.

The first commercially available, modern optical computer mice were the [Microsoft IntelliMouse](https://en.wikipedia.org/wiki/Microsoft_IntelliMouse) with IntelliEye and IntelliMouse Explorer, introduced in 1999 using technology developed by Hewlett-Packard. It worked on almost any surface, and represented a welcome improvement over mechanical mice, which would pick up dirt, track capriciously, invite rough handling, and need to be taken apart and cleaned frequently. Other manufacturers soon followed Microsoft's lead using components manufactured by the HP spin-off [Agilent Technologies](https://en.wikipedia.org/wiki/Agilent_Technologies), and over the next several years mechanical mice became obsolete. The technology underlying the modern optical computer mouse is known as [digital image correlation](https://en.wikipedia.org/wiki/Digital_image_correlation), a technology pioneered by the defense industry for tracking military targets. A simple binary-image version of digital image correlation was used in the 1980 Lyon optical mouse. Optical mice use image sensors to image naturally occurring texture in materials such as wood, cloth, mouse pads and [Formica](https://en.wikipedia.org/wiki/Formica_(plastic)). These surfaces, when lit at a grazing angle by a light emitting diode, cast distinct shadows that resemble a hilly terrain lit at sunset. Images of these surfaces are captured in continuous succession and compared with each other to determine how far the mouse has moved.To understand how [optical flow](https://en.wikipedia.org/wiki/Optical_flow) is used in optical mice, imagine two photographs of the same object except slightly offset from each other. Place both photographs on a [light table](https://en.wikipedia.org/wiki/Light_table) to make them transparent, and slide one across the other until their images line up. The amount that the edges of one photograph overhang the other represents the offset between the images, and in the case of an optical computer mouse the distance it has moved.Optical mice capture one thousand successive images or more per second. Depending on how fast the mouse is moving, each image will be offset from the previous one by a fraction of a pixel or as many as several pixels. Optical mice mathematically process these images using cross correlation to calculate how much each successive image is offset from the previous one.An optical mouse might use an image sensor having an 18 × 18 pixel array of monochromatic pixels. Its sensor would normally share the same [ASIC](https://en.wikipedia.org/wiki/Application-specific_integrated_circuit) as that used for storing and processing the images. One refinement would be accelerating the correlation process by using information from previous motions, and another refinement would be preventing deadbands when moving slowly by adding interpolation or frame-skipping.The development of the modern optical mouse at Hewlett-Packard Co. was supported by a succession of related projects during the 1990s at HP Laboratories. In 1992 William Holland was awarded US Patent 5,089,712 and John Ertel, William Holland, Kent Vincent, Rueiming Jamp, and Richard Baldwin were awarded US Patent 5,149,980 for measuring linear paper advance in a printer by correlating images of paper fibers. Ross R. Allen, David Beard, Mark T. Smith, and Barclay J. Tullis were awarded US Patents 5,578,813 (1996) and 5,644,139 (1997) for 2-dimensional optical navigational (i.e., position measurement) principles based on detecting and correlating microscopic, inherent features of the surface over which the navigation sensor travelled, and using position measurements of each end of a linear (document) image sensor to reconstruct an image of the document. This is the freehand scanning concept used in the HP CapShare 920 handheld scanner. By describing an optical means that explicitly overcame the limitations of wheels, balls, and rollers used in contemporary computer mice, the optical mouse was anticipated. These patents formed the basis for US Patent 5,729,008 (1998) awarded to Travis N. Blalock, Richard A. Baumgartner, Thomas Hornak, Mark T. Smith, and Barclay J. Tullis, where surface feature image sensing, image processing, and image correlation was realized by an integrated circuit to produce a position measurement. Improved precision of 2D optical navigation, needed for application of optical navigation to precise 2D measurement of media (paper) advance in HP DesignJet large format printers, was further refined in US Patent 6,195,475 awarded in 2001 to Raymond G. Beausoleil, Jr., and Ross R. Allen.While the reconstruction of the image in the document scanning application (Allen et al.) required resolution by the optical navigators on the order of 1/600th of an inch, implementation of optical position measurement in computer mice not only benefit from the cost reductions inherent in navigating at lower resolution, but also enjoy the advantage of visual feedback to the user of the cursor position on the computer display. In 2002, [Gary Gordon](https://en.wikipedia.org/wiki/Gary_Babcock_Gordon), Derek Knee, Rajeev Badyal and Jason Hartlove were awarded US Patent 6,433,780[[12]](https://en.wikipedia.org/wiki/Optical_mouse#cite_note-12) for an optical computer mouse that measured position using image correlation. Some small [trackpads](https://en.wikipedia.org/wiki/Trackpad) (such as those on Blackberry smartphones) work like an optical mouse.



**Figure 2:** Optical Mouse, with top cover removed

The following table describes the advantages and disadvantages of the Optical and Laser Mouse.

**Table 2:** Advantage and disadvantage of the Optical and Laser Mouse

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Allows better precision with lesser hand movements. * Longer life-span. | * Prone to button switches degradation. * Does not function properly while on a polished surface. |

## LITERATURE SURVEY

**Robertson, P., Laddaga, R. and Van Kleek, M., 2004, January. Virtual mouse vision based interface. In *Proceedings of the 9th international conference on Intelligent user interfaces* (pp. 177-183).no**

The work in this paper is focusing on virtual mouse in Tangible Interfaces, Multimedia Interfaces, Computer Vision, Intelligent UI for better presentation and design.this vision based mouse interface is described that the utilizes a robotic head and visual tracking of the user head and hand positions and recognization of the user.the user directed this proposed system by their hand gesture sign and control an intelligent koisk.the user interface support among other things, smooth control of the mouse pointer and buttons using hand signs and movements. The algorithms and architecture of real-time vision and robot controller are described.This project is introduce to reduce the increasing amounts of time holding unplanned meetings in public spaces, such as along corridors, or in lounges or kitchenettes.these spaces harbor a relaxed social atmosphere, where people feel naturally inclined to gather and talk casually about anything that may be on their minds.

As a result, these spaces encourage social connections to be made, shared interests to be discovered, and, perhaps most importantly, collaborations to form among people who may otherwise never have realized the opportunity to work together.The field of Intelligent Environments is concerned with studying how to use technological aids to improve the experience of humans in working, living, moving and other structured spaces. Ubiquitous computing1 is devoted to changing the relationship between humans and the computers with which we interact, towards allowing computers to become invisible and recede into the periphery of people’s lives.Ubiquitous computing is concerned with bringing the same degree of naturalness of interaction to the personal and business computers that are currently proliferating our work and play environments.The primary prototype test bed for the e21 architecture is embodied in a conference room known as The Intelligent Room.Three Kiosk prototypes were designed, for four separate environments. These were the lobby prototype, the hallway prototype, and the lounge prototype.This Ki/o installation will differ from the lobby prototype in several significant ways. First, users will most likely be interacting with it in a group, sitting around the glass table.The challenges faced the author with thw new display configuration of user interaction but the problem was sloved by the laser pointer.In this project work the author did not provide any limitation of this system.In this paper the future work is very narrow.

**Visual panel: virtual mouse, keyboard and 3D controller with an ordinary piece of paper by** Z. Zhang, Y. Wu, Y. Shan and S. Shafer .The paper was published on

The work in this project is focusing on a vision baised interface system that uses visual panel which employs an arbitrary quadrangle-shaped panel (e.g., an ordinary piece of paper) and a tip pointer (e.g., fingertip) as an intuitive, wireless and mobile input device.

## The system can accurately and reliably track the panel and the tip pointer. The panel tracking continuously determines the projective mapping between the panel at the current position and the display, which in turn maps the tip position to the corresponding position on the display.

As modern technology of human computer interactions become important in our everyday lives, varieties of mouse with all kind of shapes and sizes were invented, from a casual office mouse to a hard-core gaming mouse. However, there are some limitations to this hardware as they are not as environmental friendly as it seems. For example, the physical mouse requires a flat surface to operate, not to mention that it requires a certain area to fully utilize the

functions offered. Furthermore, some of this hardware are useless when it comes to interact with the computers remotely due to the cable lengths limitations, rendering it inaccessible.

In this paper the work in distributed in different virtual device like mouse, keyboard, joystick and even as a toy remote controller. This is obviously a cost effective approach .They have developed a vision-based interface prototype system called VISUAL PANEL that takes advantage of an arbitrary quadrangle shaped planar object as a panel such that a user can use any tip pointer such as his fingertip to interact with the computer. This proposed system capture a 3d image of the hand and then recognize the hand gesture and then accordingly perform the operations. They had also used the concept of dynamic programming. The step involved to implement this project is Quadrangle Representation, Automatic Detection, Trackingthrough Dynamic Program and more. The limitation of this project work is we have to colour our hand with different colour so that the system can recognize the gesture performed by the user.

**A three-dimensional virtual mouse generates synthetic training data for behavioral analysis by** [**Luis A. Bolaños**](http://europepmc.org/search/?scope=fulltext&page=1&query=AUTH:%22Bola%26%23x000f1%3Bos%20LA%22)**,**[**Dongsheng Xiao**](http://europepmc.org/search/?scope=fulltext&page=1&query=AUTH:%22Xiao%20D%22)**,**[**Nancy L. Ford**](http://europepmc.org/search/?scope=fulltext&page=1&query=AUTH:%22Ford%20NL%22)**,**[**Jeff M. LeDue**](http://europepmc.org/search/?scope=fulltext&page=1&query=AUTH:%22LeDue%20JM%22)**,**[**Pankaj K. Gupta**](http://europepmc.org/search/?scope=fulltext&page=1&query=AUTH:%22Gupta%20PK%22)**,. This paper is published on 2021**

### The work in this project is foucusing on 3D synthetic animated mouse based on CT scans that is actuated using animation and semi-random, joint-constrained movements to generate synthetic behavioral data with ground-truth label locations.The method include in this project is Blender, CT scan preprocessing and mouse behavioral videos. 3D model creation,Rigging of the model,Synthetic data generation. Rendering and data export. In addition, more.in this work two camera is used that is the main drawback. They have not proposed any limitation or future work for this system.

Hand Gestures - Virtual Mouse for Human Computer Interaction by Sherin Mohammed Sali Shajideen and Preetha V H from Department of Electronics and Communication SCT College of Engineering Trivandrum, India.This paper is published in International Conference on Smart Systems and Inventive Technology (ICSSIT 2018).

The work in this project is focusing on HCI module based virtual mouse and controlled by hand gesture. In this project work mainly working with National User Interface (NUI) and MATHLAB softwear.The implementation step involved in this work are Detection of Hand region, Hand Feature Tracking, Estimation of 3D Pointing Direction,At the time of processing the image two usb camera is used.This system can perform the operation with upto 90% accuracy.

**IMPLEMENTATION OF VIRTUAL MOUSE BASED ON MACHINE VISION by LI WENS HENG1 , DENG CHUNJIAN2 , LV YI3.This paper was published on**[**The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding**](https://ieeexplore.ieee.org/xpl/conhome/5697534/proceeding)**.**

The work in this project is focusing on colour fingertip biased virtual mouse with hand gesture. Color-based target tracking methods are normally based on various target-color filters and region segmentation techniques. These methods have many advantages. First, processing color is much more computationally inexpensive. Second, color models are scale/orientation/rotation invariant. In this work, they put forward a method to implement virtual mouse based on fingertip tracking that simulate mouse operation through hand gesture and make it possible for user to control the window-based applications by hand gesture. This project present an efficient algorithm based on color to track the movement of multiple fingertips in real time. Second, this system solve the problem due to the difference of resolution between the input image from the camera and the monitor in order to make the motion of the mouse pointer on the monitor smooth. The framework mainly consists of two parts: The first is the server of virtual mouse, the second is virtual mouse based applications (the client). The server is responsible for the detection and tracking of fingertips, constructing appropriate message and sending such messages to the client. The client responses to the messages and obtains the coordinates of multiple fingertips by calling the API function provided by the server, then does the appropriate message processing to achieve the control of the application. The server is divided into three parts: adaptive on-line training of fingertip color, tracking of fingertips, and mouse message simulating. The first part obtains the color histogram of fingertip through a small sample area of fingertip. The second part is fingertip detection and tracking. It first calculates the probability density of the video image through back-projection on the color histogram and gets the binary image by doing binarization, then detects the targets of fingertip through edge detection, and finally it locates the centroid of fingertip through Mean Shift Algorithm. The third part constructs the messages according to the results of fingertip tracking and sending them to the client. Since the project is use a database to communicate in between the client and the server. If any case the database is damaged the client server, relationship is break and the operations are not executed. This is major drawback of the project.

**Adaptable Virtual Keyboard and Mouse for People with Special Needs by Alexandre Henzen from Graduate Program in Electrical and Computer Engineering Federal University of Technology Curitiba, Brazil and Percy Nohama from Graduate Program in Electrical and Computer Engineering Federal University of Technology Curitiba, Brazil.This paper was published on *IEEE 2016 Future Technologies Conference (FTC) - San Francisco, CA, USA in 2016.***

*This work in this project is* focusing on the keyboard and mouse via sensors connected to a user's body part in which it maintains motor control. The application developed constitutes a keyboard and mouse emulator, configurable and adaptable, which was named ETM (Emulator Keyboard and Mouse). The software captures trigger’s signals connected to the USB (Universal Serial Bus), through the adaptation of the contacts of the buttons of a joystick. It optimized communication and user learning, minimized motor limitations due to neurological diseases, and it has expanded his accessibility and digital and social inclusion. In this Enhanced Thematic Mapper (ETM

) is used. This is to prevent the disabled user to unwillingly interfere with the setting parameters. Delays, sizes, sensors choices, background and foreground colors are among settings (configurations) offered to customize the software. It is a voice controllable system so in the noisy environment that system cannot work properly.

**Hand Gesture Recognition Based Virtual Mouse Events by**[**Manav Ranawat**](https://ieeexplore.ieee.org/author/37088893709)**;**[**Madhur Rajadhyaksha**](https://ieeexplore.ieee.org/author/37088894650)**;**[**Neha Lakhani**](https://ieeexplore.ieee.org/author/37088893738)**;**[**Radha Shankarmani**](https://ieeexplore.ieee.org/author/37548047300)**.This paper is published on** [**2021 2nd International Conference for Emerging Technology (INCET)**](https://ieeexplore.ieee.org/xpl/conhome/9456097/proceeding)**.**

The work in this project is focusing on Hand Gesture Recognition Based Virtual Mouse by using python, opencv and pyautoGUI. In this work the hand gesture are recognized very well by the system and perform the mouse function very accurately. This is a modern python libery biased work so no extra things are needed like caps or colour taps. This perticulcular system can work accurately in any lighting condition.

**Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition by Vantukala VishnuTeja Reddy1 , Thumma Dhyanchand2 , Galla Vamsi Krishna3 , Satish Maheshwaram4 Department of Electronics and Communication Engineering, National Institute of Technology Warangal, Warangal, India – 506004.This paper is published on** [**IEEE-HYDCON**](https://ieeexplore.ieee.org/xpl/conhome/9242647/proceeding)**(2020).**

The work in this project is focusing on virtual mouse implemented with finger tip recognition and hand gesture tracking based on image in a live video is one of the studies. In this paper, virtual mouse control using fingertip identification and hand gesture recognition. Image processing, a division of signal processing, can consists of an image or a video as input and output as an image or various parameters of it. Gesture recognition and tracking is a kind of image processing process.In this project work three different colour caps is used and are placed on the fingertip and then the system detected mouse operation is performed.

**2.1 Problem Statement**

It is no surprised that every technological devices have its own limitations, especially when it comes to computer devices. After the review of various type of the physical mouse, the problems are identified and generalized. The following describes the general problem that the current physical mouse suffers:

* Physical mouse is subjected to mechanical wear and tear.
* Physical mouse requires special hardware and surface to operate.
* Physical mouse is not easily adaptable to different environments and its performance varies depending on the environment.
* Mouse has limited functions even in present operational environments.
* All wired mouse and wireless mouse have its own lifespan.

Sixth Sense technology is a gesture based wearable interface that links the digital information around us with the physical world and it allows us to use our natural hand gestures to communicate or interact with the digital information. Several approaches have been presented on the concept of virtual mouse with different ideas. The approaches were done in which involved the concept of Image Processing and Image Acquisition. According to the study, the motto is to make a virtual mouse which is mainly useful for saving manual work. The future modification can use complex mouse workings using this simple image processing technique. By this concept real world is interacting and getting well with the digital world using the concept of this technology known to be as Sixth Sense. Many works are done using Sixth sense technology some even uses IOT interaction with it as the use of RFID tags and image processing for potholes detection to overcome accidents that’s a main problem in many parts of the world. Another work by the author, they are using different colour cap, we have to wear the caps before the experiments.so that is the a limiatation for this project.we are not using any caps or colour taps to recognize the finger gesture.Another work that is done in the similar domain is that train autonomous cars using block chain methods for faster and safer experience, the autonomous cars can use a review or ratting system which can help them to stack up which road is safe and shortest, this way a healthy route can be created for the autonomous industry be it cars or other autonomous vehicles.Image base one time password is also a factor nowadays to enhance the security of One Time Passwords it also includes machine learning algorithms for detection of image OTP’s Smart Image attendance based systems are also in use now a days which makes the attendance system more error free and less time consuming image processing tools and algorithm are used for student face detection. In the field of medical science the use of machine learning and Artificial Intelligence is growing at an extraordinary pace image processing are used to classify heart attack traits and diabetes traits even using skin images the type of infection or malignancy can be tested . The many works that have been done on this Sixth Sense Technology also included use of mouse by colour recognition, that is working of the mouse with colour coordinated finger caps the image captures the colour and coordinates or maps the mouse with its position

## 2.2 Motivation of Virtual Mouse

It is fair to say that the Virtual Mouse will soon to be substituting the traditional physical mouse in the near future, as people are aiming towards the lifestyle where that every technological devices can be controlled and interacted remotely without using any peripheral devices such as the remote, keyboards, etc. it doesn't just provides convenience, but it's cost effective as well.

## 2.3 Convenient

It is known in order to interact with the computer system, users are required to use an actual physical mouse, which also requires a certain area of surface to operate, not to mention that it suffers from cable length limitations. Virtual Mouse requires none of it, as it only a webcam to allow image capturing of user's hand position in order to determine the position of the pointers that the user want it to be. For example, the user will be able to remotely control and interact the computer system by just facing the webcam or any other image capturing devices and moving your fingers, thus eliminating the need to manually move the physical mouse, while able to interact with the computer system from few feet away.

## 2.4 Cost Effective

A quality physical mouse is normally cost from the range of 200 to 500 rupees, depending on their functionality and features. Since the Virtual Mouse requires only a webcam, a physical mouse are no longer required, thus eliminating the need to purchase one, as a single webcam is sufficient enough to allow users to interact with the computer system through it, while some other portable computer system such as the laptop, are already supplied with a built-in webcam, could simply utilize the Virtual Mouse software without having any concerns about purchasing any external peripheral devices.

## 2.5 Project Scope

Virtual Mouse that will soon to be introduced to replace the physical computer mouse to promote convenience while still able to accurately interact and control the computer system. To do that, the software requires to be fast enough to capture and process every image, in order to successfully track the user's gesture. Therefore, this project will develop a software application with the aid of the latest software coding technique and the open-source computer vision library also known as the OpenCV. The scope of the project is as below:

* Real time application.
* User friendly application.
* Removes the requirement of having a physical mouse.

The process of the application can be started when the user's gesture was captured in real time by the webcam, which the captured image will be processed for segmentation to identify which pixels values equals to the values of the defined colour. After the segmentation is completed, the overall image will be converted to Binary Image where the identified pixels will show as white, while the rest are black. The position of the white segment in the image will be recorded and set as the position of the mouse pointer, thus resulting in simulating the mouse pointer without using a physical computer mouse. The software application is compatible with the Windows platform. The functionality of the software will be coded with C++ programming language code with the integration of an external library that does the image processing known as the OpenCV.

## 2.6 Project Objective

The purpose of this project is to develop a Virtual Mouse application that targets a few aspects of significant development. For starters, this project aims to eliminate the needs of having a physical mouse while able to interact with the computer system through webcam by using various image processing techniques. Other than that, this project aims to develop a Virtual Mouse application that can be operational on all kind of surfaces and environment.

The following describes the overall objectives of this project:

* To design to operate with the help of a webcam.The Virtual Mouse application will be operational with the help of a webcam, as the webcam are responsible to capture the images in real time. The application would not work if there are no webcam detected.
* To design a virtual input that can operate on all surface.

The Virtual Mouse application will be operational on all surface and indoor environment, as long the users are facing the webcam while doing the motion gesture.

* To program the camera to continuously capturing the images, which the images will be analyzed, by using various image processing techniques.As stated above, the Virtual Mouse application will be continuously capturing the images in real time, where the images will be undergo a series of process, this includes HSV conversion, Binary Image conversion, salt and pepper noise filtering, and more.
* To convert hand gesture/motion into mouse input that will be set to a particular screen position.The Virtual Mouse application will be programmed to detect the position of the defined colours where it will be set as the position of the mouse pointers. Furthermore, a combination of different colors may result in triggering different types of mouse events, such as the right/left clicks, scroll up/down, and more.

## 2.7 Impact, Significance and Contribution

The Virtual Mouse application is expected to replace the current methods of utilizing a physical computer mouse where the mouse inputs and positions are done manually. This application offers a more effortless way to interact with the computer system, where every task can be done by gestures. Furthermore, the Virtual Mouse application could assist the motor-impaired users where he/she could interact with the computer system by just showing the correct combination of colors to the webcam.

## 2.8 Portable Vision-Based Human Computer Interaction (HCI)

Another "Ubiquitous Computing" approach proposed by Chu-Feng Lien (2015), requires only finger-tips to control the mouse cursor and click events. The proposed system doesn't requires hand-gestures nor colour tracking in order to interact with the system, instead it utilize a feature name Motion History Images(MHI) , a method that used to identify movements with a row of images in time. The proposed system are not capable to detect fast moving movements as the frame-rates are not able to keep up, thus leading to increase of error rate. Furthermore, due to the mouse click events occurred when the finger hold on a certain positions, this may lead to user constant finger movements to prevent false alarm, which may result inconvenience. To detect the colours, they have utilized the MATLAB's built in "*subtract*" function, with the combination of the noise filtering by using median filter, which are effective in filtering out or at least reduce the "salt and pepper" noise. The captured image will be converted to Binary Scale Image by using MATLAB's built in "*im2bw*" function to differentiate the possible values for each pixel. When the conversion is done, the captured image will undergo another filtering process by using "*bwareaopen*" to remove the small areas in order to get an accurate number of the object detected in the image.

Even though the proposed system possess good accuracy in a well-controlled environment, it does has its own limitations. The proposed system are not capable to detect fast moving movements as the frame-rates are not able to keep up, thus leading to increase of error rate. Furthermore, due to the mouse click events occurred when the finger hold on a certain positions, this may lead to user constant finger movements to prevent false alarm, which may result inconvenience.

* + 1. **2.9 Hardware Requirement**

The following describes the hardware needed in order to execute and develop the Virtual Mouse application:

**2.9.1 Computer Desktop or Laptop**

The computer desktop or a laptop will be utilized to run the visual software in order to display what webcam had captured. A notebook which is a small, lightweight and inexpensive laptop computer is proposed to increase mobility.

System will be using

Processor : Core i5

Main Memory : 4GB RAM

Hard Disk : 500GB

Display : 14" Monitor

**2.9.2 Webcam**

A webcam is a [video camera](https://en.wikipedia.org/wiki/Video_camera) that feeds or [streams](https://en.wikipedia.org/wiki/Streaming_media) an image or video in real time to or through a [computer network](https://en.wikipedia.org/wiki/Computer_network), such as the [Internet](https://en.wikipedia.org/wiki/Internet). Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a [video chat](https://en.wikipedia.org/wiki/Videotelephony) session involving two or more people, with conversations that include live audio and video.Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires much [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing)), such streams usually use [compressed formats](https://en.wikipedia.org/wiki/Video_compression). The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions. The term "webcam" (a [clipped compound](https://en.wikipedia.org/wiki/Clipped_compound)) may also be used in its original sense of a [video camera](https://en.wikipedia.org/wiki/Video_camera) connected to the [Web](https://en.wikipedia.org/wiki/World_Wide_Web) continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its [web page](https://en.wikipedia.org/wiki/Web_page) over the Internet. Some of them, for example, those used as online [traffic cameras](https://en.wikipedia.org/wiki/Traffic_camera), are expensive, rugged [professional video cameras](https://en.wikipedia.org/wiki/Professional_video_camera).Webcams are known for their low [manufacturing](https://en.wikipedia.org/wiki/Manufacturing) cost and their high flexibility, making them the lowest-cost form of [videotelephony](https://en.wikipedia.org/wiki/Videotelephony). As webcams evolved simultaneously with display technologies, USB interface speeds and broadband internet speeds, the resolution went up from gradually from 320×240, to 640×480, and some now even offer 1280×720 (aka [720p](https://en.wikipedia.org/wiki/720p)) or 1920×1080 (aka [1080p](https://en.wikipedia.org/wiki/1080p)) resolution.

Webcams can come with different presets and Fields of View (FOV). Individual users can make use of less than 90° Horizontal FOV for home offices and live streaming. Webcams with as much as 360° Horizontal FOV can be used for small- to medium- sized rooms (sometimes even large rooms). Depending on the users' purposes, webcams in the market can display the whole room or just the general vicinity.Despite the low cost,the [resolution](https://en.wikipedia.org/wiki/Image_resolution) offered as of 2019 is impressive, with now the low-end webcams offering resolutions of [720p](https://en.wikipedia.org/wiki/720p), mid-range webcams offering [1080p](https://en.wikipedia.org/wiki/1080p) resolution, and high-end webcams offering 4K resolution at 60 fps.Webcams have become a source of security and privacy issues, as some built-in webcams can be remotely activated by [spyware](https://en.wikipedia.org/wiki/Spyware). To address this concern, many webcams come with a physical lens cover.

## 2.10 Software Requirement

The following describes the software needed in-order to develop the Virtual Mouse application:

**2.10.1 Python Language**

Python is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)), [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Its design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with the use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule).

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Type_system#DYNAMIC) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). It is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).[Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) began working on Python in the late 1980s as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)) and first released it in 1991 as Python 0.9.0.Python 2.0 was released in 2000 and introduced new features such as [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [cycle-detecting](https://en.wikipedia.org/wiki/Cycle_detection) garbage collection, [reference counting](https://en.wikipedia.org/wiki/Reference_counting), and [Unicode](https://en.wikipedia.org/wiki/Unicode) support. Python 3.0, released in 2008, was a major revision that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility) with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.Python was conceived in the late 1980s[]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-venners-interview-pt-1-41)by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) at [Centrum Wiskunde & Informatica](https://en.wikipedia.org/wiki/Centrum_Wiskunde_%26_Informatica) (CWI) in the [Netherlands](https://en.wikipedia.org/wiki/Netherlands) as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)), which was inspired by [SETL](https://en.wikipedia.org/wiki/SETL), capable of [exception handling](https://en.wikipedia.org/wiki/Exception_handling) and interfacing with the [Amoeba](https://en.wikipedia.org/wiki/Amoeba_(operating_system)) operating system.Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "[benevolent dictator for life](https://en.wikipedia.org/wiki/Benevolent_dictator_for_life)", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.Python 2.0 was released on 16 October 2000, with many major new features.Python 3.0, released on 3 December 2008, with many of its major features [backported](https://en.wikipedia.org/wiki/Backporting) to Python 2.6.x and 2.7.x. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)) was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No further security patches or other improvements will be released for it. With Python 2's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)), only Python 3.6.xand later were supported. Later, support for 3.6 was also discontinued. In 2021, Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues leading to possible [remote code execution](https://en.wikipedia.org/wiki/Remote_code_execution)and [web cache poisoning](https://en.wikipedia.org/wiki/Cache_poisoning).

In 2022, Python 3.10.4 and 3.9.12 were expeditedand so were older releases including 3.8.13, and 3.7.13 because of many security issues in 2022.

Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support functional programming and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including [metaprogramming](https://en.wikipedia.org/wiki/Metaprogramming) and [metaobjects](https://en.wikipedia.org/wiki/Metaobject) [magic methods] ). Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract)and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing) and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management).It uses dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

Its design offers some support for functional programming in the [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) tradition. It has filter,mapandreduce functions; [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [dictionaries](https://en.wikipedia.org/wiki/Associative_array), sets, and [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)) expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)) and [Standard ML](https://en.wikipedia.org/wiki/Standard_ML).

Its core philosophy is summarized in the document The [Zen of Python](https://en.wikipedia.org/wiki/Zen_of_Python) (PEP 20), which includes [aphorisms](https://en.wikipedia.org/wiki/Aphorism) such as:

* Beautiful is better than ugly.
* Explicit is better than implicit.
* Simple is better than complex.
* Complex is better than complicated.
* Readability counts.

Rather than building all of its functionality into its core, Python was designed to be highly [extensible](https://en.wikipedia.org/wiki/Extensibility) via modules. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with [ABC](https://en.wikipedia.org/wiki/ABC_(programming_language)), which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to [Perl](https://en.wikipedia.org/wiki/Perl)'s "[there is more than one way to do it](https://en.wikipedia.org/wiki/There_is_more_than_one_way_to_do_it)" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" philosophy.[Alex Martelli](https://en.wikipedia.org/wiki/Alex_Martelli), a [Fellow](https://en.wikipedia.org/wiki/Fellow) at the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation) and Python book author, wrote: "To describe something as 'clever' is not considered a compliment in the Python culture."Python's developers strive to avoid [premature optimization](https://en.wikipedia.org/wiki/Premature_optimization) and reject patches to non-critical parts of the [CPython](https://en.wikipedia.org/wiki/CPython) reference implementation that would offer marginal increases in speed at the cost of clarity.[[71]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-20-71) When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C; or use [PyPy](https://en.wikipedia.org/wiki/PyPy), a [just-in-time compiler](https://en.wikipedia.org/wiki/Just-in-time_compilation). [Cython](https://en.wikipedia.org/wiki/Cython) is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.Python's developers aim for it to be fun to use. This is reflected in its name—a tribute to the British comedy group [Monty Python](https://en.wikipedia.org/wiki/Monty_Python)[[72]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-24-72)—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (a reference to a [Monty Python sketch](https://en.wikipedia.org/wiki/Spam_(Monty_Python))) instead of the standard [foo and bar](https://en.wikipedia.org/wiki/Foobar).A common [neologism](https://en.wikipedia.org/wiki/Neologism) in the Python community is pythonic, which has a wide range of meanings related to program style. "Pythonic" code may use Python idioms well, be natural or show fluency in the language, or conform with Python's minimalist philosophy and emphasis on readability. Code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic.Python users and admirers, especially those considered knowledgeable or experienced, are often referred to as Pythonistas.

**2.10.2 Pycharm IDE**

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive [Python](https://www.jetbrains.com/help/pycharm/python.html), [web](https://www.jetbrains.com/help/pycharm/web-frameworks.html), and [data science](https://www.jetbrains.com/help/pycharm/scientific-tools.html) development.

PyCharm is available in three editions:

* *Community* (free and [open-sourced](https://github.com/JetBrains/intellij-community/blob/master/LICENSE.txt)): for smart and intelligent Python development, including code assistance, refactorings, visual debugging, and version control integration.
* *Professional* ([paid](https://www.jetbrains.com/pycharm/buy/#commercial?billing=yearly)) : for professional Python, web, and data science development, including code assistance, refactorings, visual debugging, version control integration, remote configurations, deployment, support for popular web frameworks, such as Django and Flask, database support, scientific tools (including Jupyter notebook support), big data tools.
* *Edu* (free and [open-sourced](https://github.com/JetBrains/intellij-community/blob/master/LICENSE.txt)): for learning programming languages and related technologies with integrated educational tools.

### Supported languages﻿

To start developing in Python with PyCharm you need to download and install Python from [python.org](http://www.python.org/) depending on your platform.

PyCharm supports the following versions of Python:

* **Python 2:** version 2.7
* **Python 3:** from the version 3.6 up to the version 3.11

Besides, in the *Professional* edition, one can develop Django , Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to Settings | Plugins or PyCharm | Preferences | Plugins for macOS users, to find out more or set them up during the first IDE launch).

### Supported platforms﻿

PyCharm is a cross-platform IDE that works on Windows, macOS, and Linux. Check the system requirements:

**Table 3:** pycharm Specification

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Minimum** | **Recommended** |
| RAM | 4 GB of free RAM | 8 GB of total system RAM |
| CPU | Any modern CPU | Multi-core CPU. PyCharm supports multithreading for different operations and processes making it faster the more CPU cores it can use. |
| Disk space | 2.5 GB and another 1 GB for caches | SSD drive with at least 5 GB of free space |
| Monitor resolution | 1024×768 | 1920×1080 |
| Operating system | * Microsoft Windows 8 or later * macOS 10.14 or later * Any Linux distribution that supports Gnome, KDE , or Unity DE. PyCharm is not available for some Linux distributions, | Latest 64-bit version of Windows, macOS, or Linux (for example, Debian, Ubuntu, or RHEL) |

**2.10.3 Numpy**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.NumPy (**Numerical Python**) is an open source Python library that’s used in almost every field of science and engineering. It’s the universal standard for working with numerical data in Python, and it’s at the core of the scientific Python and PyData ecosystems. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.

The NumPy library contains multidimensional array and matrix data structures (you’ll find more information about this in later sections). It provides **ndarray**, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

**2.10.3 Open CV**

OpenCV (*Open Source Computer Vision Library*) is a [library of programming functions](https://en.wikipedia.org/wiki/Library_(computing)) mainly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision).[[1]](https://en.wikipedia.org/wiki/OpenCV#cite_note-1) Originally developed by [Intel](https://en.wikipedia.org/wiki/Intel_Corporation), it was later supported by [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage) then Itseez (which was later acquired by Intel[]](https://en.wikipedia.org/wiki/OpenCV#cite_note-2)). The library is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) and free for use under the [open-source](https://en.wikipedia.org/wiki/Open-source_software) [Apache 2 License](https://en.wikipedia.org/wiki/Apache_License). Starting with 2011, OpenCV features GPU acceleration for real-time operations.OpenCV are also included in the making of this program. **OpenCV (O**pen **S**ource **C**omputer **V**ision) is a library of programming functions for real time computer vision. OpenCV have the utility that can read image pixels value, it also have the ability to create real time eye tracking and blink detection.

Officially launched in 1999 the OpenCV project was initially an [Intel Research](https://en.wikipedia.org/wiki/Intel_Research_Lablets) initiative to advance [CPU](https://en.wikipedia.org/wiki/Central_processing_unit)-intensive applications, part of a series of projects including [real-time](https://en.wikipedia.org/wiki/Real-time_computing) [ray tracing](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)) and [3D display](https://en.wikipedia.org/wiki/3D_Display) walls.[[4]](https://en.wikipedia.org/wiki/OpenCV#cite_note-KaehlerBradski2016-4) The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

* Advance vision research by providing not only open but also [optimized code](https://en.wikipedia.org/wiki/Code_optimization) for basic vision infrastructure. No more [reinventing the wheel](https://en.wikipedia.org/wiki/Reinventing_the_wheel).
* Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
* Advance vision-based commercial applications by making [portable](https://en.wikipedia.org/wiki/Portability_(computer_science)), performance-optimized code available for free – with a license that did not require code to be open or free itself.

The first alpha version of OpenCV was released to the public at the [IEEE Conference on Computer Vision and Pattern Recognition](https://en.wikipedia.org/wiki/Conference_on_Computer_Vision_and_Pattern_Recognition) in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.

The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the [C++](https://en.wikipedia.org/wiki/C%2B%2B) interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team supported by commercial corporations.In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.In May 2016, Intel signed an agreement to acquire Itseez,[]](https://en.wikipedia.org/wiki/OpenCV#cite_note-9) a leading developer of OpenCV.In July 2020, OpenCV announced and began a Kickstarter campaign for the [OpenCV AI Kit](https://opencv.org/introducing-oak-spatial-ai-powered-by-opencv/), a series of hardware modules and additions to OpenCV supporting Spatial AI.The first alpha version of OpenCV was released to the public at the [IEEE Conference on Computer Vision and Pattern Recognition](https://en.wikipedia.org/wiki/Conference_on_Computer_Vision_and_Pattern_Recognition) in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the [C++](https://en.wikipedia.org/wiki/C%2B%2B) interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six monthsand development is now done by an independent Russian team supported by commercial corporations.In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.In May 2016, Intel signed an agreement to acquire Itseez,[[9]](https://en.wikipedia.org/wiki/OpenCV#cite_note-9) a leading developer of OpenCV.In July 2020, OpenCV announced and began a Kickstarter campaign for the [OpenCV AI Kit](https://opencv.org/introducing-oak-spatial-ai-powered-by-opencv/), a series of hardware modules and additions to OpenCV supporting Spatial AI.

OpenCV's application areas include:

* 2D and 3D feature toolkits
* [Egomotion](https://en.wikipedia.org/wiki/Egomotion) estimation
* [Facial recognition system](https://en.wikipedia.org/wiki/Facial_recognition_system)
* [Gesture recognition](https://en.wikipedia.org/wiki/Gesture_recognition)
* [Human–computer interaction](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction) (HCI)
* [Mobile robotics](https://en.wikipedia.org/wiki/Mobile_robotics)
* Motion understanding
* [Object detection](https://en.wikipedia.org/wiki/Object_detection)
* [Segmentation](https://en.wikipedia.org/wiki/Segmentation_(image_processing)) and recognition
* [Stereopsis](https://en.wikipedia.org/wiki/Stereopsis) stereo vision: depth perception from 2 cameras
* [Structure from motion](https://en.wikipedia.org/wiki/Structure_from_motion) (SFM)
* [Motion tracking](https://en.wikipedia.org/wiki/Video_tracking)
* [Augmented reality](https://en.wikipedia.org/wiki/Augmented_reality)

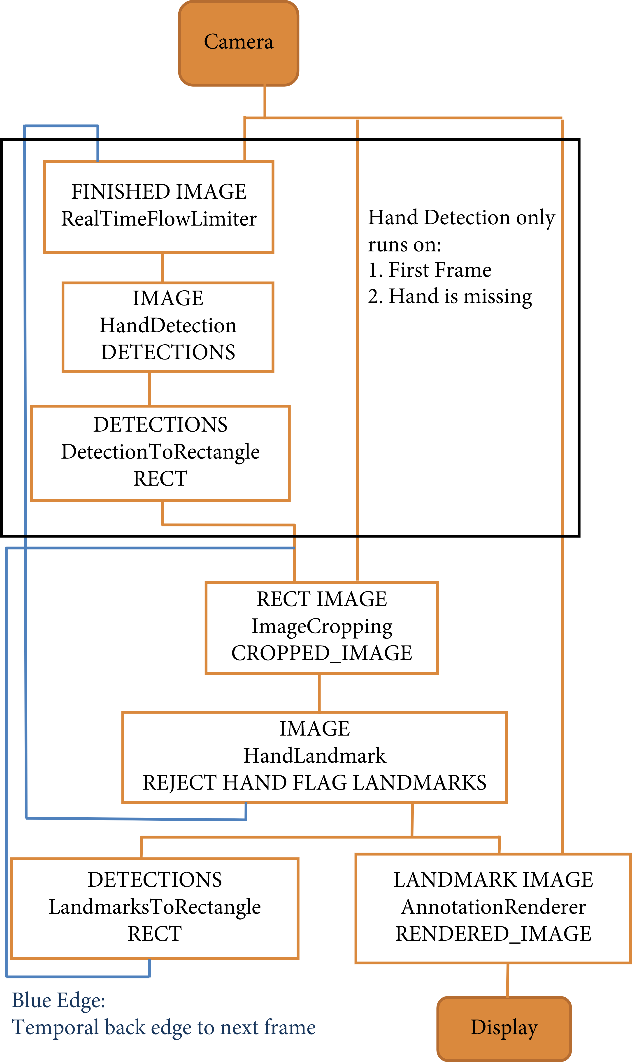
To support some of the above areas, OpenCV includes a statistical [machine learning](https://en.wikipedia.org/wiki/Machine_learning) library that contains:

* [Boosting](https://en.wikipedia.org/wiki/Boosting_(meta-algorithm))
* [Decision tree learning](https://en.wikipedia.org/wiki/Decision_tree_learning)
* [Gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting) trees
* [Expectation-maximization algorithm](https://en.wikipedia.org/wiki/Expectation-maximization_algorithm)
* [k-nearest neighbor algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm)
* [Naive Bayes classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)
* [Artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network)
* [Random forest](https://en.wikipedia.org/wiki/Random_forest)
* [Support vector machine](https://en.wikipedia.org/wiki/Support_vector_machine) (SVM)
* [Deep neural networks](https://en.wikipedia.org/wiki/Deep_neural_network) (DNN)[[11]](https://en.wikipedia.org/wiki/OpenCV#cite_note-DNN-11)

**2.10.5. Mediapipe**

MediaPipe is a framework which is used for applying in a machine learning pipeline, and it is an opensource framework of Google. The MediaPipe framework is useful for cross platform development since the framework is built using the time series data. The MediaPipe framework is multimodal, where this framework can be applied to various audios and videos . The MediaPipe framework is used by the developer for building and analyzing the systems through graphs, and it also been used for developing the systems for the application purpose.

The steps involved in the system that uses MediaPipeare carried out in the pipeline configuration. MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano. The pipeline created can run in various platforms allowing scalability in mobile and desktops. The MediaPipe framework is based on three fundamental parts; they are performance evaluation, framework for retrieving sensor data, and a collection of components which are called calculators , and they are reusable.



**Figure 3:**MediaPipe hand recognition graph

A pipeline is a graph which consists of components called calculators, where each calculator is connected by streams in which the packets of data flow through. Developers are able to replace or define custom calculators anywhere in the graph creating their own application. The calculators and streams combined create a data-flow diagram; the graph (Figure 1) is created with MediaPipe where each node is a calculator and the nodes are connected by stream

## Single-shot detector model is used for detecting and recognizing a hand or palm in real time. The single-shot detector model is used by the MediaPipe. First, in the hand detection module, it is first trained for a palm detection model because it is easier to train palms. Furthermore, the non maximum suppression works significantly better on small objects such as palms or fists . A model of hand landmark consists of locating 21 joint or knuckle co-ordinates in the hand region, as shown in Figure .

## C:\Users\sankha\Downloads\8133076.fig.002.png

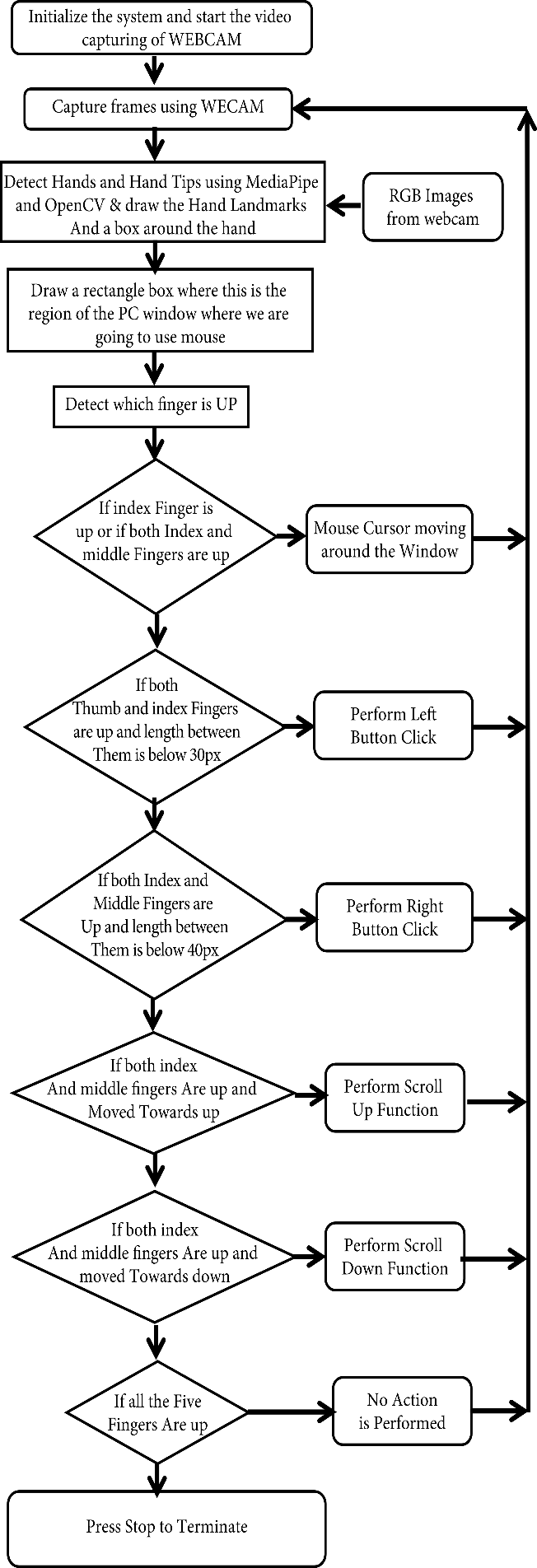
**Figure 4:**Co-ordinates or land marks in the hand

**2.10.6. Autopy**

# It is cross platform GUI automation module for python. That module keeps tracks of finger in this proposed system.Autopy track the fingertip and tell us which finger is up and which one is down This process is happing by giving system an input in the form of 0 and 1.From this module the mediapipe module takes output and done the process and give the proper output. From this outputopencv visualize everything and create the proper frame from image.

# **AutopyMoudule for working with mouse**

This module contains functions for getting the current state of and controlling the mouse cursor.

Unless otherwise stated, coordinates are those of a screen coordinate system, where the origin is at the top left.

**Figure 5**: The Flow Chart of Portable Vision-Based Human Computer Interaction

## 3. METHODOLOGY

Visualization of mouse actions for a given hand gesture is a challenging problem in the deep learning domain. In this article, we will use different techniques of computer vision and Deep learning to recognize the hand gestures and perform the mouse operations. We will build a Ai Virtual mouse by using Autopy and Mediapipe. Ai Virtual Mouse is a challenging HCI problem where a hand gesture work as a mouse. It requires both methods from computer vision to understand the content of the image and a language model from the field of Deep learning to turn the understanding of the image into gesture in the right order. Recently, deep learning methods have achieved state-of-the-art results on examples of this problem. Deep learning methods have demonstrated state-of-the-art results on gesture generation problems. What is most impressive about these methods is a single end-to-end model can be defined to predict a gesture, given a photo, instead of requiring sophisticated data preparation or a pipeline of specifically designed models.

The system can be broken down in three main modules.

**Image Acquisition:** We need a sensor for the system to detect the user's hand movements. As a sensor, the computer's camera is used. The webcam captures real-time video at a fixed frame rate and resolution determined by the camera's hardware. If necessary, the system allows you to change the frame rate and resolution. When the camera takes a picture, it inverts it. The pointer image moves to the right, and versa if we move the color pointer to the left. It is similar to the picture we get when we stand before a mirror, but it is used to avoid the picture flickering.

**Image processing and hand detection:** The computational complexity of a grey picture is less than a colorful one. This transforms the image into a grey image. All necessary operations were performed once the image was converted to greyscale. Then you have to subject a noise filter, smoothing, and threshold.

**Hand Gesture**: After completing the previous modules, the mouse movement, left-click, right-click, drag/select, scroll up and scroll down will be carried out with color caps or tapes on our fingers. For this project we'll be using the Agile Software Development methodology approach in developing the application. The stated approach is an alternative to the traditional waterfall model that helps the project team respond to unpredictability through incremental and iterative work. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible respond to change. The following describes the principles of the Agile Software Development methodology.

● Satisfy the customer by early and continuous delivery of workable software.

● Encourage changes of requirement.

● Workable software is delivered frequently.

● Continuous collaboration between the stakeholders and the developers.

● Project are developed around motivated individuals.

● Encourage informal meetings.

● Operational software is the principle measure of progress.

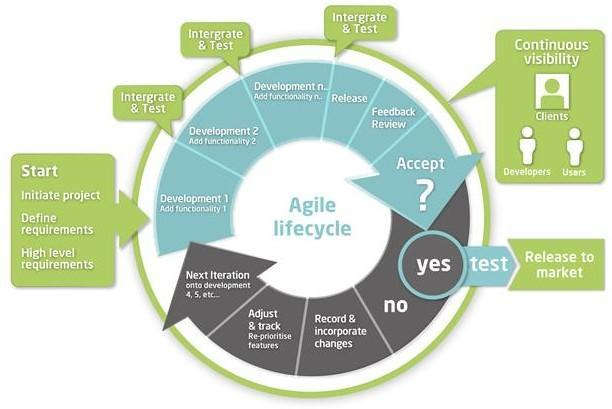
● Sustainable development, able to maintain a constant pace.

● Continuous attention to technical excellence and good design

● Simplicity

● Self-organizing teams

● Regular adaption to changing circumstances



**Figure 6:** Agile method overview

The reason for choosing this methodology is due to the fact that the Virtual Mouse are still

considered to be at the introduction stage, which means it still requires a great deal of extensive research and development before it could actually make it into the market. Therefore, this project requires a thorough yet iterative planning and requirements gathering where the lifecycle will be continually revisited to re- evaluates the direction of the project and to eliminate the ambiguities in the process of the development, and at the same time welcome changes of requirements, which promotes adaptability and flexibility. Furthermore, due to the Virtual Mouse application are more towards serving the users, this project requires continuous customer collaboration, as they're essential for gathering the proper requirements in all aspects. This is why that the agile methodology is the ideal approach for developing the project.

The following describes the phases within the agile methodology approach:

● Planning

A thorough planning will be conducted in this phase where the existing systems/product, for

this case, physical computer mouse will be reviewed and studied to identify the problems

existed, a comparison of problems will be made to compare which problems are more crucial

and requires improvement. An outline objective and the scope will be identified in order to

provide an alternative solution to the problem.

● Requirement Analysis

The phase that gathers and interpreting the facts, diagnosing problems and recommending

improvements to the system. In this phase, the collected problem statements will be extensively studied in order to find a proper solution or at least an improvements to the proposed system. All proposed solutions will be converted into requirements where it will be documented in a requirement specification.

● Designing

The requirement specification from the previous phase will be studied and prioritize to

determine which requirement are more important where the requirement with the highest

priority will be delivered first. After the study, the system design will be prepared as it helps

in defining the overall system architecture and specifying the hardware and the software

requirements.

● Building

The phase where the actual coding implementation takes place. By referring to the inputs from the system design, the system will be developed based on the prioritize requirements. However, due to we're using the agile methodology approach, the developed system will be considered 26 as a prototype system where it will be integrated and tested by the users.

● Testing

The phase where the prototype system going through a series of test. The prototype system will first undergo integration where the features from the previous iteration cycle are added to the latest cycle. After the integration, the prototype system will be thoroughly tested by the users to determine whether they are satisfied with the latest deliverables, the completion of the project depends on whether they've accepted it or otherwise. If the users requires additional features or modification, feedback gathering will be conducted, which resulted in further modification of the requirements and features where it will recorded and documented for the requirement analysis phase on the next iteration

**3.1 Import the required libraries**

First we have imported the required libery in pycharm IDE

##### **3.2 Capturing the Image and Processing**

For the proposed model, the system functions on the basis of frames that are captured by the in-built camera or a peripheral web camera. With the help of the Python library OpenCV, the camera window opens and starts capturing video. The web camera then sends these video frames to the AI mouse system.

**3.3 Preparation of Gesture Encoding**

The web camera keeps on collecting the frames until the underlying program is closed. The captured frames of video are collected in the BGR color format from the web camera. In order for OpenCV to process the frames, the BGR color format has to be converted to the RGB color format. Subsequently, OpenCV processes the frames to detect hand/s.

**3.4 Converting Mediapipe Landmarks To Recognizable Gestures**

At this point, the hand is being tracked and if any finger is held pointed, MediaPipe recognizes the finger and the tip with the help of the 21 co-ordinates on the fingers and after processing the gesture, the appropriate mouse operation is handled.

**3.4.2 Model complexity:**It is used to specify the complexity of the pose landmark model: 0, 1, or 2. As the model complexity of the model increases the landmark accuracy and latency increase. The default value is 1.

**3.4.3 Smooth landmarks:**This parameter is used to reduce the jitter in the prediction by filtering pose landmarks across different input images. The default value is True.

**3.4.4 Min detection confidence:**It is used to specify the minimum confidence value with which the detection from the person-detection model needs to be considered as successful. Can specify a value in [0.0,1.0]. The default value is 0.5.

**3.4.5 Min tracking confidence:**

It is used to specify the minimum confidence value with which the detection from the landmark-tracking model must be considered as successful. Can specify a value in [0.0,1.0]. The default value is 0.5.

Detecting Face and Hand landmarks from the image. Holistic model processes the image and produces landmarks for Face, Left Hand, Right Hand and also detects the Pose of the

1. Capture the frames continuously from the camera using OpenCV.
2. Convert the BGR image to an RGB image and make predictions using initialized holistic model.
3. The predictions made by the holistic model are saved in the results variable from which we can access the landmarks using results.face\_landmarks, results.right\_hand\_landmarks, results.left\_hand\_landmarks respectively.
4. Draw the detected landmarks on the image using the draw\_landmarks function from drawing utils.
5. Display the resulting Image.The holistic model produces 21 Left-Hand landmarks, and 21 Right-Hand landmarks. The individual landmarks can be accessed by specifying the index of the required landmark. Example: results.left\_hand\_landmarks.landmark[0]

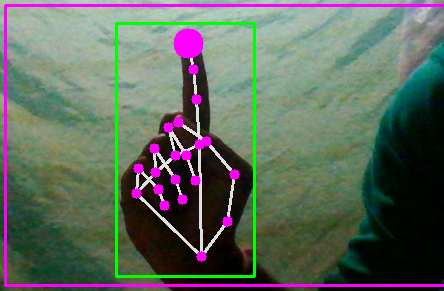
**3.5 Finding Gesture Using Current finger State**

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 and according to that, the particular mouse function is performed

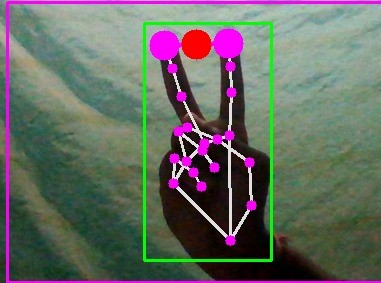
**3.6 Denoise**

* Need to delete noisy pixels from the image.
* We use an image morphology algorithm that performs image erosion and image dilation to eliminate noise .
* Erosion trims down the image area where the hand is not present .
* Dilation expands the area of the Image pixels which are not eroded.

**3.7 Finding finger tip:**

* The convex hull algorithm is used to solve the problem of finding the biggest polygon including all vertices.
* we can detect finger tips on the hand. We used this algorithm to recognize if a finger is folded or not.
* Check the distance between the center and a pixel which is in convex hull set.

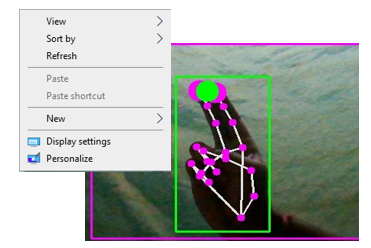
###### 3.7.1 For the Mouse to Perform Left Button Click

If both the index finger with tip Id = 1 and the thumb finger with tip Id = 0 are up and the distance between the two fingers is lesser than 30px, the computer is made to perform the left mouse button click using the pynput Python package, as shown in Figures 24.

**Figure 7**: Gesture for the computer to perform left button click.

###### 3.7.2 For the Mouse to Perform Right Button Click

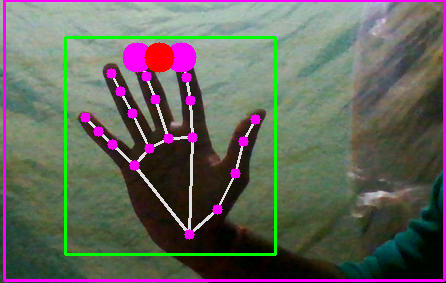
If both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up and the distance between the two fingers is lesser than 40 px, the computer is made to perform the right mouse button click using the pynput Python package, as shown in Figure 25.

**Figure 8**:Gesture for the computer to perform right button click.

**3.8 Stabilize The Cursor**

The typical desktop Windows, Icons, Menus, and Pointer (WIMP) interface is a special case of imageplane selection. Although not a true three dimensional representation of a desktop, common operating systems such as Microsoft Windows and the Windows system typically display windows in a depth ordered stack with the currently active window on top. The mouse cursor in these systems almost always remains visible in front of the desktop content. If we consider the mouse as residing somewhere in between the user viewpoint and the desktop windows, then we can see that selecting the first visible object with a ray cast from the viewpoint through the tip of the cursor emulates the familiar behavior of the mouse. The desktop situation is a simplified case of image-plane selection where the display is mono scopic, the user viewpoint remains fixed and the virtual cursor is restricted to moving in a plane a fixed distance between the viewpoint and desktop content.

Virtual mouse stabilization is modulated by placing the actual cursor plane a fraction of the distance d c between effective cursor plane and content underneath the virtual cursor.

****

**Figure 10**: Virtual mouse stabilization modulated by replacing the actual cursor plane and

Creating a virtual (PINK) border

**3.9 Hold the Position To Change Status**

At this stage, system will wait for 5 frames and they perform the operations. If the program execute the gesture, immediately many actions will not be recognizable by the system and the mouse does not work properly.

3.10 **Denoise**

* Need to delete noisy pixels from the image.
* We use an image morphology algorithm that performs image erosion and image dilation to eliminate noise .
* Erosion trims down the image area where the hand is not present .
* Dilation expands the area of the Image pixels which are not eroded.

## 4. RESULT AND DISCUSSION

**4.1 overview**

The Virtual Mouse Colour Recognition requires being able to recognize most of the colours provided by the users with high accuracy, consistency, and minimal performance impact on other processes. However, the recognition results may varies whenever the qualities of the captured frames have changed, as it may be affected by different situation in terms of environment, brightness, and the weather. The following describes the situations which may result in false detection and/or any other problem that may occur during recognition phase:

1. The real-time images are taken under dark or bright environment condition.
2. The real-time images are taken in a colour conflicts background.
3. The users interact with the program in near or far distance.
4. The real-time images are rotated in a clockwise or anti-clockwise rotation. In order to achieve greater accuracy and consistency throughout the whole recognition cycle, verification plan is required to be implemented in order for the program to perform flawlessly. The verification plans is as follows:

In order to achieve accuracy, and consistency of the Virtual Mouse colour recognition, testing phase have been conducted on various scenarios.

## 4.2 Performance in Various Environments

The purpose of testing phase is to ensure that the final deliverable is able to perform flawlessly in terms of accuracy, consistency, and performance. To achieve that, the program has to able to recognize the colours input provided by the users with minimal adjustment, provide that the coloursare thoroughly calibrated at first hand. Furthermore, the program is require to be able to execute the mouse functions efficiently and accurately as well.

The following describes the outcome of the program testing in various environments:

**Table 6:** Different brightness testing results

|  |  |
| --- | --- |
| **Brightness** | **Results** |
| **Normal Environment** |  |
|  | All Gestures are successfully recognized. The three highlighted circles indicate that the targeted finger are identified, compared, and execute a  ccordingly. |
| **Brighter Environment** |  |
|  | All Gestures are successfully recognized. The three highlighted circles indicate that the targeted finger are identified, compared, and execute accordingly. |
| **Dark Environment** |  |
|  | All Gestures are successfully recognized. The three highlighted circles indicate that the targeted finger are identified, compared, and execute accordingly. |

**Table 6:** Different brightness testing results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL No** | **Sample** | **Lighting Condition** | **Fps** | **Accuracy(%)** | **Action Performed** |
| 1 | 1 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 2 | 2 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 3 | 3 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 4 | 4 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 5 | 5 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 6 | 6 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 7 | 7 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 8 | 8 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 9 | 9 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 10 | 10 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 11 | 11 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 12 | 12 | >=1500 Lumens | 24 | 97 | Cursor Movement |
| 13 | 13 | >=1500 Lumens | 24 | 98 | Cursor Movement |
| 14 | 14 | >=1500 Lumens | 24 | 98 | Cursor Movement |
| 15 | 15 | >=1500 Lumens | 24 | 99 | Cursor Movement |
| 16 | 16 | >=1500 Lumens | 24 | 99 | Cursor Movement |
| 17 | 17 | >=1500 Lumens | 24 | 99 | Cursor Movement |
| 18 | 18 | >=1500 Lumens | 24 | 99 | Cursor Movement |
| 19 | 19 | >=1500 Lumens | 24 | 99 | Cursor Movement |
| 20 | 20 | >=1500 Lumens | 24 | 98 | Cursor Movement |
| 21 | 21 | >=1500 Lumens | 24 | 98 | Cursor Movement |
| 22 | 22 | >=1500 Lumens | 23 | 98 | Cursor Movement |
| 23 | 23 | >=1500 Lumens | 24 | 98 | Cursor Movement |
| 24 | 24 | >=1500 Lumens | 25 | 98 | Cursor Movement |
| 25 | 25 | >=1500 Lumens | 26 | 100 | Cursor Movement |
| 26 | 26 | >=1500 Lumens | 27 | 100 | Cursor Movement |
| 27 | 27 | >=1500 Lumens | 28 | 100 | Left Click |
| 28 | 28 | >=1500 Lumens | 29 | 100 | Left Click |
| 29 | 29 | >=1500 Lumens | 30 | 100 | Left Click |
| 30 | 30 | >=1500 Lumens | 31 | 100 | Left Click |
| 31 | 31 | >=1500 Lumens | 32 | 98 | Left Click |
| 32 | 32 | >=1500 Lumens | 33 | 98 | Left Click |
| 33 | 33 | >=1500 Lumens | 22 | 98 | Left Click |
| 34 | 34 | >=1500 Lumens | 22 | 98 | Left Click |
| 35 | 35 | >=1500 Lumens | 22 | 98 | Left Click |
| 36 | 36 | >=1500 Lumens | 22 | 98 | Left Click |
| 37 | 37 | >=1500 Lumens | 22 | 98 | Left Click |
| 38 | 38 | >=1500 Lumens | 24 | 98 | Left Click |
| 39 | 39 | >=1500 Lumens | 24 | 98 | Left Click |
| 40 | 40 | >=1500 Lumens | 24 | 98 | Left Click |
| 41 | 41 | >=1500 Lumens | 24 | 98 | Left Click |
| 42 | 42 | >=1500 Lumens | 24 | 98 | Left Click |
| 43 | 43 | >=1500 Lumens | 24 | 98 | Left Click |
| 44 | 44 | >=1500 Lumens | 24 | 98 | Left Click |
| 45 | 45 | >=1500 Lumens | 24 | 98 | Left Click |
| 46 | 46 | >=1500 Lumens | 24 | 98 | Right Click |
| 47 | 47 | >=1500 Lumens | 24 | 98 | Right Click |
| 48 | 48 | >=1500 Lumens | 23 | 98 | Right Click |
| 49 | 49 | >=1500 Lumens | 23 | 98 | Right Click |
| 50 | 50 | >=1500 Lumens | 23 | 98 | Right Click |
| 51 | 51 | >=1500 Lumens | 23 | 98 | Right Click |
| 52 | 52 | >=1500 Lumens | 23 | 98 | Right Click |
| 53 | 53 | >=1500 Lumens | 23 | 98 | Right Click |
| 54 | 54 | >=1500 Lumens | 23 | 98 | Right Click |
| 55 | 55 | >=1500 Lumens | 23 | 98 | Right Click |
| 56 | 56 | >=1500 Lumens | 23 | 98 | Right Click |
| 57 | 57 | >=1500 Lumens | 23 | 98 | Right Click |
| 58 | 58 | >=1500 Lumens | 23 | 98 | Right Click |
| 59 | 59 | >=1500 Lumens | 23 | 98 | Right Click |
| 60 | 60 | >=1500 Lumens | 23 | 98 | Right Click |
| 61 | 61 | >=1500 Lumens | 23 | 98 | Right Click |
| 62 | 62 | >=1500 Lumens | 23 | 98 | Right Click |
| 63 | 63 | >=1500 Lumens | 23 | 98 | Right Click |
| 64 | 64 | >=2300 Lumens | 25 | 98 | Right Click |
| 65 | 65 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 66 | 66 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 67 | 67 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 68 | 68 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 69 | 69 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 70 | 70 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 71 | 71 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 72 | 72 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 73 | 73 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 74 | 74 | >=2300 Lumens | 25 | 98 | Cursor Movement |
| 75 | 75 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 76 | 76 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 77 | 77 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 78 | 78 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 79 | 79 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 80 | 80 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 81 | 81 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 82 | 82 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 83 | 83 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 84 | 84 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 85 | 85 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 86 | 86 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 87 | 87 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 88 | 88 | >=2300 Lumens | 22 | 98 | Cursor Movement |
| 89 | 89 | >=2300 Lumens | 22 | 98 | Cursor Movement |
| 90 | 90 | >=2300 Lumens | 22 | 98 | Cursor Movement |
| 91 | 91 | >=2300 Lumens | 22 | 98 | Cursor Movement |
| 92 | 92 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 93 | 93 | >=2300 Lumens | 24 | 98 | Cursor Movement |
| 94 | 94 | >=2300 Lumens | 24 | 98 | Left Click |
| 95 | 95 | >=2300 Lumens | 24 | 98 | Left Click |
| 96 | 96 | >=2300 Lumens | 24 | 98 | Left Click |
| 97 | 97 | >=2300 Lumens | 24 | 98 | Left Click |
| 98 | 98 | >=2300 Lumens | 24 | 98 | Left Click |
| 99 | 99 | >=2300 Lumens | 24 | 98 | Left Click |
| 100 | 100 | >=2300 Lumens | 24 | 98 | Left Click |
| 101 | 101 | >=2300 Lumens | 24 | 98 | Left Click |
| 102 | 102 | >=2300 Lumens | 24 | 98 | Left Click |
| 103 | 103 | >=2300 Lumens | 25 | 98 | Left Click |
| 104 | 104 | >=2300 Lumens | 25 | 98 | Left Click |
| 105 | 105 | >=2300 Lumens | 25 | 98 | Left Click |
| 106 | 106 | >=2300 Lumens | 25 | 98 | Left Click |
| 107 | 107 | >=2300 Lumens | 25 | 98 | Left Click |
| 108 | 108 | >=2300 Lumens | 25 | 98 | Left Click |
| 109 | 109 | >=2300 Lumens | 25 | 98 | Left Click |
| 110 | 110 | >=2300 Lumens | 25 | 98 | Left Click |
| 111 | 111 | >=2300 Lumens | 25 | 98 | Right Click |
| 112 | 112 | >=2300 Lumens | 25 | 98 | Right Click |
| 113 | 113 | >=2300 Lumens | 25 | 98 | Right Click |
| 114 | 114 | >=2300 Lumens | 25 | 98 | Right Click |
| 115 | 115 | >=2300 Lumens | 25 | 98 | Right Click |
| 116 | 116 | >=2300 Lumens | 25 | 98 | Right Click |
| 117 | 117 | >=2300 Lumens | 25 | 98 | Right Click |
| 118 | 118 | >=2300 Lumens | 25 | 98 | Right Click |
| 119 | 119 | >=2300 Lumens | 25 | 98 | Right Click |
| 120 | 120 | >=2300 Lumens | 24 | 98 | Right Click |
| 121 | 121 | >=2300 Lumens | 24 | 98 | Right Click |
| 122 | 122 | >=2300 Lumens | 24 | 98 | Right Click |
| 123 | 123 | >=2300 Lumens | 24 | 98 | Right Click |
| 124 | 124 | >=2300 Lumens | 24 | 98 | Right Click |
| 125 | 125 | >=2300 Lumens | 24 | 98 | Right Click |
| 126 | 126 | >=2300 Lumens | 24 | 98 | Right Click |
| 127 | 127 | >=2300 Lumens | 24 | 98 | Right Click |
| 128 | 128 | >=2300 Lumens | 24 | 98 | Right Click |
| 129 | 129 | >=2300 Lumens | 24 | 98 | Right Click |
| 130 | 130 | >=2300 Lumens | 24 | 98 | Right Click |
| 131 | 131 | >=2300 Lumens | 24 | 98 | Right Click |
| 132 | 132 | >=2300 Lumens | 24 | 98 | Right Click |
| 133 | 133 | >=2500 lumans | 24 | 98 | Cursor Movement |
| 134 | 134 | >=2500 lumans | 24 | 98 | Cursor Movement |
| 135 | 135 | >=2500 lumans | 24 | 98 | Cursor Movement |
| 136 | 136 | >=2500 lumans | 25 | 98 | Cursor Movement |
| 137 | 137 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 138 | 138 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 139 | 139 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 140 | 140 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 141 | 141 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 142 | 142 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 143 | 143 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 144 | 144 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 145 | 145 | >=2500 lumans | 25 | 99 | Cursor Movement |
| 146 | 146 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 147 | 147 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 148 | 148 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 149 | 149 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 150 | 150 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 151 | 151 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 152 | 152 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 153 | 153 | >=2500 lumans | 25 | 100 | Cursor Movement |
| 154 | 154 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 155 | 155 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 156 | 156 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 157 | 157 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 158 | 158 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 159 | 159 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 160 | 160 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 161 | 161 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 162 | 162 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 163 | 163 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 164 | 164 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 165 | 165 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 166 | 166 | >=2500 lumans | 26 | 100 | Cursor Movement |
| 167 | 167 | >=2500 lumans | 26 | 100 | Left Click |
| 168 | 168 | >=2500 lumans | 26 | 100 | Left Click |
| 169 | 169 | >=2500 lumans | 26 | 100 | Left Click |
| 170 | 170 | >=2500 lumans | 26 | 100 | Left Click |
| 171 | 171 | >=2500 lumans | 26 | 100 | Left Click |
| 172 | 172 | >=2500 lumans | 26 | 100 | Left Click |
| 173 | 173 | >=2500 lumans | 24 | 97 | Left Click |
| 174 | 174 | >=2500 lumans | 24 | 97 | Left Click |
| 175 | 175 | >=2500 lumans | 24 | 97 | Left Click |
| 176 | 176 | >=2500 lumans | 24 | 97 | Left Click |
| 177 | 177 | >=2500 lumans | 24 | 97 | Left Click |
| 178 | 178 | >=2500 lumans | 24 | 97 | Left Click |
| 179 | 179 | >=2500 lumans | 24 | 97 | Left Click |
| 180 | 180 | >=2500 lumans | 24 | 97 | Left Click |
| 181 | 181 | >=2500 lumans | 24 | 97 | Left Click |
| 182 | 182 | >=2500 lumans | 24 | 97 | Left Click |
| 183 | 183 | >=2500 lumans | 24 | 97 | Left Click |
| 184 | 184 | >=2500 lumans | 24 | 97 | Left Click |
| 185 | 185 | >=2500 lumans | 24 | 97 | Left Click |
| 186 | 186 | >=2500 lumans | 24 | 100 | Right Click |
| 187 | 187 | >=2500 lumans | 24 | 100 | Right Click |
| 188 | 188 | >=2500 lumans | 24 | 100 | Right Click |
| 189 | 189 | >=2500 lumans | 24 | 100 | Right Click |
| 190 | 190 | >=2500 lumans | 24 | 100 | Right Click |
| 191 | 191 | >=2500 lumans | 24 | 99 | Right Click |
| 192 | 192 | >=2500 lumans | 23 | 100 | Right Click |
| 193 | 193 | >=2500 lumans | 23 | 100 | Right Click |
| 194 | 194 | >=2500 lumans | 25 | 100 | Right Click |
| 195 | 195 | >=2500 lumans | 25 | 100 | Right Click |
| 196 | 196 | >=2500 lumans | 25 | 100 | Right Click |
| 197 | 197 | >=2500 lumans | 25 | 100 | Right Click |
| 198 | 198 | >=2500 lumans | 25 | 100 | Right Click |
| 199 | 199 | >=2500 lumans | 25 | 100 | Right Click |
| 200 | 200 | >=2500 lumans | 25 | 100 | Right Click |

**Table 9:** Distance testing results

|  |  |
| --- | --- |
| **Distance** | **Result** |
| Near (15cm away from webcam) |  |
| C:\Users\sankha\Desktop\1ss.jpeg | All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly |
| **Far (25cm away from webcam)** |  |
|  | All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly. |
| **Farther (35cm away from webcam)** |  |
|  | All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SL No | Sample | Distance | FPS | Accuracy(%) | Action performed |
| 1 | 1 | >=15 Cm | 25 | 100 | Cursor Movement |
| 2 | 2 | >=15 Cm | 25 | 100 | Cursor Movement |
| 3 | 3 | >=15 Cm | 25 | 100 | Cursor Movement |
| 4 | 4 | >=15 Cm | 25 | 100 | Cursor Movement |
| 5 | 5 | >=15 Cm | 25 | 100 | Cursor Movement |
| 6 | 6 | >=15 Cm | 25 | 100 | Cursor Movement |
| 7 | 7 | >=15 Cm | 25 | 100 | Cursor Movement |
| 8 | 8 | >=15 Cm | 25 | 100 | Cursor Movement |
| 9 | 9 | >=15 Cm | 25 | 100 | Cursor Movement |
| 10 | 10 | >=15 Cm | 25 | 100 | Cursor Movement |
| 11 | 11 | >=15 Cm | 25 | 100 | Cursor Movement |
| 12 | 12 | >=15 Cm | 25 | 100 | Cursor Movement |
| 13 | 13 | >=15 Cm | 25 | 100 | Cursor Movement |
| 14 | 14 | >=15 Cm | 25 | 100 | Cursor Movement |
| 15 | 15 | >=15 Cm | 25 | 100 | Cursor Movement |
| 16 | 16 | >=15 Cm | 25 | 100 | Cursor Movement |
| 17 | 17 | >=15 Cm | 25 | 100 | Cursor Movement |
| 18 | 18 | >=15 Cm | 25 | 100 | Cursor Movement |
| 19 | 19 | >=15 Cm | 25 | 100 | Cursor Movement |
| 20 | 20 | >=15 Cm | 24 | 99 | Cursor Movement |
| 21 | 21 | >=15 Cm | 24 | 99 | Cursor Movement |
| 22 | 22 | >=15 Cm | 24 | 99 | Cursor Movement |
| 23 | 23 | >=15 Cm | 24 | 99 | Cursor Movement |
| 24 | 24 | >=15 Cm | 24 | 99 | Cursor Movement |
| 25 | 25 | >=15 Cm | 24 | 99 | Left Click |
| 26 | 26 | >=15 Cm | 24 | 99 | Left Click |
| 27 | 27 | >=15 Cm | 24 | 99 | Left Click |
| 28 | 28 | >=15 Cm | 24 | 99 | Left Click |
| 29 | 29 | >=15 Cm | 24 | 99 | Left Click |
| 30 | 30 | >=15 Cm | 24 | 99 | Left Click |
| 31 | 31 | >=15 Cm | 25 | 99 | Left Click |
| 32 | 32 | >=15 Cm | 25 | 99 | Left Click |
| 33 | 33 | >=15 Cm | 25 | 99 | Left Click |
| 34 | 34 | >=15 Cm | 25 | 99 | Left Click |
| 35 | 35 | >=15 Cm | 25 | 99 | Left Click |
| 36 | 36 | >=15 Cm | 25 | 99 | Left Click |
| 37 | 37 | >=15 Cm | 25 | 99 | Left Click |
| 38 | 38 | >=15 Cm | 25 | 99 | Left Click |
| 39 | 39 | >=15 Cm | 25 | 99 | Left Click |
| 40 | 40 | >=15 Cm | 25 | 99 | Left Click |
| 41 | 41 | >=15 Cm | 25 | 99 | Left Click |
| 42 | 42 | >=15 Cm | 25 | 99 | Right Click |
| 43 | 43 | >=15 Cm | 25 | 99 | Right Click |
| 44 | 44 | >=15 Cm | 25 | 99 | Right Click |
| 45 | 45 | >=15 Cm | 25 | 99 | Right Click |
| 46 | 46 | >=15 Cm | 25 | 99 | Right Click |
| 47 | 47 | >=15 Cm | 25 | 100 | Right Click |
| 48 | 48 | >=15 Cm | 25 | 100 | Right Click |
| 49 | 49 | >=15 Cm | 25 | 100 | Right Click |
| 50 | 50 | >=15 Cm | 25 | 100 | Right Click |
| 51 | 51 | >=15 Cm | 25 | 100 | Right Click |
| 52 | 52 | >=15 Cm | 25 | 100 | Right Click |
| 53 | 53 | >=15 Cm | 25 | 100 | Right Click |
| 54 | 54 | >=15 Cm | 25 | 100 | Right Click |
| 55 | 55 | >=15 Cm | 25 | 100 | Right Click |
| 56 | 56 | >=15 Cm | 25 | 100 | Right Click |
| 57 | 57 | >=15 Cm | 25 | 100 | Right Click |
| 58 | 58 | >=15 Cm | 25 | 100 | Right Click |
| 59 | 59 | >=15 Cm | 24 | 100 | Right Click |
| 60 | 60 | >=15 Cm | 24 | 100 | Right Click |
| 61 | 61 | >=15 Cm | 24 | 100 | Right Click |
| 62 | 62 | >=15 Cm | 24 | 100 | Right Click |
| 63 | 63 | >=15 Cm | 24 | 100 | Right Click |
| 64 | 64 | >=15 Cm | 24 | 100 | Right Click |
| 65 | 65 | >=15 Cm | 24 | 99 | Right Click |
| 66 | 66 | >=30 Cm | 24 | 99 | Cursor Movement |
| 67 | 67 | >=30 Cm | 24 | 99 | Cursor Movement |
| 68 | 68 | >=30 Cm | 24 | 99 | Cursor Movement |
| 69 | 69 | >=30 Cm | 24 | 99 | Cursor Movement |
| 70 | 70 | >=30 Cm | 24 | 99 | Cursor Movement |
| 71 | 71 | >=30 Cm | 24 | 99 | Cursor Movement |
| 72 | 72 | >=30 Cm | 24 | 99 | Cursor Movement |
| 73 | 73 | >=30 Cm | 23 | 99 | Cursor Movement |
| 74 | 74 | >=30 Cm | 23 | 99 | Cursor Movement |
| 75 | 75 | >=30 Cm | 23 | 99 | Cursor Movement |
| 76 | 76 | >=30 Cm | 23 | 99 | Cursor Movement |
| 77 | 77 | >=30 Cm | 23 | 99 | Cursor Movement |
| 78 | 78 | >=30 Cm | 23 | 99 | Cursor Movement |
| 79 | 79 | >=30 Cm | 23 | 99 | Cursor Movement |
| 80 | 80 | >=30 Cm | 23 | 99 | Cursor Movement |
| 81 | 81 | >=30 Cm | 23 | 99 | Cursor Movement |
| 82 | 82 | >=30 Cm | 23 | 99 | Cursor Movement |
| 83 | 83 | >=30 Cm | 23 | 99 | Cursor Movement |
| 84 | 84 | >=30 Cm | 23 | 99 | Cursor Movement |
| 85 | 85 | >=30 Cm | 23 | 99 | Cursor Movement |
| 86 | 86 | >=30 Cm | 23 | 99 | Cursor Movement |
| 87 | 87 | >=30 Cm | 23 | 98 | Cursor Movement |
| 88 | 88 | >=30 Cm | 23 | 98 | Cursor Movement |
| 89 | 89 | >=30 Cm | 23 | 98 | Cursor Movement |
| 90 | 90 | >=30 Cm | 23 | 98 | Cursor Movement |
| 91 | 91 | >=30 Cm | 24 | 98 | Cursor Movement |
| 92 | 92 | >=30 Cm | 24 | 98 | Cursor Movement |
| 93 | 93 | >=30 Cm | 24 | 98 | Left Click |
| 94 | 94 | >=30 Cm | 24 | 98 | Left Click |
| 95 | 95 | >=30 Cm | 24 | 98 | Left Click |
| 96 | 96 | >=30 Cm | 24 | 98 | Left Click |
| 97 | 97 | >=30 Cm | 24 | 98 | Left Click |
| 98 | 98 | >=30 Cm | 24 | 98 | Left Click |
| 99 | 99 | >=30 Cm | 24 | 98 | Left Click |
| 100 | 100 | >=30 Cm | 24 | 98 | Left Click |
| 101 | 101 | >=30 Cm | 24 | 98 | Left Click |
| 102 | 102 | >=30 Cm | 24 | 98 | Left Click |
| 103 | 103 | >=30 Cm | 24 | 98 | Left Click |
| 104 | 104 | >=30 Cm | 24 | 98 | Left Click |
| 105 | 105 | >=30 Cm | 24 | 98 | Left Click |
| 106 | 106 | >=30 Cm | 23 | 98 | Left Click |
| 107 | 107 | >=30 Cm | 23 | 98 | Left Click |
| 108 | 108 | >=30 Cm | 23 | 98 | Left Click |
| 109 | 109 | >=30 Cm | 23 | 98 | Left Click |
| 110 | 110 | >=30 Cm | 23 | 98 | Left Click |
| 111 | 111 | >=30 Cm | 23 | 98 | Left Click |
| 112 | 112 | >=30 Cm | 23 | 98 | Left Click |
| 113 | 113 | >=30 Cm | 23 | 98 | Right Click |
| 114 | 114 | >=30 Cm | 23 | 98 | Right Click |
| 115 | 115 | >=30 Cm | 23 | 98 | Right Click |
| 116 | 116 | >=30 Cm | 23 | 99 | Right Click |
| 117 | 117 | >=30 Cm | 23 | 99 | Right Click |
| 118 | 118 | >=30 Cm | 23 | 99 | Right Click |
| 119 | 119 | >=30 Cm | 23 | 99 | Right Click |
| 120 | 120 | >=30 Cm | 23 | 99 | Right Click |
| 121 | 121 | >=30 Cm | 23 | 99 | Right Click |
| 122 | 122 | >=30 Cm | 23 | 99 | Right Click |
| 123 | 123 | >=30 Cm | 23 | 99 | Right Click |
| 124 | 124 | >=30 Cm | 23 | 99 | Right Click |
| 125 | 125 | >=30 Cm | 23 | 99 | Right Click |
| 126 | 126 | >=30 Cm | 23 | 99 | Right Click |
| 127 | 127 | >=30 Cm | 23 | 99 | Right Click |
| 128 | 128 | >=30 Cm | 23 | 99 | Right Click |
| 129 | 129 | >=30 Cm | 23 | 99 | Right Click |
| 130 | 130 | >=30 Cm | 23 | 99 | Right Click |
| 131 | 131 | >=30 Cm | 23 | 99 | Right Click |
| 132 | 132 | >=30 Cm | 23 | 99 | Right Click |
| 133 | 133 | >=45 Cm | 23 | 99 | Cursor Movement |
| 134 | 134 | >=45 Cm | 23 | 99 | Cursor Movement |
| 135 | 135 | >=45 Cm | 23 | 99 | Cursor Movement |
| 136 | 136 | >=45 Cm | 23 | 99 | Cursor Movement |
| 137 | 137 | >=45 Cm | 23 | 99 | Cursor Movement |
| 138 | 138 | >=45 Cm | 23 | 99 | Cursor Movement |
| 139 | 139 | >=45 Cm | 23 | 99 | Cursor Movement |
| 140 | 140 | >=45 Cm | 23 | 99 | Cursor Movement |
| 141 | 141 | >=45 Cm | 23 | 99 | Cursor Movement |
| 142 | 142 | >=45 Cm | 23 | 97 | Cursor Movement |
| 143 | 143 | >=45 Cm | 23 | 97 | Cursor Movement |
| 144 | 144 | >=45 Cm | 23 | 97 | Cursor Movement |
| 145 | 145 | >=45 Cm | 23 | 97 | Cursor Movement |
| 146 | 146 | >=45 Cm | 23 | 97 | Cursor Movement |
| 147 | 147 | >=45 Cm | 23 | 97 | Cursor Movement |
| 148 | 148 | >=45 Cm | 23 | 97 | Cursor Movement |
| 149 | 149 | >=45 Cm | 23 | 97 | Cursor Movement |
| 150 | 150 | >=45 Cm | 23 | 97 | Cursor Movement |
| 151 | 151 | >=45 Cm | 23 | 97 | Cursor Movement |
| 152 | 152 | >=45 Cm | 23 | 97 | Cursor Movement |
| 153 | 153 | >=45 Cm | 23 | 97 | Cursor Movement |
| 154 | 154 | >=45 Cm | 23 | 97 | Cursor Movement |
| 155 | 155 | >=45 Cm | 23 | 97 | Cursor Movement |
| 156 | 156 | >=45 Cm | 23 | 97 | Cursor Movement |
| 157 | 157 | >=45 Cm | 23 | 97 | Left Click |
| 158 | 158 | >=45 Cm | 23 | 97 | Left Click |
| 159 | 159 | >=45 Cm | 23 | 97 | Left Click |
| 160 | 160 | >=45 Cm | 23 | 97 | Left Click |
| 161 | 161 | >=45 Cm | 23 | 97 | Left Click |
| 162 | 162 | >=45 Cm | 23 | 97 | Left Click |
| 163 | 163 | >=45 Cm | 23 | 97 | Left Click |
| 164 | 164 | >=45 Cm | 23 | 97 | Left Click |
| 165 | 165 | >=45 Cm | 23 | 97 | Left Click |
| 166 | 166 | >=45 Cm | 23 | 97 | Left Click |
| 167 | 167 | >=45 Cm | 23 | 97 | Left Click |
| 168 | 168 | >=45 Cm | 23 | 97 | Left Click |
| 169 | 169 | >=45 Cm | 23 | 97 | Left Click |
| 170 | 170 | >=45 Cm | 23 | 97 | Left Click |
| 171 | 171 | >=45 Cm | 23 | 97 | Left Click |
| 172 | 172 | >=45 Cm | 23 | 97 | Left Click |
| 173 | 173 | >=45 Cm | 23 | 97 | Left Click |
| 174 | 174 | >=45 Cm | 23 | 97 | Left Click |
| 175 | 175 | >=45 Cm | 23 | 96 | Left Click |
| 176 | 176 | >=45 Cm | 23 | 97 | Left Click |
| 177 | 177 | >=45 Cm | 23 | 98 | Left Click |
| 178 | 178 | >=45 Cm | 23 | 99 | Left Click |
| 179 | 179 | >=45 Cm | 23 | 95 | Left Click |
| 180 | 180 | >=45 Cm | 23 | 95 | Left Click |
| 181 | 181 | >=45 Cm | 23 | 95 | Left Click |
| 182 | 182 | >=45 Cm | 23 | 95 | Right Click |
| 183 | 183 | >=45 Cm | 23 | 95 | Right Click |
| 184 | 184 | >=45 Cm | 23 | 95 | Right Click |
| 185 | 185 | >=45 Cm | 22 | 95 | Right Click |
| 186 | 186 | >=45 Cm | 23 | 95 | Right Click |
| 187 | 187 | >=45 Cm | 24 | 95 | Right Click |
| 188 | 188 | >=45 Cm | 22 | 95 | Right Click |
| 189 | 189 | >=45 Cm | 22 | 94 | Right Click |
| 190 | 190 | >=45 Cm | 22 | 94 | Right Click |
| 191 | 191 | >=45 Cm | 22 | 94 | Right Click |
| 192 | 192 | >=45 Cm | 22 | 94 | Right Click |
| 193 | 193 | >=45 Cm | 22 | 94 | Right Click |
| 194 | 194 | >=45 Cm | 22 | 94 | Right Click |
| 195 | 195 | >=45 Cm | 22 | 94 | Right Click |
| 196 | 196 | >=45 Cm | 22 | 94 | Right Click |
| 197 | 197 | >=45 Cm | 22 | 94 | Right Click |
| 198 | 198 | >=45 Cm | 22 | 94 | Right Click |
| 199 | 199 | >=45 Cm | 22 | 94 | Right Click |
| 200 | 200 | >=45 Cm | 22 | 94 | Right Click |

The results shows that there is no significant difference between physical mouse and virtual mouse operation. In our proposed system we can perform mouse operations with 100 percent accuracy .The results doesn’t show any difference between fainted light and normal light. We shall first discuss the fingertip tracking preference in different lighting condition, noisy background and in distant tracking conditions. In this analysis, we record rapid gestures performance with their percentage of accuracy. In figure 9 the line graph shows the accuracy level of different performed actions in our virtual mouse system. In Figure11, our experimental results compared to another approaches using gestures for virtual mouse systems. However, in our proposed system we can perform mouse operation with high accuracy.

Fig 11: Graph of Accuracy for different performed actions

In Figure12, our experimental results compared to another approaches using gestures for virtual mouse systems. But in our proposed system we can perform mouse operation with high accuracy.

# Fig 12: Accuracy graph for different available mouse syste

## 5. CONCLUSION

In conclusion, it’s no surprised that the physical mouse will be replaced by a virtual non-physical mouse in the Human-Computer Interactions (HCI), where every mouse movements can be executed with a swift of your fingers everywhere and anytime without any environmental restrictions. This project had develop a colour recognition program with the purpose of replacing the generic physical mouse without sacrificing the accuracy and efficiency, it is able to recognize colour movements, combinations, and translate them into actual mouse functions. Due to accuracy and efficiency plays an important role in making the program as useful as an actual physical mouse, a few techniques had to be implemented.

First and foremost, the coordinates of the colours that are in charge of handling the cursor movements are averaged based on a collections of coordinates, the purpose of this technique is to reduce and stabilize the sensitivity of cursor movements, as slight movement might lead to unwanted cursor movements. Other than that, several colour combinations were implemented with the addition of distance calculations between two colours within the combination, as different distance triggers different mouse functions. The purpose of this implementation is to promote convenience in controlling the program without much of a hassle. Therefore, actual mouse functions can be triggered accurately with minimum trial and errors.

Furthermore, to promote efficient and flexible tracking of colours, calibrations phase was implemented, this allows the users to choose their choices of colours on different mouse functions, as long the selected colours doesn't fall within the same/similar RGB values (e.g. blue and sky-blue). Other than that, adaptive calibrations were also implemented as well, it is basically allows the program to save different set of HSV values from different angles where it will be used during the recognition phase.

In Overall, the modern technologies have come a long way in making the society life better in terms of productivity and lifestyle, not the other way around. Therefore, societies must not mingle on the past technologies while reluctant on relating work.

## 6. LIMITATION :

In this project, there are several existing problems that may hinder the results of colour recognitions. One of the problems is the environmental factor during the recognition phase takes place. The recognition process are highly sensitive on the intensity of brightness, as immense brightness or darkness may cause the targeted colours to be undetected within the captured frames. Besides that, distance is also the one of the problem that may affect the colour recognition results, as the current detection region can support up to 25cm radius, any display of colours exceed the mentioned distance will be considered as a noise and be filtered off.

Furthermore, the performance of the program are highly dependent on the users' hardware, as processor speed and/or resolutions taken from the webcam could have an effect on performance load. Therefore, the slower the processing speed and/or the higher the resolutions, the longer time are required to process a single frame.

## 7. FUTUTRE WORKS :

There are several features and improvements needed in order for the program to be more user friendly, accurate, and flexible in various environments. The following describes the improvements and the features required:

1. Smart Recognition Algorithm

Due to the current recognition process are limited within 35cm radius, an adaptive zoom-in/out functions are required to improve the covered distance, where it can automatically adjust the focus rate based on the distance between the users and the webcam.

1. Better Performance

The response time are heavily rely on the hardware of the machine, this includes the processing speed of the processor, the size of the available RAM, and the available features of webcam. Therefore, the program may have better performance when it's running on a decent machines with a webcam that performs better in different types of lightings.

**8. APPLICATIONS :**

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

Major applications:

1. The proposed model has a greater accuracy of 99% which is far greater than the that of other proposed models for virtual mouse, and it has many applications.
2. Amidst the COVID-19 situation, it is not safe to use the devices by touching them because it may result in a possible situation of spread of the virus by touching the devices, so the proposed AI virtual mouse can be used to control the PC mouse functions without using the physical mouse(iii)The system can be used to control robots and automation systems without the usage of devices(iv)2D and 3D images can be drawn using the AI virtual system using the hand gestures(v)AI virtual mouse can be used to play virtual reality- and augmented reality-based games without the wireless or wired mouse devices(vi)Persons with problems in their hands can use this system to control the mouse functions in the computer(vii)In the field of robotics, the proposed system like HCI can be used for controlling robots(viii)In designing and architecture, the proposed system can be used for designing virtually for prototyping

**9. REFERENCES :**

1. Virtual mouse: a low cost proximity-based gestural pointing device

S. K. Tang, W. C. Tseng, W. W. Luo, K. C. Chiu, S. T. Lin and Y. P. Liu

International Conference on Human-Computer Interaction 2011

1. Banerjee, A., Ghosh, A., Bharadwaj, K., &Saikia, H. (2014). Mouse control using a web camera based on colour detection. *arXiv preprint arXiv:1403.4722*.
2. Li Wensheng, ; Deng Chunjian, ; Lv Yi, (2010). *[IEEE 2010 International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA) - Chengdu, China (2010.12.17-2010.12.19)] The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding - Implementation of virtual mouse based on machine vision. , (), 367–371.*doi:10.1109/icacia.2010.5709921
3. Henzen, Alexandre; Nohama, Percy (2016). *[IEEE 2016 Future Technologies Conference (FTC) - San Francisco, CA, USA (2016.12.6-2016.12.7)] 2016 Future Technologies Conference (FTC) - Adaptable virtual keyboard and mouse for people with special needs. , (), 1357–1360.*doi:10.1109/FTC.2016.7821782
4. Vantukala VishnuTeja Reddy;Thumma Dhyanchand;Galla Vamsi Krishna;Satish Maheshwaram; (2020). *Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition . 2020 IEEE-HYDCON, (), –.*doi:10.1109/HYDCON48903.2020.9242677
5. Chu-Feng,L. (2008).Portable Vision-Based HCI. [online] Available at: <http://www.csie.ntu.edu.tw/~p93007/projects/vision/vision_hci_p93922007.pdf> [Accessed 25 Aug. 2015].
6. Kumar N, M. (2011). *Manual Testing: Agile software development*. [online] Manojforqa.blogspot.com. Available at: <http://manojforqa.blogspot.com/2011/09/agile-software-development.html> [Accessed 27 Aug. 2015].
7. Sekeroglu, K. (2010). Virtual Mouse Using a Webcam. [online] Available at: [http://www.ece.lsu.edu/ipl/SampleStudentProjects/ProjectKazim/Virtual%20Mouse%](http://www.ece.lsu.edu/ipl/SampleStudentProjects/ProjectKazim/Virtual%20Mouse%25)20Using%20a%20Webcam\_Kazim\_Sekeroglu.pdf [Accessed 29 Aug. 2015].
8. Tutorialspoint.com, (n.d.). *SDLC - Agile Model*. [online] Available at: <http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm>[Accessed 27 Aug. 2015].
9. Tabernae.com, (n.d.). *Software Life Cycle | Web Development Outsourcing| IT Offshore Outsourcing*. [online] Available a[t: http://www.tabernae.com/process.aspx](http://www.tabernae.com/process.aspx) [Accessed 28 Aug. 2015].
10. Zhengyou, Z., Ying, W. and Shafer, S. (2001). Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. [online] Available at

<http://research.microsoft.com/en-us/um/people/zhang/Papers/PUI2001->VisualPanel.pdf [Accessed 25 Aug. 2015].

1. Vantukala VishnuTeja Reddy;Thumma Dhyanchand;Galla Vamsi Krishna;Satish Maheshwaram; (2020). *Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition . 2020 IEEE-HYDCON, (), –.*doi:10.1109/HYDCON48903.2020.9242677

14.Chowdhury, S. R., Pathak, S., & Praveena, M. D. A. (2020). *Gesture Recognition Based Virtual Mouse and Keyboard. 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184).* doi:10.1109/icoei48184.2020.9143016

15. Jeon, Changhyun; Kwon, Oh-Jin; Shin, Dongil; Shin, Dongkyoo (2017). *Hand-Mouse Interface Using Virtual Monitor Concept for Natural Interaction. IEEE Access, 5(), 25181–25188.*doi:10.1109/ACCESS.2017.2768405

Liang Chen, ; Brown, C.G. (2005). *[IEEE 2005 IEEE International Symposium on Circuits and Systems - Kobe, Japan (23-26 May 2005)] 2005 IEEE International Symposium on Circuits and Systems - A 3D Mouse for Interacting with Virtual Objects. , (), 5011–5014.*doi:10.1109/iscas.2005.1465759