

AI VIRTUAL MOUSE AND KEYBOARD USING PYTHON AND OPENCV TO ABATE THE SPREAD OF COVID-19

*A project report submitted in partial fulfillment of the requirements for the award
of the degree of*

**BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING**

Submitted by

G. Sirisha-318126512077

M.Chandra Sekhar-318126512088

U.Mahesh-318126512110

A.Trivedh-318126512061

Under the guidance of

Mrs. Ch. PADMA SREE

(Assistant Professor)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)**

*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)
Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)*

(2021-2022)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)**

*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA &
NAAC with 'A' Grade)*

Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)



ANITS

CERTIFICATE

This is to certify that the project report entitled “AI VIRTUAL MOUSE AND KEYBOARD USING PYTHON AND OPENCV TO ABATE THE SPREAD OF COVID-19” submitted by G.Sirisha(318126512077), U.Mahesh(318126512110), M.Chandra sekhar(318126512088), A.Trivedh(318126512061) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics & Communication Engineering of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

*Project Guide
Padmasree
26/05/2022*
Mrs.Ch.Padmasree
M.Tech (Ph.D)
Assistant Professor
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda

[Signature]
Head of the Department

Dr.V.Rajyalakshmi
Professor & HOD
Department of E.C.E
ANITS

Head of the Department
Department of ECE

Anil Neerukonda Institute of Technology & Sciences
Sangivalasa-531 162

Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mrs.Ch.Padma Sree**, M.Tech, (Ph.D) Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa**, for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of the Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

G.Sirisha (318126512077)

U.Mahesh (318126512110)

M.Chandra sekhar (318126512088)

A.Trivedh (318126512061)

ABSTRACT

Mouse and keyboard are one of the brilliant developments of Human-Computer Interaction (HCI). At present, wireless mouse is used which is not completely free from devices. These restrictions can be overcome by utilizing a webcam for performing mouse operations for all intents and purposes using MediaPipe and computer vision. Besides mouse operations, keyboard functions are also included without the employ of physical mouse. The algorithm used in this extend utilizes Machine Learning (ML) algorithm. This paper creates an application to envision the keyboard and operate it virtually using image processing concepts. This virtual keyboard makes use of Artificial Intelligence (AI) so that the users can work at any place same as a keyboard. This helps in getting freed of human reliance on gadgets to monitor the computer which further helps in minimizing COVID-19 spread.

CONTENTS

CHAPTER	PAGE NO.
LIST OF FIGURES	vii
LIST OF TABLES	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1 INTRODUCTION	1
1.1 Project Objective	2
1.2 Project Outline	2
CHAPTER 2 LITERATURE REVIEW	3
CHAPTER 3 THEORETICAL ASPECTS	6
3.1 Human-Computer Interaction Technology	7
3.1.1 Applications of HCI	8
3.2 Computer Vision	9
3.2.1 The Evolution of Computer-Vision	10
3.3 Image Processing	11
3.3.1 Purpose of Image Processing	12
3.3.2 Basic terms of in Image Processing	13
3.3.3 Types of Image Processing	14
3.3.4 Image Processing Techniques	15
3.3.5 Applications of Image Processing	17
3.4 Machine Learning	19
3.4.1 Applications of Machine Learning	20
3.5 Support Vector Machine	21
3.5.1 Working of SVM	24
3.5.2 Applications of SVM	25

CHAPTER 4 SYSTEM DESIGN	27
4.1 Introduction	28
4.2 Algorithm Illustration	29
4.2.1 Landmarking algorithm	29
4.2.2 MediaPipe	31
4.2.3 OpenCV	34
4.3 System Requirements	35
4.3.1 Software Requirements	35
4.3.2 Hardware Requirements	38
4.4 Methodology	39
CHAPTER 5 RESULTS AND DISCUSSIONS	42
5.1 Simulated Results	43
5.2 Performance Analysis	51
5.3 Applications of our proposed model	51
CHAPTER 6 CONCLUSION	53
FUTURE SCOPE	55
PAPER PUBLICATION DETAILS	56
REFERENCES	57

LIST OF FIGURES

Figure No.	Figure Name	Page No.
Fig 3.1	HCI and related research fields	07
Fig 3.2	Image Resizing	16
Fig 3.3	Image Filtering	16
Fig 3.4	Image Segmentation	17
Fig 3.5	Machine Learning Life-Cycle	20
Fig 3.6	Applications of Machine Learning	21
Fig 3.7	SVM classifier working	22
Fig 3.8	Original dataset	24
Fig 3.9	Data with separator added	24
Fig 3.10	Transformed data	24
Fig 4.1	Landmarks of hand	31
Fig 4.2	Flow-chart of MediaPipe	33
Fig 4.3	Flow-chart of real-time AI virtual mouse and keyboard system	41
Fig 5.1	Recognizing hand (No operation)	43
Fig 5.2	Gesture for mouse movement	44
Fig 5.3	Gesture for left click function	44
Fig 5.4	Gesture for right click function	45
Fig 5.5	Gesture for scroll up function	45
Fig 5.6	Gesture for scroll down function	46
Fig 5.7	Gesture for Volume up function	46
Fig 5.8	Gesture for Volume down function	47
Fig 5.9	Gesture for Escape function	47
Fig 5.10	Gesture for typing or clicking a letter on screen from virtual keyboard	48

Fig 5.11	Virtual keyboard	48
Fig 5.12	Graph determining accuracy level of each operation	50
Fig 5.13	Graph for comparison between different models	51

LIST OF TABLES

Table No.	Table Name	Page No.
Table 5.1	Respective tip ids for fingers	49
Table 5.2	Tested Results	49

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
HCI	Human-Computer Interaction
ML	Machine Learning
UI	User Interface
GUI	Graphical User Interface
CPU	Central Processing Unit
GPU	Graphics Processing Unit
CV	Computer Vision
DSP	Digital Signal Processing
SVM	Support Vector Machine
MATLAB	Matrix Laboratory
CT	Computed Tomography
OS	Operating System
RGB	Red-Green-Blue
OCR	Optical Character Appreciation
DNA	Deoxyribonucleic Acid
ATR	Automatic Target Recognition

CHAPTER 1
INTRODUCTION

In general, devices are becoming compact in the form of bluetooth or wireless technologies. This project proposes an AI virtual mouse system that makes use of the hand gestures and hand tip detection for performing mouse functions in the computer using computer vision.

Project Objective

The objective of this project is to provide an alternative for a routine physical mouse so that there will be a less physical contact with mouse. We can perform all the mouse operations and few keyboard operations by just recognizing different hand- gestures through web-cam.

Project Outline

The outline of the project is as follows. In today's world there is lots of development happening in the field of Technology. Today's technology is combined with the technique called Artificial Intelligence. This project is also based on small part of AI. This project presents finger movement gesture detection on our computer's window using camera & handling the whole system by just moving your one finger. Using finger detection methods for instant camera access and user-friendly user interface makes it more easily accessible. This system reduces the use of any physical mouse which saves time and also reduces efforts. AI virtual mouse and keyboard is developed using Python and OpenCV, a computer vision library. The proposed model utilizes MediaPipe package for recognizing the hands and the tip of the fingers, as well as PyAutoGUI and Autopy packages for controlling the system by performing mouse operations like right click, left click, scroll up, scroll down and keyboard operations like escape, volume up, volume down. Outcome of this model demonstrates a high-level accuracy which can function extremely well in real-time applications using only a CPU and no GPU. This system also helps in controlling robots.

In virtual keyboard basically the movement of finger tapping will be captured on the virtual keyboard which will be displayed on the screen while in the mouse finger movement will be captured with the assistance of camera.

CHAPTER 2
LITERATURE REVIEW

There are traditional approaches for virtual keyboard and mouse systems which are usually based on hand gestures. But few are done using deep learning and few using different algorithms. Our literature review focuses on the research works on virtual keyboard and virtual mouse which were published previously.

In 2016, S. Shetty et al. constructed a virtual mouse system using color detection. They used webcam for detecting mouse cursor movement and click events using OpenCV built-in functions. A mouse driver, written in java, is required as well. This system fails to perform well in rough background. P. C.

Shindhhe et al. expanded a method for mouse free cursor control where mouse cursor operations are controlled by using hand fingers. They have collected hand gestures via webcam using color detection principles. The built-in function of image processing toolbox in MATLAB and a mouse driver, written in java are used in this approach. The pointer was not too efficient on the air as the cursor was very sensitive to the motion.

In 2019, K. Hassan et al. presented a system to design and develop a hand gesture based virtual mouse. They captured different gestures via webcam and performed mouse functions according to the gestures. This system achieved 78%-90% accuracy. The system does not work efficiently in the complex or rough background.

In 2021, S. Shriram presented a hand-gesture based virtual mouse system using deep-learning. This system achieved 99% accuracy but didn't develop keyboard and there is less accuracy in right-click function, clicking and dragging operations are less accurate.

In 2010, Y. Adajania et. al developed a Virtual Keyboard Using Shadow Analysis. This system detects keyboard, hands shadow and finger tips using color. This system can analyze 3 frames per second.

In 2011, S. Hernanto et al. built a method for virtual keyboard using webcam. In this approach, two functions are used for finger detection and location. This system used two different webcams which are used to detect skin and location separately. The average time per character of this virtual keyboard is 2.92 milliseconds and the average accuracy of this system is 88.61%.

In 2015, I. Patil et al. constructed a virtual keyboard interaction system using eye gaze and eye blinking. Their system first detects face and then detects eye and nose region to recognize an eye blink. The OpenCV java framework is used in this approach. In 160X120 frame size, this approach achieves 48% accuracy and in 1280X960 frame size, 98% accuracy is achieved.

In 2016, Hubert Cecotti developed a system for disabled people named a multi-modal gaze-controlled virtual keyboard. The virtual keyboard has 8 main commands for menu selection to spell 30 different characters and a delete button to recover from error. They evaluated the performance of the system using the speed and information transfer rate at both the command and application levels.

In 2017, S. Bhuvana et al. constructed a virtual keyboard interaction system using webcam. This system can detect the hand position over the virtual keyboard. This system provides a white paper virtual keyboard image and detects which character is pointed. This approach used built-in function of Image Processing Toolbox in MATLAB.

In 2018, Jagannathan MJ et al. presented finger recognition and gesture based augmented keyboard system. As we can see from the reviewed literature, previous systems include either virtual keyboard or virtual mouse. Those systems can't fully eliminate the need of mouse and keyboard completely. This work aims to build an interactive computer system which can be operated without any physical mouse and keyboard by just recognizing hand-gestures with high accuracy.

CHAPTER 3
THEORETICAL ASPECTS

Human-Computer Interaction Technology

HCI is a multidisciplinary field of study focusing on the design of computer technology and, in particular, the interaction between humans (the users) and computers. While initially concerned with computers, HCI has since expanded to cover almost all forms of information technology design. HCI soon became the subject of intense academic investigation. Those who studied and worked in HCI saw it as a crucial instrument to popularize the idea that the interaction between a computer and the user should resemble a human-to-human, open-ended dialogue. Iterative design is one of the foundational principles of HCI. Once a designer has gained an understanding of his or her target audience, their tasks, and the empirical measurements surrounding an interaction, designers follow several iterative design steps: design the user interface; conduct user test in; analyze the results of testing; repeat. The iterative design process is repeated until a user-friendly interface is created.

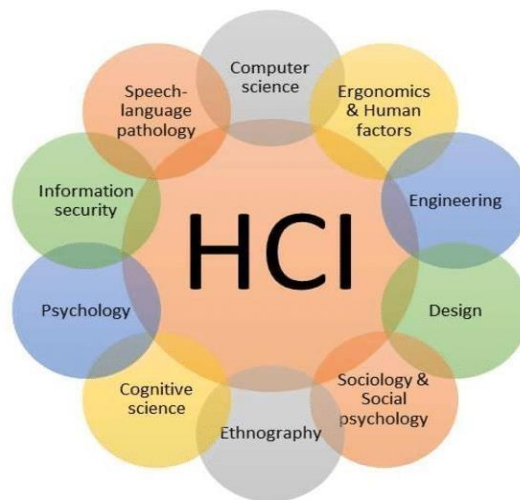


Fig 3.1: HCI and related research fields

The interaction between a machine and a human can be facilitated in multiple ways. Generally, it's possible to utilize one or more human senses to form the basis of a UI, such as tactile UI (touch), visual UI (sight), and auditory UI (sound).

HCI practitioners find the optimal combination that fits the purpose of the product. For example, for a mobile app, this might be a combination of visual UI and auditory UI. Mouse and keyboard are one of the brilliant developments of HCI.

Applications of HCI

For example, the sensory perception and interactive input devices include speech recognition, keyboards, and touch-sensitive screens; the output devices include the printers and visual display; wireless devices such as the application of wireless internet; and the virtual reality devices.

There are few devices which are related to HCI. They are

a. Mouse

A computer mouse is a handheld hardware input device that controls a cursor in a GUI for pointing, moving and selecting text, icons, files, and folders on your computer. In addition to these functions, a mouse can also be used to drag-and-drop objects and give you access to the right-click menu. For desktop computers, the mouse is placed on a flat surface (e.g., mouse pad or desk) in front of your computer. The full form of mouse is Manually Operated User Selection Equipment or Mechanically Operated User Signal Engine.

Different types of mouse:

- Wired Mouse
- Bluetooth Mouse
- Trackball Mouse
- Optical Mouse
- Laser Mouse
- Magic Mouse
- USB Mouse
- Vertical Mouse

Functions of mouse:

- Point.
- Select.

- Hover.
- Scroll.
- Drag-and-drop.
- Open & Close Program.

b. Keyboard

A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions. A keyboard is connected to a computer system using a cable or a wireless connection.

Different types of Keyboards:

- Qwerty Keyboards
- Wired Keyboards
- Numeric Keypads
- Ergonomic Keyboards
- Wireless Keyboards
- USB Keyboards
- Bluetooth Keyboards

Computer Vision

One of the most powerful and compelling types of AI is computer vision which you've almost surely experienced in any number of ways without even knowing. Computer vision is the field of computer science that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that worked in limited capacity. Thanks to advances in artificial intelligence and innovations in deep-learning and neural networks, the field has been able to take great leaps in recent years and has been able to surpass humans in some tasks related to detecting and labeling objects.

One of the driving factors behind the growth of computer vision is the amount of data we generate today that is then used to train and make computer vision better.

The Evolution of Computer Vision

Before the advent of deep learning, the tasks that computer vision could perform were very limited and required a lot of manual coding and effort by developers and human operators. For instance, if you wanted to perform facial recognition, you would have to perform the following steps:

Create a database: You had to capture individual images of all the subjects you wanted to track in a specific format.

Annotate images: Then for every individual image, you would have to enter several key data points, such as distance between the eyes, the width of nose bridge, distance between upper-lip and nose, and dozens of other measurements that define the unique characteristics of each person.

Capture new images: Next, you would have to capture new images, whether from photographs or video content. And then you had to go through the measurement process again, marking the key points on the image.

After all this manual work, the application would finally be able to compare the measurements in the new image with the ones stored in its database and tell you whether it corresponded with any of the profiles it was tracking. Machine learning provided a different approach to solving computer vision problems.

With machine learning, developers no longer needed to manually code every single rule into their vision applications. Instead they programmed “features,” smaller applications that could detect specific patterns in images. They then used a statistical learning algorithm such as linear regression, logistic regression, decision trees or support vector machines (SVM) to detect patterns and classify images and detect objects in them.

Image Processing

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Image processing is one form of signal processing in which the input is a photograph or video frame; the output may be either an image or a set of characteristics or parameters related to the image. An image contains sub-images sometimes referred as regions-of-interest, or simply regions this implies that images contain collections of objects each of which can be the basis for a region. Thus, we have chosen image processing for identifying the defects on the surface of the Rexene, where the defective part will be the area of interest.

In Image science, Image processing is any form of for which the input is an image, such as a photograph or video frame; the output of image processing may be either image or a set of characteristics or parameters related to the image. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. The acquisition of images (producing the input image in the first place) is referred to imaging.

Image processing is any form of signal processing for which the input is an, such as a photograph or video frame, the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most of the image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. Thus, we have chosen image processing for identifying the defects on the surface of the Ceramic stile, where the defective part will be the area of interest.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps. They are, importing the image with optical scanner or by digital photography. Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are

not to human eyes like satellite photographs. Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image Processing

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

Basic terms in Image Processing

Digital image Processing

Digital Image Processing deals with manipulation of digital images through a digital computer. Digital Image Processing focuses on developing a computer system which is able to perform processing on an image. The input of that system is a digital image and the system process that image using efficient algorithms, and gives an image as an output.

Processing Images

Image processing has been developed in response to three major problems concerned with pictures, picture digitization, printing and storage of pictures. Picture segmentation and description as early-stage machine vision. The most requirements for image processing of images are that the images be available in digitized form, that is, arrays of finite length binary words. For digitization, the given image is sampled on a discrete grid and each sample or pixel is quantized using a finite number of bits. The digitized image is processed by a computer. To display a digital image, it is first converted into analog signal, which is scanned onto a display.

Pixel

Pixel is the smallest element of an image. The value of a pixel at any point corresponds to the intensity of the light photons striking at that point. Each pixel stores a value proportional to the light intensity at that particular location.

Calculation of Total Number of Pixels:

We have defined an image as a two-dimensional signal or matrix. Then in that case the number of pixels would be equal to the number of rows multiply with number of columns. This can be mathematically represented as below (or) we can say that the number of (x, y) coordinate pairs is equal to the total number of pixels.

Total number of pixels = (number of rows) x (number of columns).

Resolution

The term resolution refers to the total number of count of pixels in a digital image. For example, if an image has M rows and N columns, then its resolution can be defined as M x N. If we define resolution as the total number of pixels, then pixel resolution can be defined with set of two numbers. The 1st number the pixels across columns, and the 2nd number is the pixels across its rows. We can say that the higher is the pixel resolution and the higher is the quality of the image. Size of an image = (pixel resolution) X (bits per pixel).

Types of Image Processing

The two types of image processing used are analog image processing and digital image processing.

1. Analog image processing

In electrical engineering and computer science, analog image processing is any image processing task conducted on two-dimensional analog signals by analog means (as opposed to digital image processing). Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using the visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

2. Digital image processing

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre-processing, enhancement and display, information extraction.

In this case, digital computers are used to process the image. The image will be converted into the digital form using a scanner–digitizer and then process it. It is defined as the subjecting numerical representation of objects to a series of

operations in order to obtain the desired result. It starts with one image and produces a modified version of the image. It is therefore an image that takes one image into another.

The term image processing generally refers to processing of a two-dimensional picture by a digital computer, in the broader context; it implies digital processing of a two-dimensional data. A digital image is an array of real numbers represented by a finite number of bits. The principle disadvantage of digital image processing is its versatility, repeatability and the preservation of original data precision.

Image Processing Techniques

Digital image processing deals with manipulation and analysis of images by using computer algorithm, so as to improve pictorial information for better understanding and clarity. This area is characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. Image processing involves the manipulation of images to extract information to emphasize or de-emphasize certain aspects of the information, contained in the image or perform image analysis to extract hidden information. The Computer Vision System aims at recognizing objects of interest from given images and helps in developing the machine, that can perform visual function parallel to human vision. Computer Vision System consists of filtering, coding, enhancement, restoration, feature extraction, analysis and recognition of objects from image. Processing of an image comprises of improvement in its appearance and effective representation of input image suitable for required application.

1. Image re-sizing

Re-sizing of an image is performed by the process of the interpolation. It is a process which re-samples the image to determine values between defined pixels. Thus, resized image contains more or less pixels than that of original image. The intensity values of additional pixels are obtained through interpolation if the resolution of the image is increased.

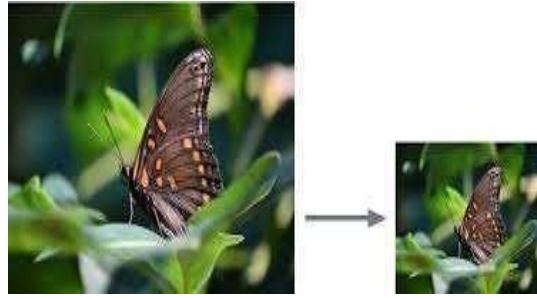


Fig. 3.2 Image resizing

2. Image filtering

Uncertainties are introduced into the image such as random image noise, partial volume effects and intensity non-uniformity artifact (INU), due to the movement of the camera. This results in smooth and slowly varying change in image pixel values and lead to information loss, SNR gain and degradation of edge and finer details of image. Spatial filters are used for noise reduction. These filters may be linear or non-linear filters.



Fig. 3.3 Image filtering

3. Image segmentation

Depending on type of input image samples, segmentation can be classified as gray scale single image segmentation and Histogram based segmentation. Here the image is converted to digital form. After converting the image into bit information, processing is performed. This processing technique may be, image enhancement, image restoration, image compression, and image segmentation. As far as our project is concerned, we used the image segmentation techniques. Image segmentation is the process of dividing or partitioning an image into multiple parts.

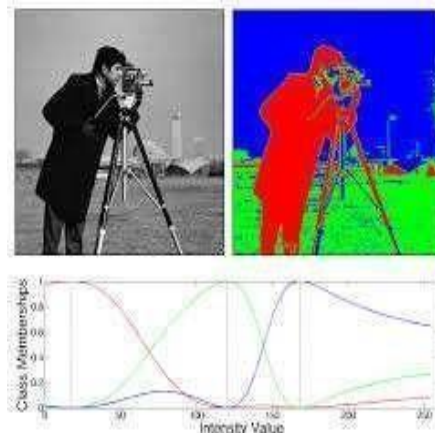


Fig. 3.4 Image segmentation

Applications of Image Processing

1. Character recognition

Optical Character appreciation, usually abbreviated to OCR, is the mechanical or electronic alteration of scanned or photo images of typewritten or printed text into machine-encoded i.e., computer-readable text. It is generally used as an appearance of records access from a little kind of original data source, whether papers, invoice, bank statement, receipts, business cards, a number of printed records or mail. It is an ordinary technique of digitizing printed manuscripts such that they can be by electronic means edited, searched, store more closely used in machine processes such as machine translation and displayed online, text-to-speech, key data extraction and text mining. OCR is a meadow of research in intelligence, pattern and computer vision. Early versions required to be automated with images of each character, and functioned none font at a time. "Intelligent" structures with a great degree of gratitude accuracy for most fonts are now regular. Some marketable methods are skilled of duplicating formatted output that very much resembles the original scanned sheet including columns, images and other non-textual components.

2. Signature verification

A digital signature is a mathematical scheme for representing the legitimacy of a digital communication. A legal digital signature affords a receiver reason to consider that the message was created by a recognized sender, such that the sender cannot reject having sent the message with non-repudiation and authentication and

the message was not changed in transfer. Digital signatures are commonly used for software allocation, financial communication, and in further cases where it is vital to detect imitation or tampering.

3. Bio-metrics

Biometrics (or biometric verification) refers to the automatic identification of humans by their behaviors or characteristics. Biometrics is recycled in computer science as a type of identification and access control. It is also used to recognize individuals in groups that are under surveillance. Biometric identifiers are the exceptional, assessable characteristics used to label and describe individuals, examples include fingerprint, face recognition, Palm print, DNA, hand geometry, iris recognition, retina and odor/smell.

4. Automatic target recognition

Automatic target recognition (ATR) is the skill for an algorithm or device to distinguish objects or targets stand on data gained from sensors. The function of regular target recognition technology is a serious element of robotic warfare. ATR machines are used in unmanned aerial vehicles and cruise missiles. Electric affords an ATRU (Automatic Target Recognition Unit) to the Land Attack Missile of Standoff, which processes post-launch and pre-launch aiming data, allows high quickness in video comparison, and permits the SLAM-ER i.e., Standoff Land Attack Missile-Expanded Response, "Fire-and-forget" missile. The fundamental version of an ATR system is the IFF transponder. Other applications of ATR include a proposed security system that uses active UWB radar signals to recognize objects or humans that have dropped onto channel tracks of rail. It is also possible to detect the damaged infrastructures caused by the earthquakes using satellite.

Machine learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning life cycle:

Machine learning has given the computer systems the abilities to automatically learn without being explicitly programmed. So, it can be described using the life cycle of machine learning. Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project.

Machine learning life cycle involves seven major steps, which are given below:

- Data preparation
- Data Wrangling

- Analyse Data
- Train the model
- Test the model
- Deployment

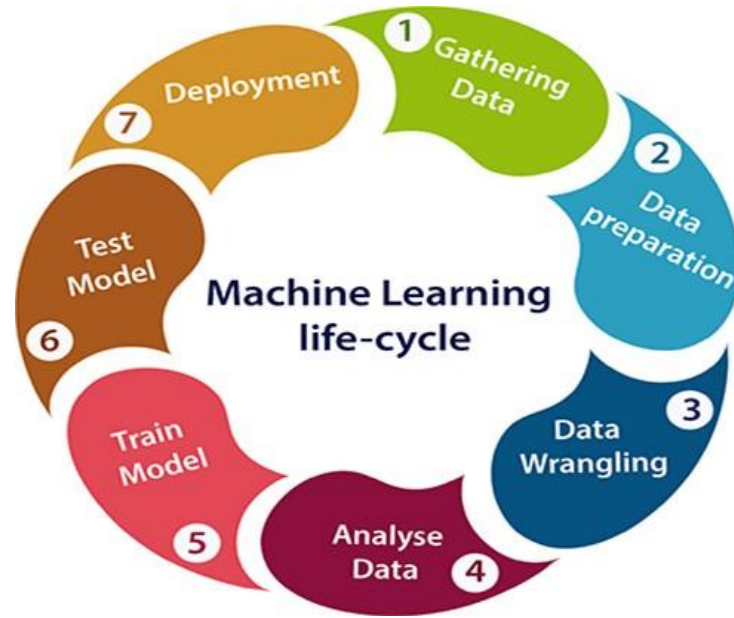


Fig 3.5: Machine Learning Life-Cycle

Applications of Machine Learning

Machine learning is a buzz word for today's technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc.

Below are some most trending real-world applications of Machine Learning:

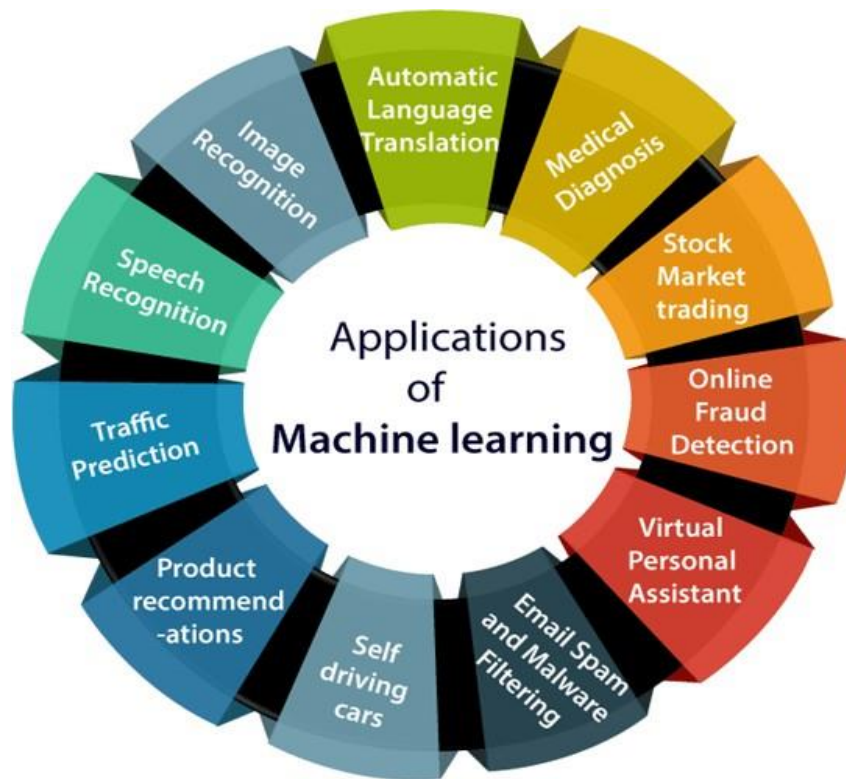


Fig 3.6: Applications of Machine Learning

Support Vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues, SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space so as to maximize the width of the gap between the two categories. New examples are then mapped in to that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform non-linear classification using what is called the kernel trick, implicitly mapping the inputs into high-dimensional feature spaces. When data are unlabeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support-vector machines, a data point is viewed as p -dimensional vector, and we want to know whether we can separate such points with a dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So, we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum-margin classifier; or equivalently, the perceptron of optimal stability.

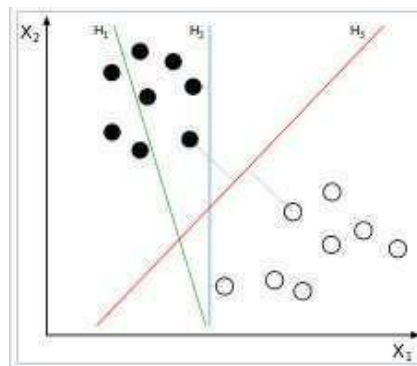


Fig. 3.7: SVM classifier working

H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximal margin.

More formally, a support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outlier's detection. Intuitively, a good

separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to discriminate are not linearly separable in that space.

For this reason, it was proposed that the original finite-dimensional space be mapped in to a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant, where such a set of vectors is an orthogonal (and thus minimal) set of vectors that defines a hyperplane. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters α of images of feature vectors that occur in the data base. With this choice of a hyperplane, the points in the feature space that are mapped into the hyperplane are defined by a relation.

Working of SVM

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found and then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

For example, consider the following figure, in which the data points fall into two different categories.

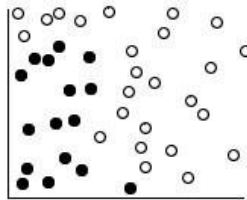


Fig 3.8: Original dataset

The two categories can be separated with a curve, as shown in the following figure.

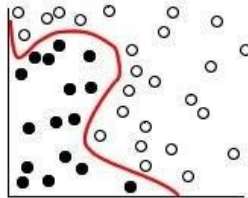


Fig 3.9: Data with separator added

After the transformation, the boundary between the two categories can be defined by a hyperplane, as shown in the following figure.

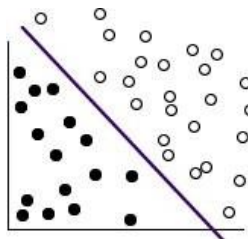


Fig 3.10: Transformed data

The mathematical function used for the transformation is known as the **kernel** function. SVM in IBMSPSS modeller supports the following kernel types:

- Linear
- Polynomial
- Radial basis function (RBF)
- Sigmoid

A linear kernel function is recommended when linear separation of the data is straight forward. In other cases, one of the other functions should be used. You will need to experiment with the different functions to obtain the best model in each case, as they each use different algorithms and parameters.

The followings are important concepts in SVM –

Support Vectors – Data points that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

Hyperplane – As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin – It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

Applications of SVM:

SVMs can be used to solve various real-world problems. SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings. Some methods for shallow semantic parsing are based on support vector machines. Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true for image segmentation systems, including

those using a modified version SVM that uses the privileged approach as suggested by Vapnik.

Classification of satellite data like SAR data using supervised SVM. Hand-written characters can be recognized using SVM. The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with upto 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism for interpretation of SVM models. Support- vector machine weights have also been used to interpret SVM models in the past. Post interpretation of support- vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

CHAPTER 4
SYSTEM DESIGN

Introduction

Brain with the development technologies in the areas of augmented reality and devices that we use in our daily life, these devices are becoming compact in the form of Bluetooth or wireless technologies. This project proposes an AI virtual mouse system that makes use of the hand gestures and hand tip detection for performing mouse functions in the computer using computer vision.

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

In the proposed AI virtual mouse system, the model makes use of the MediaPipe package for the tracking of the hands and for tracking of the tip of the hands, and also, Autopy and PyAutoGUI packages were used for moving around the window screen of the computer for performing functions such as left click, right click, and scrolling functions. The results of the proposed model showed very high accuracy level, and the proposed model can work very well in real-world application with the use of a CPU without the use of a GPU.

Algorithm Illustration

In this project we have used mainly Landmarking Algorithm in which we will be having palm model and hand landmark model. This algorithm uses machine learning algorithm and is present in MediaPipe package. Palm model and hand landmark model are described below.

Land marking algorithm

a) Palm Detection Model

To detect initial hand locations, we designed a single-shot detector model optimized for mobile real-time uses in a manner similar to the face detection mode in MediaPipe Face Mesh. Detecting hands is a decidedly complex task: our lite model and full model have to work across a variety of hand sizes with a large scale span (~20x) relative to the image frame and be able to detect occluded and self- occluded hands. Whereas faces have high contrast patterns, e.g., in the eye and mouth region, the lack of such features in hands makes it comparatively difficult to detect them reliably from their visual features alone. Instead, providing additional context, like arm, body, or person features, aids accurate hand localization.

Our method addresses the above challenges using different strategies. First, we train a palm detector instead of a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. In addition, as palms are smaller objects, the non-maximum suppression algorithm works well even for two-hand self-occlusion cases, like handshakes. Moreover, palms can be modeled using square bounding boxes (anchors in ML terminology) ignoring other aspect ratios, and therefore reducing the number of anchors by a factor of 3-5. Second, an encoder-decoder feature extractor is used for bigger scene context awareness even for small objects (similar to the Retina Net approach). Lastly, we minimize the focal loss during training to support a large amount of anchors resulting from the high scale variance.

With the above techniques, we achieve an average precision of 95.7% in palm detection. Using a regular cross entropy loss and no decoder gives a baseline of

just 86.22%.

b) Hand Landmark Model:

After the palm detection over the whole image our subsequent hand landmark model performs precise keypoint localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions.

To obtain ground truth data, we have manually annotated ~30K real-world images with 21 3D coordinates, as shown below (we take Z-value from image depth map, if it exists per corresponding coordinate). To better cover the possible hand poses and provide additional supervision on the nature of hand geometry, we also render a high- quality synthetic hand model over various backgrounds and map it to the corresponding 3D coordinates.

The mechanism of detecting the famous human-made sculptures, buildings, and monuments inside an image is defined as Landmark Detection. You can simply compare it with the famous application of google known as google Landmark Detection, which is used by google maps. Landmarks are favored picture highlights for an assortment of computer vision errands such as picture mensuration, enlistment, camera calibration, movement examination, 3D scene remaking, and protest acknowledgment.

In image classification with localization, we train a neural network to detect objects and then localize by predicting coordinates of the bounding box around it. Landmark detection is a fundamental building block of computer vision applications such as face recognition, pose recognition, emotion recognition, hand recognition, head recognition to put the crown on it, and many more in augmented reality.

In this project we are considering landmarks of hand shown in Fig 4.1. According to this, particular operation will be done when particular finger is raised. This is in-built present in MediaPipe package which is trained already and is present in Python programming language.

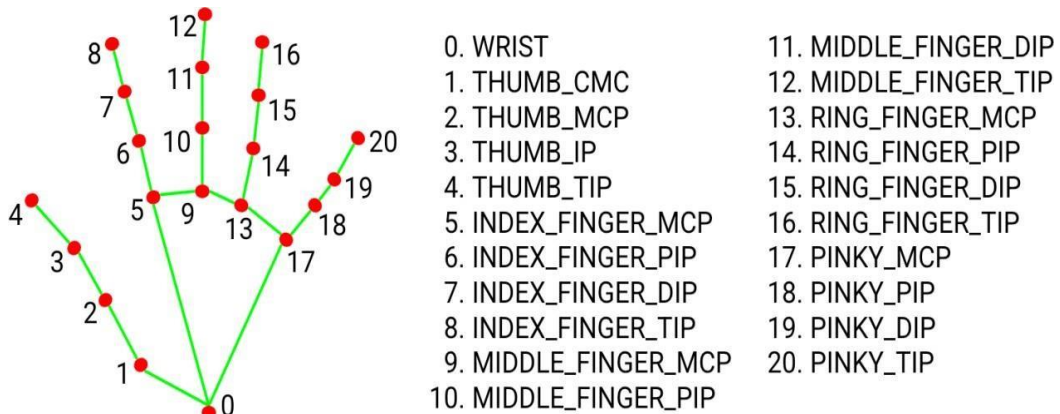


Fig 4.1: Landmarks of a hand

MediaPipe

The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms. For example, it can form the basis for sign language understanding and hand gesture control, and can also enable the overlay of digital content and information on top of the physical world in augmented reality. While coming naturally to people, robust real-time hand perception is a decidedly challenging computer vision task, as hands often occlude themselves or each other (e.g: finger/palm occlusions and handshakes) and lack high contrast patterns.

MediaPipe hand is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame. Whereas current state-of-the-art approaches rely primarily on powerful desktop environments for inference, our method achieves real-time performance on a mobile phone, and even scales to multiple hands. We hope that providing this hand perception functionality to the wider research and development community will result in an emergence of creative use cases, stimulating new applications and new research avenues.

MediaPipe hands utilize an ML pipeline consisting of multiple models working together. ML pipeline is described below.

a) About ML(Machine Learning) Pipeline

MediaPipe hands utilize an ML pipeline consisting of multiple models working together.

A palm detection model that operates on the full image and returns an oriented hand bounding box. A hand landmark model that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand key points.

This strategy is similar to that employed in our MediaPipe Face Mesh solution, which uses a face detector together with a face landmark model.

Providing the accurately cropped hand image to the hand landmark model drastically reduces the need for data augmentation (e.g. rotations, translation and scale) and instead allows the network to dedicate most of its capacity towards coordinate prediction accuracy. In addition, in our pipeline the crops can also be generated based on the hand landmarks identified in the previous frame, and only when the landmark model could no longer identify hand presence is palm detection invoked to re-localize the hand.

The pipeline is implemented as a MediaPipe graph that uses a hand landmark tracking subgraph from the hand landmark module, and renders using a dedicated hand renderer subgraph. The hand landmark tracking subgraph internally uses a hand landmarks graph from the same module and a palm detection subgraph from the palm detection model.

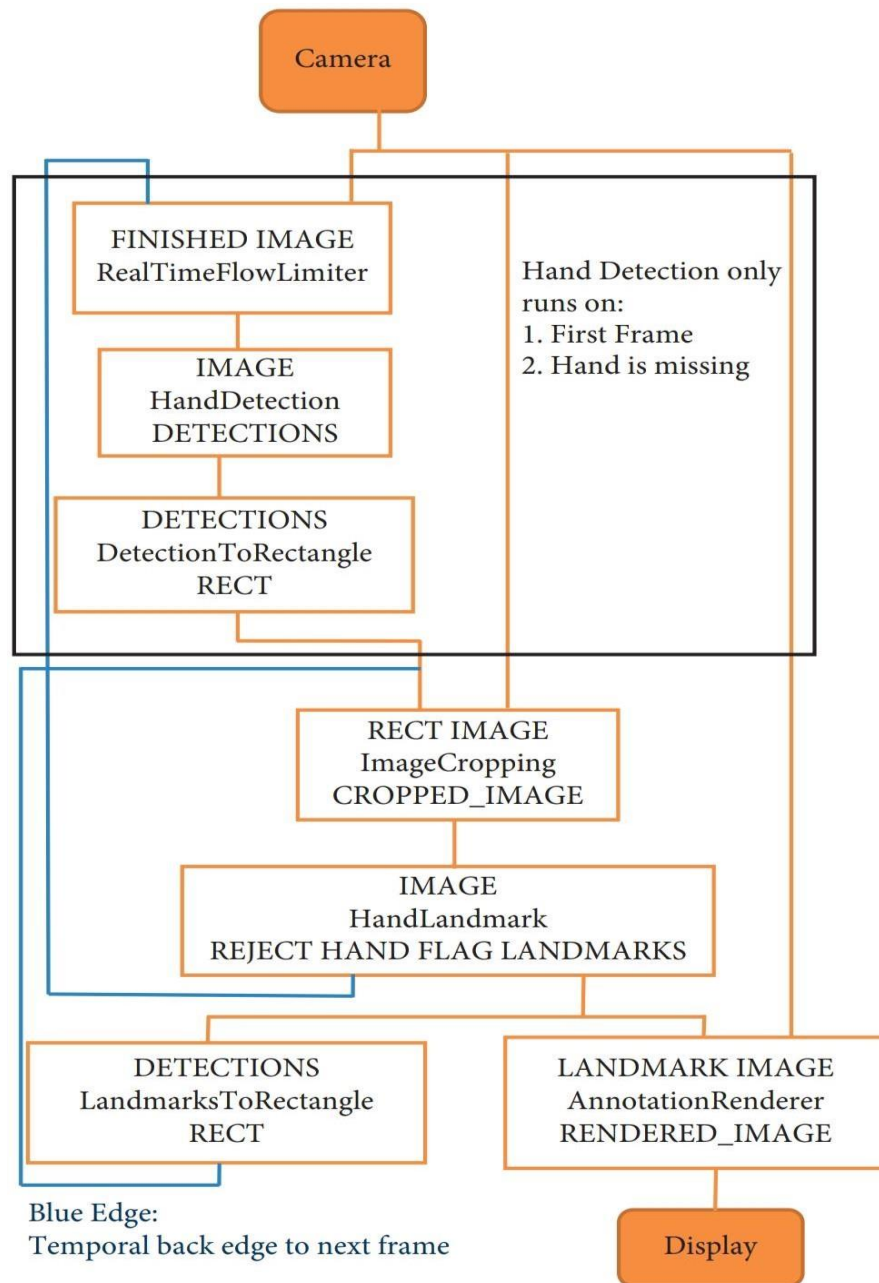


Fig 4.2: Flowchart of MediaPipe

OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it is integrated with various libraries, such as NumPy, Python is capable of processing the OpenCV array structure for analysis. To identify image patterns and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Applications of OpenCV:

There are lots of applications which are solved using OpenCV, some of them are listed below

- Face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- Object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

OpenCV Functionality:

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (object detection, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, video stab)
- Computational photography (photo, video, suppress)
- Machine learning & clustering (ml, flann)

System requirements

Software Requirement:

1. PYTHON (version 8.0 and above):

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is designed to be highly readable. It uses english keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Applications of Python

As mentioned before, Python is one of the most widely used languages over the web. Here are few applications of Python:

- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.

- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

2. OPENCV:

Sample code for web-cam on

```
import cv2

vid = cv2.VideoCapture(0)

while(True):

cv2.imshow('frame', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

vid.release()

cv2.destroyAllWindows()
```

3. MEDIAPIPE:

Hand-Tracking module will be importing from mediapipe package where we will be having the record of landmarks.

4. AUTOPY:

AutoPy is a simple, cross-platform GUI automation library for Python. It includes functions for controlling the keyboard and mouse, finding colors and bitmaps on-screen, and displaying alerts. Currently supported on macOS, windows and X11 with the XTest extension. AutoPy includes a number of functions for controlling the mouse.

5. PYAUTOGUI:

PyAutoGUI is a Python automation library used to click, drag, scroll, move, etc. It can be used to click at an exact position.

PyAutoGUI works across Windows, MacOS X and Linux. Different operations using PyautoGUI:

```
import pyautogui

screen Width,
```

```
screen Height = pyautogui.size()
current MouseX, current MouseY = pyautogui.position()
pyautogui.moveRel(None, 10) # move mouse 10 pixels down
pyautogui.doubleClick()
pyautogui.typewrite('Hello world!', interval=0.25) # type with quarter-second
pause in between each key
pyautogui.press(['left', 'left', 'left', 'left', 'left', 'left'])
pyautogui.keyUp('shift')
pyautogui.hotkey('ctrl', 'c')
```

6. PYCHARM:

This complete project is executed in Pycharm platform which is user-friendly. PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a subscription-funded proprietary license and also an educational version.

PyCharm provides an API so that developers can write their own plugins to extend PyCharm features. Several plugins from other JetBrains IDE also work with PyCharm. There are more than 1000 plugins which are compatible with PyCharm.

Hardware requirement:

1. WEBCAM:

A webcam is a digital video device commonly built into a computer. Its main function is to transmit pictures over the Internet. It is popularly used with instant messaging services and for recording images. A **webcam** is video camera that feeds or streams an image or video in real time to or through a computer network, such as the 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 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, such streams usually use compressed formats. 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) may also be used in its original sense of a video camera connected to the Web continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its web page over the Internet. Some of them, for example, those used as online traffic cameras, are expensive, rugged professional video cameras.

METHODOLOGY

The various functions and conditions used in the system are explained in the flowchart of the real-time AI virtual mouse system shown in Fig 4.3.

1. Camera Used in the AI Virtual Mouse System

The proposed AI virtual mouse system is based on the frames that have been captured by the webcam in a laptop or PC. By using the Python computer vision library OpenCV, the video capture object is created and the web camera will start capturing video. The web camera captures and passes the frames to the AI virtual system.

2. Capturing the Video and Processing

The AI virtual mouse system uses the webcam where each frame is captured till the termination of the program. The video frames are processed from BGR to RGB color space to find the hands in the video frame by frame.

3. (Virtual Screen Matching) Region for moving through the window

The AI virtual mouse system makes use of the transformational algorithm, and it converts the coordinates of fingertip from the webcam screen to the computer window full screen for controlling the mouse. When the hands are detected and when we find which finger is up for performing the specific mouse function, the web-cam captures that particular frame and process further operation.

4. Detecting Which Finger Is Up and Performing the Particular Mouse Function

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

5. Mouse and keyboard Functions Depending on the Hand Gestures and Hand Tip Detection Using Computer Vision

1. If index finger with tip id 1 is up then the mouse is moved around the window of the computer by using AutoPy package.
2. If thumb finger with tip id 0 is up then the Left-click operation is performed by using PyautoGUI package.

3. If both index finger with tip id 1 and middle finger with tip id 2 are up then the Right-click operation is performed by using PyautoGUI package.
4. If both thumb finger with tip id 0 and index finger with tip id 1 are up then the scroll-up operation is performed by using PyautoGUI package.
5. If middle finger with tip id 2, ring finger with tip id 3 and little finger with tip id 4 are up then the scroll-down operation is performed by using PyautoGUI package.
6. If thumb finger with tip id 0, index finger with tip id 1 middle finger with tip id 2 are up then the volume-up operation is performed by using PyautoGUI package.
7. If ring finger with tip id 3 and little finger with tip id 4 are up then the volume-down operation is performed by using PyautoGUI package.
8. If no finger is up then the escape operation is performed by using PyautoGUI package.
9. Clicking on a letter or typing using a virtual keyboard is done if both index finger with tip id 1 and middle finger with tip id 2 are up and distance between them is less than 20.

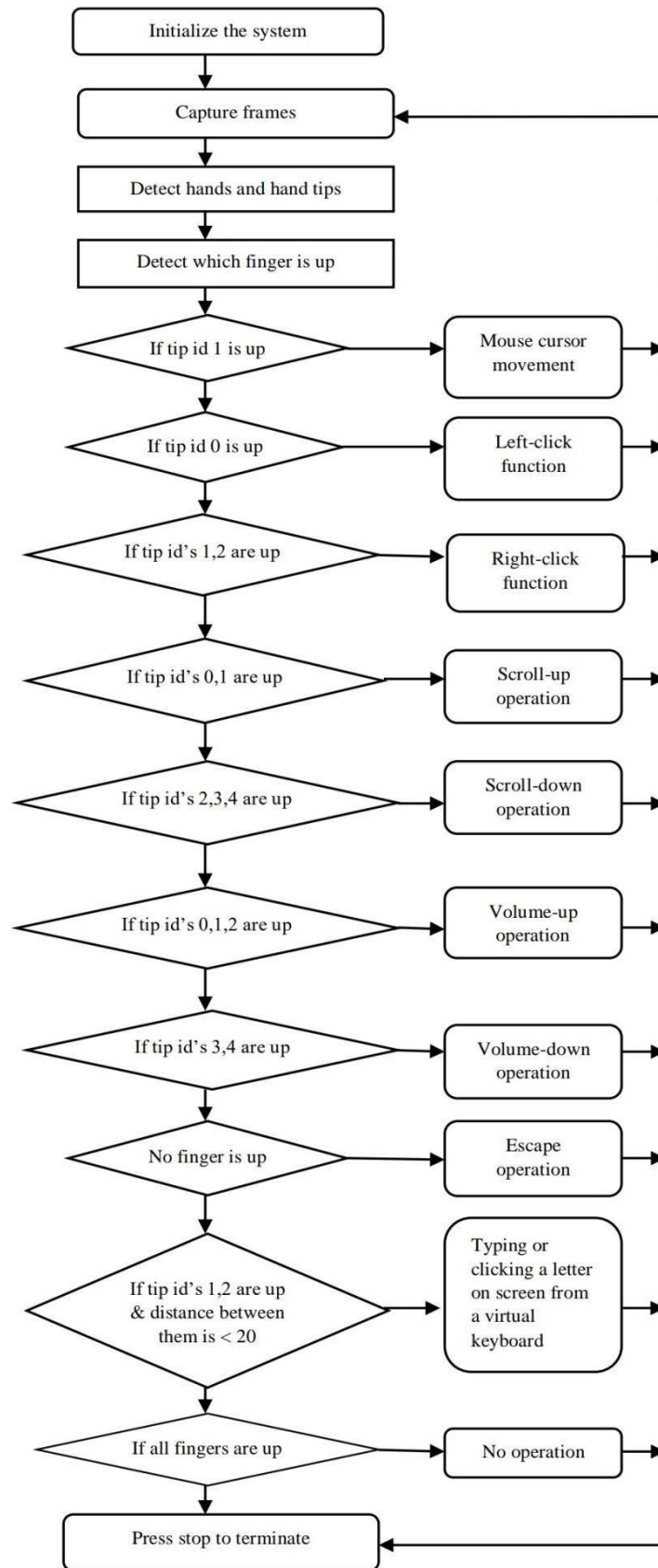


Fig 4.3: Flow-chart of real-time AI virtual mouse and keyboard system

CHAPTER 5
RESULTS AND DISCUSSIONS

SIMULATED RESULTS

This AI virtual mouse system and virtual keyboard can be created totally utilizing open- source software. So, anyone can utilize anywhere with computers, no particular preparing ought to be required to function the framework. They just need to know the hand gestures for particular operation. This project uses the concept of advancing the HCI using computer vision. In this proposed system, there is no drawback of detecting of different skin colors of hand.

The proposed systems use the following tools i.e., Python3.8 and above, OpenCV, MediaPipe, Numpy, Autopy, PyAutoGUI and time. This complete process is implemented in the PyCharm platform. Once after running the program, the camera of your device will be automatically accessed and you can start operating your system with different hand gestures. Different hand gestures for computer to perform mouse operations are given below:



Fig 5.1: Recognizing hand (No operation)

If all tip id's are up then hand is recognized, that can be observed in Fig 5.1.

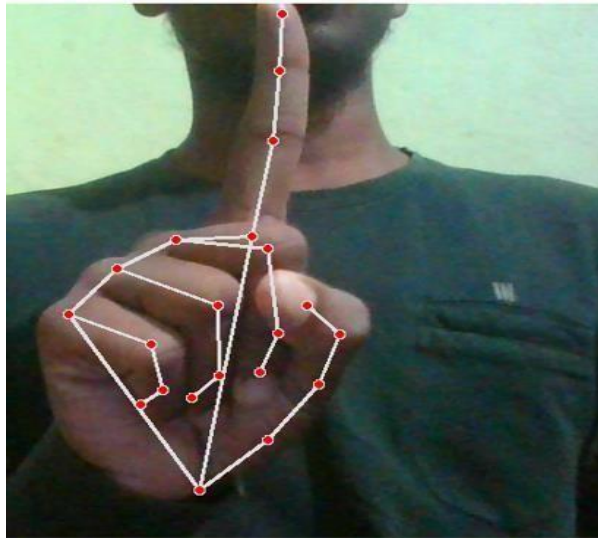


Fig 5.2: Gesture for mouse movement

If index finger with tip id 1 is up then the mouse is moved around the window of the computer by using AutoPy package as shown in Fig 5.2.

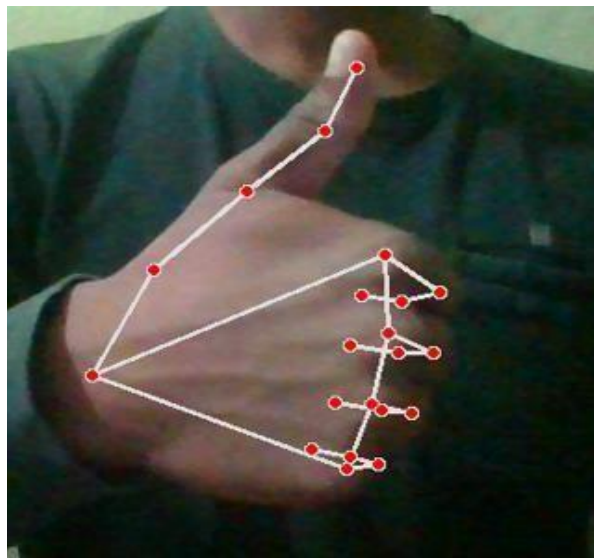


Fig 5.3: Gesture for left click function

If thumb finger with tip id 0 is up then the Left-click operation is performed by using PyautoGUI package as shown in Fig 5.3.

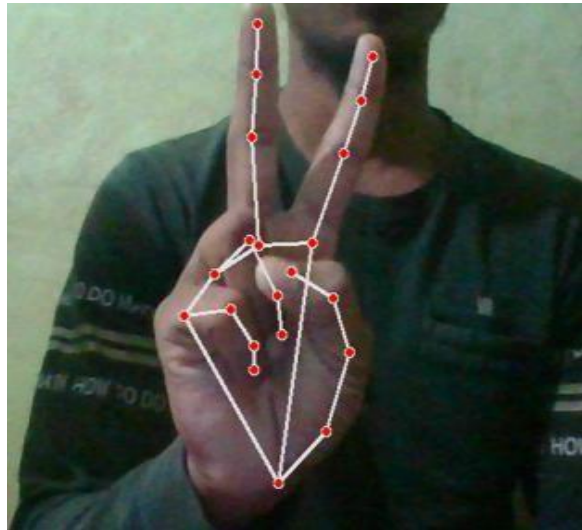


Fig 5.4: Gesture for right click function

If both index finger with tip id 1 and middle finger with tip id 2 are up then the Right-click operation is performed by using PyautoGUI package as shown in Fig 5.4.



Fig 5.5: Gesture for scroll up function

If both thumb finger with tip id 0 and index finger with tip id 1 are up then the scroll-up operation is performed by using PyautoGUI package as shown in Fig 5.5.



Fig 5.6: Gesture for scroll down function

If middle finger with tip id 2, ring finger with tip id 3 and little finger with tip id 4 are up then the scroll-down operation is performed by using PyautoGUI package as shown in Fig 5.6.

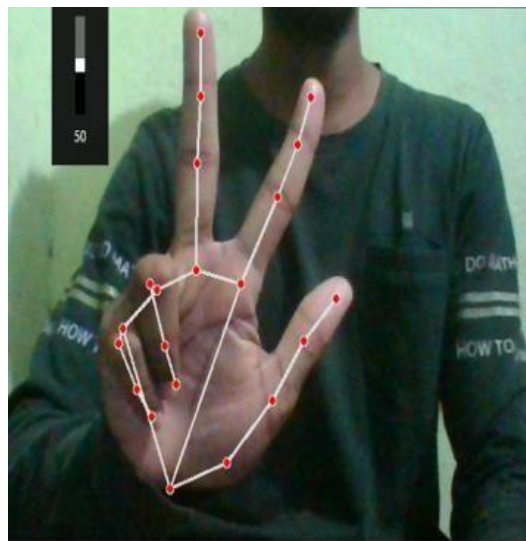


Fig 5.7: Gesture for Volume up function

If thumb finger with tip id 0, index finger with tip id 1 middle finger with tip id 2 are up then the volume-up operation is performed by using PyautoGUI package as shown in Fig 5.7.

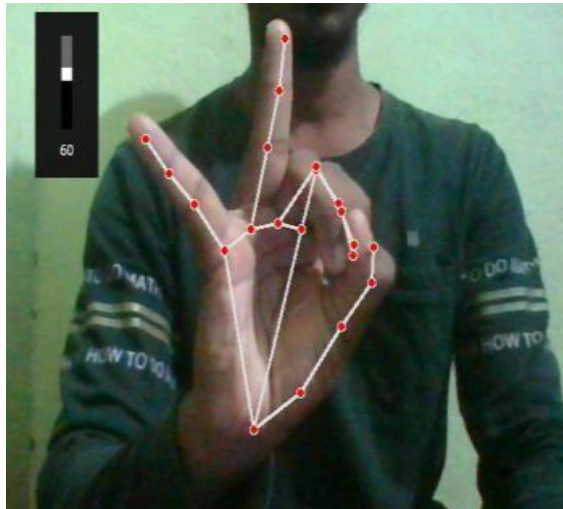


Fig 5.8: Gesture for Volume down function

If ring finger with tip id 3 and little finger with tip id 4 are up then the volume down operation is performed by using PyautoGUI package as shown in Fig5.8.



Fig 5.9: Gesture for Volume down function

If no finger is up then the escape operation is performed by using PyautoGUI package as shown in Fig 5.9.



Fig 5.10: Gesture for typing or clicking a letter on screen from virtual keyboard

Clicking on a letter or typing using a virtual keyboard is done if both index finger with tip id 1 and middle finger with tip id 2 are up and distance between them is less than 20 as shown in Fig 5.10.

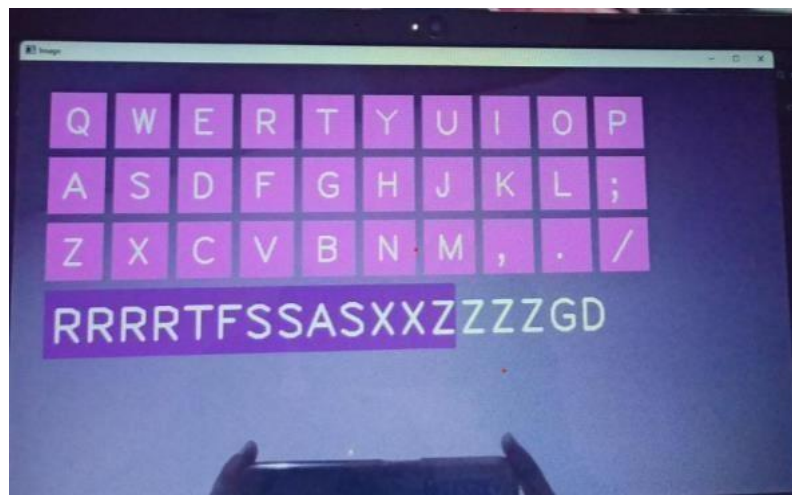


Fig 5.11: Virtual keyboard

The prototype of virtual keyboard is shown in Fig 5.11 where the typing is possible only in the specified field given on the screen when virtual keyboard is displayed.

TABLE 5.1
RESPECTIVE TIP ID'S FOR FINGERS

Tip ID	Respective finger
0	Thumb Finger
1	Index Finger
2	Middle Finger
3	Ring Finger
4	Little Finger

TABLE 5.2
TESTED RESULTS

Tip ID's	Operation performed	Success	Failure	Accuracy (%)
Tip ID 1 is up	Mouse movement	100	0	100
Tip ID 0 is up	Left button click operation	100	0	100
Tip ID's 1 and 2 are up	Right button click operation	100	0	100
Tip ID's 0 and 1 are up	Scroll-up operation	98	2	98
Tip ID's 2,3 and 4 are up	Scroll-down operation	100	0	100
Tip ID's 0,1 and 2 are up	Volume-up operation	100	0	100
Tip ID's 3 and 4 are up	Volume-down operation	100	0	100
No finger is up	Escape operation	100	0	100
All tip IDs are up (0,1,2,3,4)	No operation	100	0	100
Tip IDs 1 and 2 are up with distance between them is less than 20	Clicking on a letter or typing using a virtual keyboard	97	3	97

* Respective tip ids are given in Table 5.1.

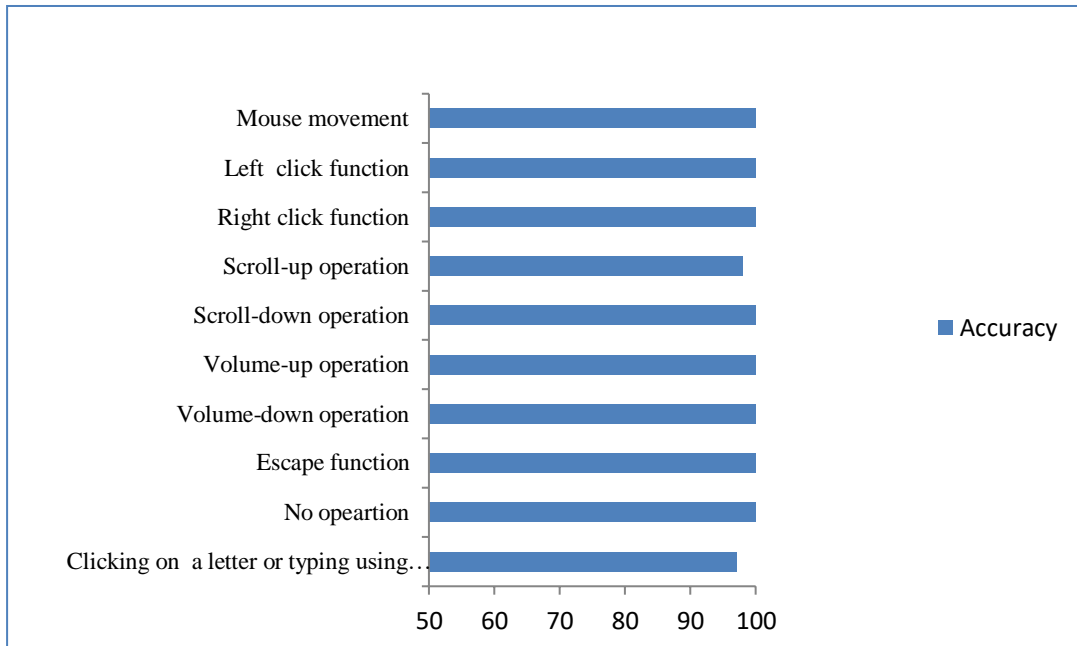


Fig 5.12: Graph determining accuracy level of each operation

From the above Table 5.12, our proposed AI virtual mouse system is 99.8% accurate and AI virtual keyboard is 97% accurate, which justifies that our system performed well. There is a bit less accuracy in scroll-up operation since we have given less clicks for one time scrolling. Since this is an open source, you can edit and provide how much scrolling you need. When compared to previous models of AI virtual mouse and keyboard, our model worked very well and the accuracy level can be observed in Table 5.2.

Performance Analysis

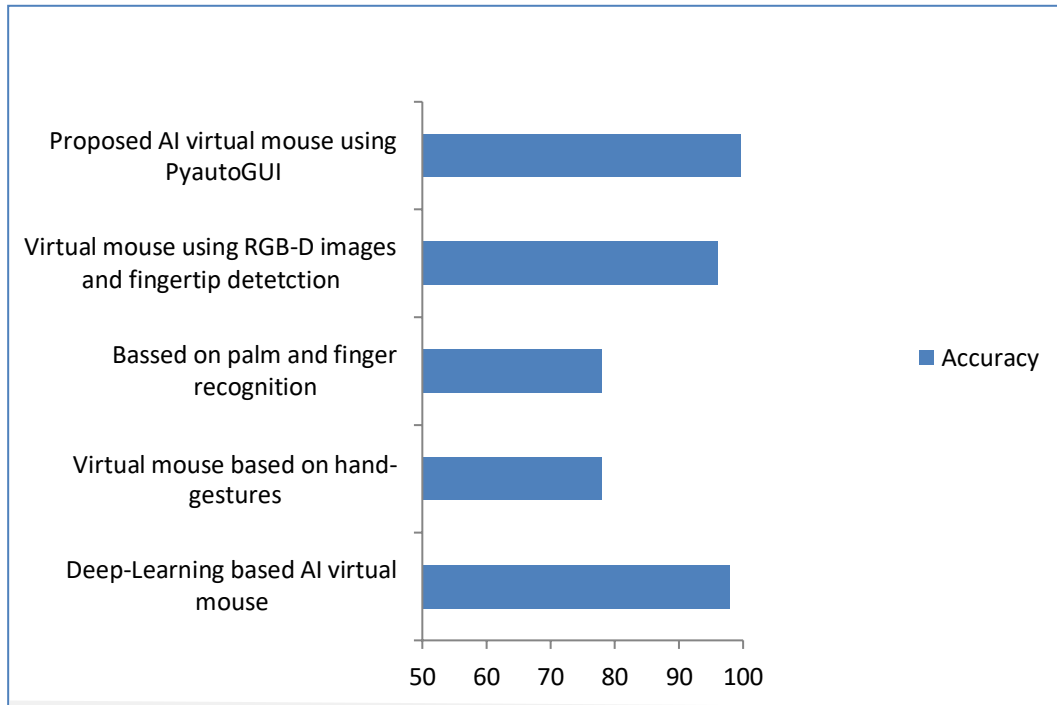


Fig 5.13: Graph for comparison between different models

We can observe the accuracy level when compared to previous models is more in our proposed model in Table 3. The graph for comparison between the models is also shown in Fig 5.13.

Applications of our proposed model

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 99.7% which is far greater than that of other proposed models for virtual mouse and keyboard, 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.

3. The system can be used to control robots and automation systems without the usage of devices.
4. 2D and 3D images can be drawn using the AI virtual system using the hand gestures.
5. AI virtual mouse can be used to play virtual reality and augmented reality-based games without the wireless or wired mouse devices.
6. In the field of robotics, the proposed system like HCI can be used for controlling robots.
7. In designing and architecture, the proposed system can be used for designing virtually for prototyping.

CHAPTER 6
CONCLUSION

CONCLUSION

This system proposes a framework that recognizes hand motions and getting freed of the requirement for a mouse and keyboard. This framework is based on computer vision calculations and can perform all mouse errands. The past demonstrate has a few confinements such as little diminish in precision in right-click and a few challenges in clicking and dragging to choose the content. From the results of the demonstrated system is ready to conclude that the proposed AI virtual mouse framework has executed well and features a more prominent precision when compared to the existing models. Additionally, this model overcomes the impediments of the existing frameworks. The AI virtual mouse and keyboard are useful for many applications such as designing and architecture, controlling robots, automation systems and also this model is used to lessen the spread of COVID-19 as there will be no human interventions.

FUTURE SCOPE

In this proposed system just the prototype for a keyboard is represented but it can be developed in such a way which can be used to type anywhere rather than in only a specified place on the virtual keyboard.

Paper Publication details

The proposed paper is communicated to the 7th International Conference on Micro-Electronics, Electromagnetics and Telecommunications, ICMEET 2022, 22-23 JULY 2022, organized by Department of Electronics and Communication Engineering, Shri Vishnu Engineering College for Women (Autonomous), Bhimavaram, Andhra Pradesh, India.

REFERENCES

- [1] J. Katona, "A review of human–computer interaction and virtual reality research fields in cognitive Info Communications," *Applied Sciences*, vol. 11, no. 6, p. 2646, 2021.
- [2] P. M. Game and A. R Mahajan, "A gestural user interface to Interact with computer system ", *International Journal on Science and Technology (IJSAT)*, vol. II, no. I, pp. 018- 027, Jan. - Mar.2011.
- [3] Erdem, E. Yardimci, Y. Atalay, V. Cetin, 2002. Computer vision -based mouse, *Proceedings. (ICASS). IEEE International Conference.*
- [4] Crowley, J., Bérard, F., and Coutaz, J. "Finger tacking as an input device for augmented reality, *Automatic Face and Gesture Recognition*", Zurich, 195--200,1995.
- [5] G.R. Bradski, "Computer video face tracking for use in a perceptual user interface", *2nd Quarter, Intel Technology Journal*, 1998.
- [6] Tsung-Hsiang Chang, Tom Yeh, and Robert C Miller. 2010. "GUI Testing Using Computer Vision". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2010)*, 1535–1544.
- [7] T. Murase, et al., "Gesture keyboard requiring only one camera," *Proc. of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology*, Santa Barbara, USA, pp. 9-10, Oct.2011.
- [8] Cecotti, H.: A multimodal gaze-controlled virtual keyboard. *IEEE Transactions on Human-Machine Systems* 46(4), 601–606(2016).
- [9] P. Nandhini, J. Jaya, and J. George, "Computer vision system for food quality evaluation—a review," in *Proceedings of the 2013 International Conference on Current Trends in Engineering and Technology (ICCTET)*, pp. 85–87, Coimbatore, India, July2013.
- [10] Hernanto, S., Suwardi, I.S.: Webcam virtual keyboard. In: *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. pp. 1–5. IEEE (2011).
- [11] Yousaf, M.H., Habib, H.A.: Virtual keyboard: real-time finger joints tracking for keystroke detection and recognition. *Arabian journal for science and engineering* 39(2), 923–934(2014).
- [12] Shetty, S., Yadav, S., Upadhyay, R., Bodade, V.: Virtual mouse using colour detection

(2016).

- [13] D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, “Real-time virtual mouse system using RGB-D images and fingertip detection,” *Multimedia Tools and Applications* *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10473–10490,2021.
- [14] *International Research Journal of Engineering and Technology (IRJET)* “VIRTUAL MOUSE APPLICATION”, Volume: 08 Issue: 07 | July2021.
- [15] H. Shibly, S. Kumar Dey, M. A. Islam, and S. Iftexhar Showrav, “Design and development of hand gesture based virtual mouse,” in *Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1– 5, Dhaka, Bangladesh,May2019.
- [16] J.T.Camillo Lugaesi, “Media Pipe: A Framework for Building Perception Pipelines,”2019.
- [17] D - H. Liou, D. Lee, and C.-C. Hsieh, “A real time hand gesture recognition system using motion history image,” in *Proceedings of the 2010 2nd International Conference on Signal Processing Systems*, July2010.
- [18] Haria, A. Subramanian, N. Asokkumar, S. Poddar, andJ. S. Nayak, “Hand gesture recognition for human computer interaction,” *Procedia Computer Science*, vol. 115, pp. 367–374,2017.
- [19] Adajania, Y., Gosalia, J., Kanade, A., Mehta, H., Shekokar, N.: Virtual keyboard using shadow analysis. In: *2010 3rd International Conference on Emerging Trends in Engineering and Technology*. pp. 163–165. IEEE (2010).
- [20] Krejov P, Bowden R (2013) Multi-touch less: Real-time fingertip detection and tracking using geodesic maxima. In: *Proceedings IEEE International Conference on Automatic Face and Gesture Recognition*.
- [21] S.Shriram, B.Nagaraj, J.Jaya, S.Shankar, P.Ajay (2021), Deep learning-based real-time AI Virtual mouse system using computer vision to avoid COVID-19 Spread, *Hindawi* . Volume 2021, Article ID 8133076.