**HAND GESTURE RECOGNITION**

A Project-I Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Bhumi Gupta -EN19CS301094**

**Bhojraj Pathak -EN19CS301093**

**Ayush Goyal -EN19CS301086**

Under the Guidance of

**Prof. Snehal Moghe**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**AUGUST-DECEMBER 2022**

**Report Approval**

The project work **“HAND GESTURE DETECTION FOR MOUSE CONTROL "is** hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

**Internal Examiner**

Name: Prof. Snehal Moghe

Designation: Assistant Professor

Affiliation: Medi-Caps University, Indore

**External Examiner**

Name:

Designation:

Affiliation:

**Declaration**

I/We hereby declare that the project entitled **“HAND GESTUERE DETECTION FOR MOUSE CONTROL”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications in ‘COMPUTER SCIENCE AND ENGINEERING DEPEARTMENT’ completed under the supervision of **Mrs. Snehal Moghe, Assistant Professor Computer Science & Engineering,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Bhumi Gupta 23/11/2022**

**Certificate**

I, **Snehal Moghe** certify that the project entitled **“HAND GESTURE DETECTION FOR MOUSE CONTROL”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications by **Bhumi Gupta** istherecordcarried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Mrs. Snehal Moghe |  | Dr. Pramod S. Nair |
| Computer Science & Engineering |  | Head of the Department |
| Medi-Caps University, Indore |  | Computer Science & Engineering |
|  |  | Medi-Caps University, Indore |

**Acknowledgements**

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) D. K. Patnaik,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) Suresh Jain,** Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my Guide, **Mrs. Snehal Moghe ,** Assistant Professor, Department of Computer Science & Engineering, MU, without whose continuous help and support, this project would ever have reached to the completion.

I would also like to thank to my team students name who extended their kind support and help towards the completion of this project.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

**Bhumi Gupta**

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

**Abstract**

This project and experiment were conducted with the aim of utilizing the human hands as an object to operate computers. It is intended to support and use technologies in the field of contactless shopping/payments. The program is developed by using python programming language with the help of additional libraries such as OpenCV. In order to use this program, the person needs to be in front of a computer webcam. The webcam will be used to recognize the shape and the pattern of the presenters’ hands. The program will display results of recognized hand patterns of gestures on a live video frame stream. The result of this project is a program that can be used for improve the user experience of contactless systems and make safer transactions in a time of global pandemic of 2020, where social distancing is one of the main things to consider.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **Page No.** |
|  |  |  |
|  | Report Approval | ii |
|  | Declaration | iii |
|  | Certificate | iv |
|  | Acknowledgement | v |
|  | Abstract | vi |
|  | Table of Contents | vii |
|  | List of figures | viii |
|  | List of tables | ix |
|  | Abbreviations | x |
|  | Notations & Symbols | xi |
| Chapter 1 | Introduction |  |
|  | 1.1 Introduction | 1 |
|  | 1.2 Literature Review | 2 |
|  | 1.3 Objectives |  |
|  | 1.4 Scope |  |
|  | 1.5 Research Design |  |
|  | 1.6 Source of Data |  |
|  | 1.7 Chapter Scheme |  |
| Chapter 2 | System Requirement Specification |  |
|  | 2.1 Existing System |  |
|  | 2.2 Proposed System |  |
|  | 2.2 Procedures Adopted |  |
|  | 2.3 System Feasibility |  |
|  | 2.4 Hardware Specification |  |
|  | 2.5 Software Specification |  |
|  | 2.6 Functional Requirement |  |
|  | 2.7 Non-Functional Requirement |  |
| Chapter 3 | System Analysis & Design |  |
|  | 3.1 ER Diagram |  |
|  | 3.2 Data Flow Diagram |  |
|  | 3.3 Use case Diagram |  |
|  | 3.4 Class Diagram |  |
|  | 3.5 Activity Diagram |  |
|  | 3.6 Sequence Diagram |  |
|  | 3.7 Architectural Design |  |
| Chapter 4 | Implementation |  |
|  | 4.1 Procedural Description |  |
|  | 4.2 Module Description |  |
|  | 4.3 Database Implementation (with screenshots) |  |
|  | 4.4 User Interface Implementation (with screenshots) |  |
|  | 4.5 Technical Discussions |  |
| Chapter 5 | Testing |  |
|  | 5.1 Unit Testing |  |
|  | 5.2 Integration Testing |  |
|  | 5.3 Validation Testing |  |
|  | 5.4 White Box Testing |  |
|  | 5.5 Performance Testing |  |
| Chapter 7 | Results and Discussions |  |
| Chapter 8 | Summary and Conclusions |  |
| Chapter 9 | Future scope |  |
|  | Appendix |  |
|  | Bibliography |  |
|  | List of Publications (If any) |  |
|  | Reprints of publications (If any) |  |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **Page No.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **Page No.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Abbreviations**

|  |  |
| --- | --- |
| SRS | SOFTWARE REQUIREMENT SPECIFICATION |
|  |  |

**Chapter-1**

**Introduction**

**1.1 Introduction**

Gesture recognition is an active research field in Human-Computer Interaction technology. It has many applications in virtual environment control and sign language translation, robot control, or music creation. In this machine learning project on Hand Gesture Recognition, we are going to make a real-time Hand Gesture Recognizer using the Media Pipe framework and Tensor flow in OpenCV and Python. OpenCV is a Realtime Computer vision and image processing framework built on C/C++. But we’ll use it on python via the OpenCV-python package.

This project promotes an approach for the Human Computer Interaction (HCI) where cursor movement can be controlled using a real-time camera, it is an alternative to the current methods including manual input of buttons or changing the positions of a physical computer mouse. Instead, it utilizes a camera and computer vision technology to control various mouse events and is capable of performing every task that the physical computer mouse can. The Virtual Mouse color recognition program will constantly acquiring real-time images where the images will undergo a series of filtration and conversion. Whenever the process is complete, the program will apply the image processing technique to obtain the coordinates of the targeted colors position from the converted frames. After that, it will proceed to compare the existing colors within the frames with a list of color combinations, where different combinations consist of different mouse functions. If the current colors combination found a match, the program will execute the mouse function, which will be translated into an actual mouse function to the users' machine.

**1.2 Literature Review**

[1] James Davis, Mubarak Shah, ― Recognizing Hand Gestures‖ Computer Vision Laboratory University of Central Florida, Orlando FL 32816 USA To appear in ECCV-94, Stockholm, Sweden, page 245 -258, 1994.

[2] Rafiqul Zaman Khan, Noor Adnan Ibraheem, ―hand gesture recognition: a literature review‖ department of computer science, a.m.u. Aligarh, India international journal of artificial intelligence & applications (ijaia), vol.3, no.4, page 161 to 174, 2012.

[3] Sushmita Mitra, Tinku Acharya, ―Gesture Recognition: A Survey‖ ieee transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 37, no. 3, page 311 to 324,2007.

[4] Abhik Banerjee, Abhirup Ghosh, Kousumi Bharadwaj,” Mouse Control using a Web Camera based on Color Detection”, IJCTT,vol.9, 2014.

**2.1 Objective**

The objective of this paper is to Develop and implement a computer application that utilizes alternate methods for cursor control. Thus, proposes a novel vision-based cursor control system, using hand gestures captured from a webcam by using colour detection technique. The goal of this paper is to create a system that will recognize the hand gestures and control the computer/laptop according to those gestures using colour detection technique.

1. Objective of this project is to create a complete system to detect, recognize and interpret the hand gestures through computer vision.

2. Objective of the project is therefore to provide a new low-cost, high speed and colour image acquisition system.

**2.2 Scope**

The Hand Gesture recognition is moving at tremendous speed for the futuristic products and services and major companies are developing technology based on the hand gesture system and that includes companies like Microsoft, Samsung, Sony and it includes the devices like Laptop, Hand held devices, Professional and LED lights. The verticals include where the Gesture technology is and will be evident are Entertainment, Artificial Intelligence, Education and Medical and Automation fields. And with lot of Research and Development in the field of Gesture Recognition Field, the use and adoption will become more cost effective and cheaper. It’s a brilliant feature turning data into features with mix of technology and Human wave. Smart phones have been experiencing enormous amount of Gesture Recognition Technology with look and views and working to manage the Smartphone in reading, viewing and that includes what we call touch less gestures. Google Glass has been also in the same cadre. And the Technology has also been embedded into smart televisions nowadays as well, which can easily control and managed by Voice and Hand options. In the medical fields Hand Gesture may also be experienced in terms of Robotic Nurse and medical assistance. As the Technology is always revolving and changing the future is quite unpredictable but we have to be certain the future of Gesture Recognition is here to stay with more and eventful and Life touching experiences.

2.3 Chapter’s Key

The Chapter Scheme in the project report is planned to be as under:

Chapter 1 Introduction

Chapter 2 System Requirement Specification

Chapter 3 System Analysis and Design

Chapter 4 Implementation

Chapter 5 Testing

Chapter 6 Results and Discussions

Chapter 7 Summary and Conclusion

Chapter 8 Future Scope

**Chapter-2**

**System Requirement Specification**

**2.1 Existing System**

The existing system consists of a mouse that can be either wireless or wired to control the cursor, know we can use hand gestures to monitoring the system. The existing virtual mouse control system consists of the simple mouse operation using the coloured tips for detection which are captured by web-cam, hence coloured fingers act as an object which the web-cam sense colour like red, green, blue colour to monitor the system, whereas could perform basic mouse operation like minimize, drag, scroll up, scroll down, left-click right-click using hand gestures without any coloured finger because skin colour recognition system is more flexible than the existing system. In the existing system use static hand recognition like fingertip identification, hand shape, Number of fingers to defined action explicitly, which makes a system more complex to understand and difficult to use.

**2.2 Proposed System**

Most gesture recognition methods usually contain three major stages. The first stage is the object detection. The target of this stage is to detect hand objects in the digital images or videos. Many environment and image problems are needed to solve at this stage to ensure that the hand contours or regions can be extracted precisely to enhance the recognition accuracy. Common image problems contain unstable brightness, noise, poor resolution and contrast. The better environment and camera devices can effectively improve these problems. However, it is hard to control when the gesture recognition system is working in the real environment or is become a product. Hence, the image processing method is a better solution to solve these image problems to construct an adaptive and robust gesture recognition system. The second stage is object recognition. The detected hand objects are recognized to identify the gestures. At this stage, differentiated features and effective classifiers selection are a major issue in most researches. The third stage is to analyse sequential gestures to identify users instructs or behaviours.[2]

**2.3Procedures-Adopte**

1. Initialize webcam and capture video sources.

2. The system performs Hand detection.

3. Any Hand in the image is filtered and the frame is blurred.

4. The frame is then passed to filter out backgrounds and extract features.

5. With only the image of hands remain in the image, the system will locate the

hand and perform geometry translations.

6. Perform action:

a. Mouse control mode: detect hand movements and translate the coordinates to user’s screen.

i. cursor movement

ii. right click

iii. left click

b. Browsing mode: detect gestures and perform the following actions:

i. Swipe left

ii. Swipe right

iii. Scroll up

iv. Scroll down.

7. Repeat whole cycle until system pause or system end.

**2.4 System Feasibility**

This proposed system presented a new approach to hand gesture recognition for mouse control using a combination of geometry algorithm and a deep learning method to achieve detection and gesture recognition tasks. This approach exhibited not only highly accurate gesture estimates, but also suitability for practical applications. The proposed method has many advantages, for example, working well in changing light levels or with complex backgrounds, accurate detection of hand gestures at a longer distance, experimental results indicated that this approach is a promising technique for hand-gesture-based interfaces in real time. System is not feasible for recognizing the hand gestures of multiple people. The future work with hand gesture recognition, we intend to expand our system to handle more hand gestures and apply our method in more other practical applications.

**2.5 Hardware Specification:**

•Processor Intel Core i5 -3230M 2.6GHz

•Memory 8GB DDR3 RAM

•Graphics NVIDIA GeForce GT 720M 2GB Graphics or above

•Display 14" HD Display (16:9) Wide Screen

•Camera Integrated Web Camera

•Communications Wi-Fi 802.11b/g/n, Ethernet Port

•Software Genuine Windows 10

•Webcam 1MP Fixed Focus CMOS camera on the laptop

**2.6 Software Specification:**

Microsoft Window 8 or above Support Architecture:

• 32bit(x86)

• 64bit(x86)

• 1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)

• 16 GB available hard disk space (32-bit)

• Spyder

• Python 3.8.3

• Open cv 4.5.3

• NumPy 1.19

2.7 Functional Requirement

Functional requirements specify the main technical functionalities and specifications that the system should incorporate.

* **Face Detection:**

This software shall utilize a face detection system to filter out faces from the video capturing device. By applying face detection, the system can disregard the region where the face is located and thus reducing the amount of calculation needed to perform hand detection. The face detection unit will be implemented with the help of OpenCV.

* **Skin Detection Module:**

This software shall perform skin color detection and filter out all objects that do not contain the color of skin. By filtering object of non-skin colored, the system can then use its remaining resources and focus on hand detection and gesture recognitions.

* **Object Location**:

Upon detection, the system shall be able to compute the location of the object using simple trigonometry math.

* **Mouse Movement Gesture Control Mode:**

After obtaining the location of the hand, the software shall use the detected location as the mouse cursor point. As the user moves his/her hands, the mouse should follow promptly on the screen.

* **Browsing Gesture Control Mode:**

The software shall allow the user to use the “Browsing Gesture Mode”. In this mode, user’s hand gesture will only be recognized for commands including previous page, next page, scroll up and scroll down

**Non-Functional Requirements:**

Non-functional requirements specify the criteria in the operation and the architecture of the system.

* **Efficiency in Computation**:

When HGR is executing, the software shall utilize less system’s CPU resource and very little of system memory.

* **Extensibility:**

The software shall be extensible to support future developments and add-ons to the HGR software. The gesture control module of HGR shall be extensible to allow new gesture recognition features to be added to the system.

* **Portability:**

The HGR software shall be 100% portable to all operating platforms. Therefore, this software should not depend on the different operating systems.

* **Performance**:

This software shall minimize the number of calculations needed to perform image processing and hand gesture detection.

* **Reliability:**

The HGR software shall be operable in all lighting conditions. Regardless of the brightness level in user’s operating environment, the program shall always detect user’s hands.

* **Usability**:

This software shall be easy to use for all users with minimal instructions. The graphical user interface (GUI) shall be intuitive and understandable by non-technical users too.

**Chapter-3**

**System Analysis & Design**

**3.1 Use case Diagram**

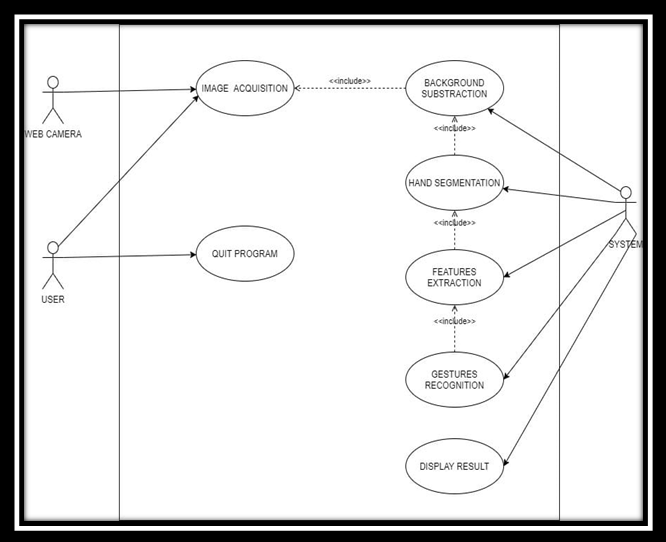


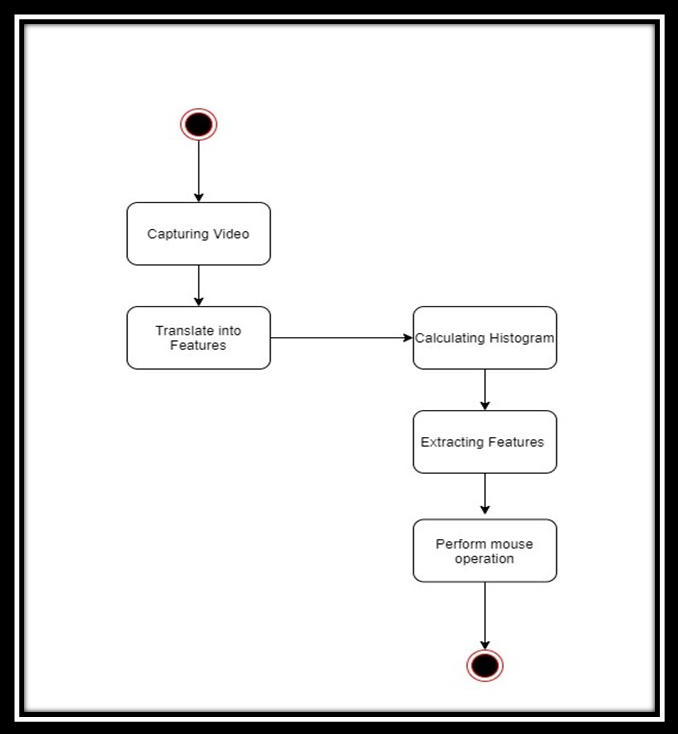
Figure .1

USE CASE DESCRIPTION

* Figure.1Shows that the use case diagram has three actors which is the user the camera and system. The user is the one that can initialize the system and quit the program after that. Besides, user also associate with the camera which is the laptop webcam or an external webcam that capture user image in real-time as the system input. In addition, the use case diagram also consists of 7 use cases where the image acquisition, background subtraction, hand segmentation, features extraction, gesture recognition and display output are the core processing stages of the real-time gesture recognition system. The use cases of the system as below is describing the actions that perform by the actors and what will be the expected outcome.
* Use Case 1: Image Acquisition
* Actor: Camera/ User
* Goal: To capture the video sequence of user hand image as system input.
* Overview: The laptop webcam or external webcam is used to capture the user’s hand image.
* Use Case 2: Background Subtraction
* Actor: Camera/ User
* Goal: To process the video sequence for extracting the user’s hand and remove unnecessary background and noise that associate with it.
* Overview: Background subtraction, color space conversion, thresholding and morphological transformation will be performed in order to prepare the binary image of user hand without unnecessary object and noises from a clustered background for the next processing stage.

* Use Case 3: Hand Segmentation
* Actor: Camera/ User
* Goal: To obtain the hand contour and maximum contour of hand.
* Overview: Hand contour is obtained from the binary image and get the largest contour in the image for the next stage.
* Use Case 4: Features Extraction
* Actor: Camera/ User
* Goal: To obtain a set of hand features as the useful information that will be used for analyzing and determine the meaning of the gesture input to perform specific function.
* Overview: It is the process to transform the image data into a set of hand features such as palm center, convex hull, fingertips, hand defect points, area of hand, area ratio which is the percentage of area not covered by hand in convex hull and angle of finger as well.
* Use Case 5: Gesture Recognition
* Actor: Camera/ User
* Goal: To apply set of rules on the extracted information to determine the meaning of the gesture input and display the gesture that has been recognized.
* Overview: The meaning of the gesture input will be determined by set of rules which include hand area, area ratio, number of defect point, number of fingers, and angle of finger as well. The meaning of gesture will be displayed after it is being recognized.
* Use Case 6: Quit program
* Actor: User
* Goal: To quit the program after it is being initialized.
* Overview: User press on the key to quit the program.
* Use Case 7: Display output
* Actor: System
* Goal: To perform mouse operations after recognition.
* Overview: Perform mouse operation according to your data

***3.2State chart Diagram***



**Figure .2**

***State diagram description***

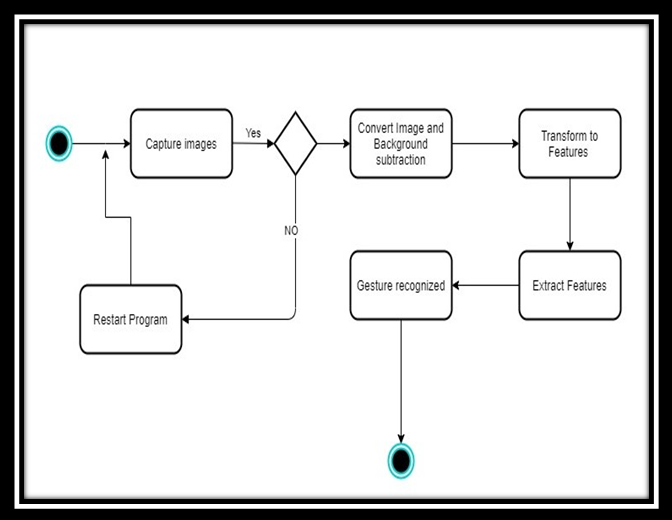
State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists, and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination. State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system. When structuring a state diagram, there are several notations and symbols you can use. With these tools, you can have an excellent presentation and a better understanding of the system you are trying to project.

•A State: It is the current physical situation of an object. A rounded edge rectangle usually represents it, and in the middle of it is a labeling option. You can also fill it with a color of your choice. The state of an object will generally change due to changing events.

•Start state: It is the point from which a state diagram's flow begins. It's the initial step or stage before any events start happening. The symbol is indicated by a plain circle that you can fill with any color on E draw Max.

•End State: The flow of a state diagram will always come to an end. This endpoint is indicated by the end state, a circle with a round border. In the current E draw Max version, it appears as a plain circle when filled with color.

***3.3Activity Diagram***



**Figure 3**

ACTIVITY DIAGRAM DESCRIPTION

Basic Activity diagram notation and symbols •Start State: A small, filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swim lanes, make sure the start point is placed in the top left corner of the first column. •Activity or Action state: An action state represents the non-interruptible action of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.

• Action flow: Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.

• Object flow: Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.

• Decisions and Branching: A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labelled with a condition or guard expression. You can also label one of the paths "else."

• Guards In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.

• Synchronization

A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram. A join node joins multiple concurrent flows back into a single outgoing flow. A fork and join mode used together are often referred to as synchronization.

•Sent and Received

Signals represent how activities can be modified from outside the system. They usually appear in pairs of sent and received signals, because the state can't change until a response is received, much like synchronous messages in a sequence diagram. For example, an authorization of payment is needed before an order can be completed.

• Swim lanes Swim lanes group related activities into one column.

• Final State or End Point An arrow pointing to a filled circle nested inside another circle represents the final action state

**3.4 *Class Diagram***

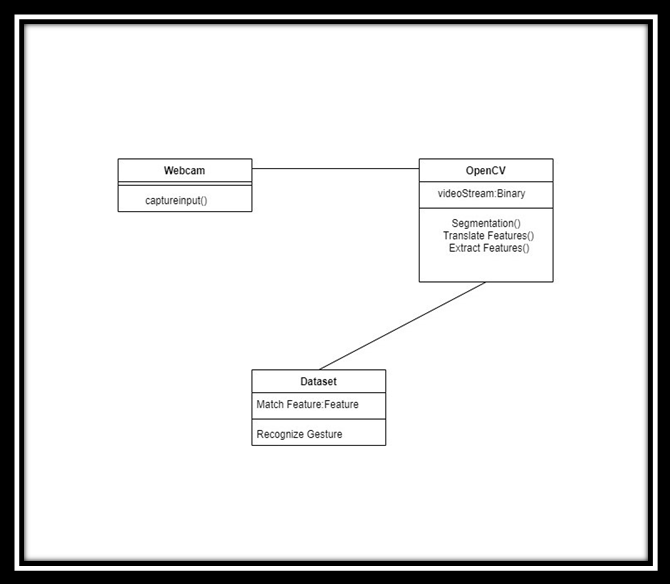


Figure .4

**Class diagram Description**

The UML Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's: classes their attributes, operations (or methods), and the relationships among objects.

Class Name:

• The name of the class appears in the first partition.

Class Attributes:

• Attributes are shown in the second partition.

• The attribute type is shown after the colon.

• Attributes map onto member variables (data members) in code.

Class Operations (Methods):

• Operations are shown in the third partition. They are servicing the class provides.

• The return type of a method is shown after the colon at the end of the method signature.

• The return type of method parameters is shown after the colon following the parameter name. Operations map onto class methods in code

Class Visibility:

The +, - and # symbols before an attribute and operation name in a class denote the visibility of the attribute and operation.

• + denotes public attributes or operations.

• - denotes private attributes or operations • # denotes protected attributes or operations.

***3.5 Sequence Diagram***

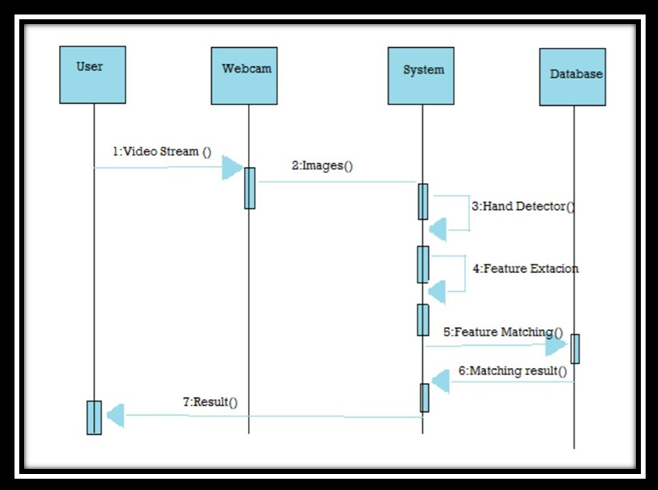


Figure .5

Sequence Diagram Description

Sequence Diagram The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of a Sequence Diagram:

1. To model high-level interaction among active objects within a system.

2. To model interaction among objects inside a collaboration realizing a use case.

3. It either model's generic interactions or some certain instances of interaction.

Notations of a Sequence Diagram:

Lifeline:

An individual participant in the sequence diagram is represented by a lifeline. It is positioned at the top of the diagram.

Actor:

A role played by an entity that interacts with the subject is called as an actor. It is out of the scope of the system. It represents the role, which involves human users and external hardware or subjects. An actor may or may not represent a physical entity, but it purely depicts the role of an entity. Several distinct roles can be played by an actor or vice versa.

Activation:

It is represented by a thin rectangle on the lifeline. It describes that time period in which an operation is performed by an element, such that the top and the bottom of the rectangle is associated with the initiation and the completion time, each respectively.

Messages:

The messages depict the interaction between the objects and are represented by arrows. They are in the sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines.

Following are types of messages enlisted below:

• Call Message: It defines a particular communication between the lifelines of an interaction, which represents that the target lifeline has invoked an operation.

• Return Message: It defines a particular communication between the lifelines of interaction that represent the flow of information from the receiver of the corresponding caller message.

• Self Message: It describes a communication, particularly between the lifelines of an interaction that represents a message of the same lifeline, has been invoked.

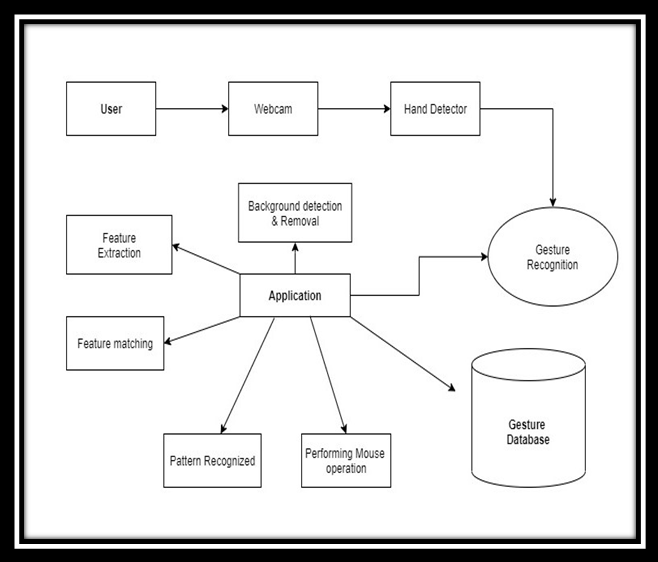
• Recursive Message: A self-message sent for recursive purpose is called a recursive message. In other words, it can be said that the recursive message is a special case of the self-message as it represents the recursive calls.

• Create Message: It describes a communication, particularly between the lifelines of an interaction describing that the target (lifeline) has been instantiated.

• Destroy Message: It describes a communication, particularly between the lifelines of an interaction that depicts a request to destroy the lifecycle of the target.

• Duration Message: It describes a communication particularly between the lifelines of an interaction, which portrays the time passage of the message while modelling a system.

***3.6 D*ataflow diagram**

**Figure .6**

Data flow Diagram Description

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.

Data Flow: A curved line shows the flow of data into or out of a process or data store.

Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

***3.7 Block –Diagram***

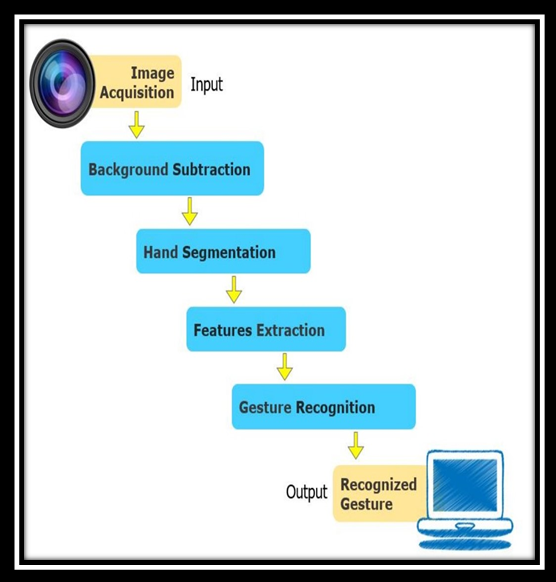


Figure .7

**Block -Diagram Description**

The main goal of drawing a block diagram is to give an overview of the workflow that could be expected from the system post its completion. With the clear illustration, it becomes easy for the engineers to assess the smooth functioning of the process and to identify the existing elements (or the missing ones) that might obstruct, hinder, or unnecessarily delay the output. The Benefit of having a block diagram is that it helps the engineers and their teammates to understand the idea without going through lengthy sentences. This helps the staff overcome the language barrier that many organizations/departments have to face when they hire people from varied regions of the world.

***3.8 Architectural-Diagram***

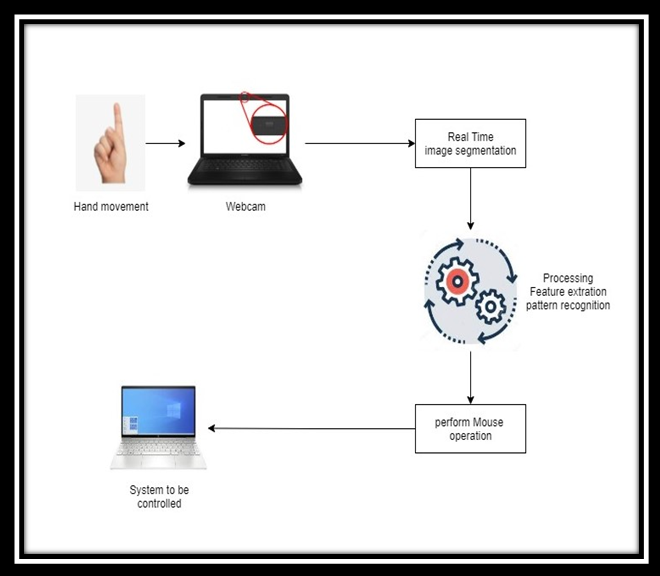


Figure .8

**Architectural-Diagram Description**

An architecture diagram is a diagram that depicts a system that people use to abstract the software system's overall outline and build constraints, relations, and boundaries between components. It provides a complete view of the physical deployment of the evolution roadmap of the software system.

A diagram is similar to a picture. The architecture diagram examples serve various functions. It always helps the relevant users to learn about system architecture and apply it in the decision-making procedures. It is crucial to communicate information regarding architecture. However, people must follow specific steps before making a diagram for architecture. These are:

• Breaking down communication barriers

• Reaching a consensus

• Decreasing ambiguity

**Chapter-4**

**Implementation**

**4.1 Procedural Description**

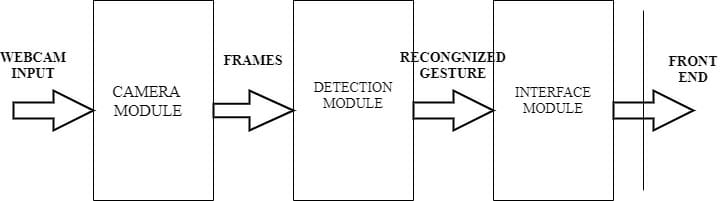
that utilizes alternate methods for cursor control. Thus, proposes a vision-based cursor control system, using hand gestures captured from a webcam by using media pipe hand detection technique. The goal is to create a system that will recognize the hand gestures and control the computer/laptop according to those gestures.

We have to follow the procedure: -

1. Download zip File and extract the folder named Hand Tracking Module.
2. Open folder in PyCharm
3. Import open cv
4. Import media pipe
5. Import autopy
6. Run the application

**4.2 Module description**

The overall system consists of two parts, back end and front-end. The back-end system consists of three modules: Camera module, Detection module and Interface module as shown in Fig. 1. They are summarized as follows:



**Camera module**: This module is responsible for connecting and capturing input through the different types o image detectors and sends this image to the detection module for processing in the form of frames. The commonly used methods of capturing input are data gloves, hand belts and cameras. In our system, we use the webcam inbuilt which is cost efficient to recognize both static and dynamic gestures. The system has suitable provision to allow input from a USB based webcam as well but this would require some expenditure from the user. The image frames obtained are in the form of a video.

**Detection module**: This module is responsible for the image processing. The output from camera module is subjected to different image processing techniques such as color conversion, noise removal, thresholding following which the image undergoes contour extraction. If the image contains defects, then convexity defects are found according to which the gesture is detected. If there are no defects, then the image is classified using Haar cascade to detect the gesture. In the case of dynamic gestures, the detection module does the following; If Microsoft PowerPoint has been launched with a slideshow being enabled and the webcam detects palm in movement, for 5 continuous frames then the dynamic gesture swipe is detected.

**Interface module**: This module is responsible for mapping the detected hand gestures to their associated actions. These actions are then passed to the appropriate application. The front end consists of three windows. The first window consists of the video input that is captured from the camera with the corresponding name of the gesture detected. The second window displays the contours found within the input images. The third window displays the smooth threshold version of the image. the advantage of adding the threshold and contour window as a part of the Graphical User Interface is to make the user aware of the background inconsistencies that would affect the input to the system and thus, they can adjust their laptop or desktop web camera in order to avoid them. This would result in better performance.

**4.3 Database implementation**

**DATA SET**

The most significant piece of the whole report spins around the predefined dataset, which incorporates the motions to be perceived by means of information signal

1.For moving the cursor, we will use the single finger.



2.For double click, we will use two fingers together.



3.For left click we will use, finger and thumb together.



4.This set contains 4 hand-gestures, each one composed of two different hand-poses:

A: scroll down

B: swipe left

C: swipe right

D: scroll up

**Chapter-5**

**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Test scenario ID | Hand Tracking mouse 1 | Test case ID | Hand Tracking mouse |
| Test case description | Mouse using real time hand tracking | Test priority | High |
| Prerequisite | All the above mentioned software | Post- Requisite | NA |
|  |  |  |  |

Test Execution Steps:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.No | Actions | | Input | Expected output | | Actual output | Test Browser | Test Result | Test Comments |
| 1. | Launch  Application | | Run code in PyCharm | Camera popped up | | Camera popped up | PyCharm | Pass | [Ishank 5/11/2021 11:44 AM]: Launch successful |
| 2. | Moving Index finger to control cursor | | Index finger moving input | Cursor moving | | Cursor moving | PyCharm | Pass | [Ishank 27/10/2021 11:45 AM]: cursor movement success |
| 3. | | Using two finger to perform click operation of mouse | Joining Two fingers for performing click operation | | Click operation performed | Click operation performed | PyCharm | Pass | [Ishank 5/11/2021 11:46 AM]: action preforms successful |

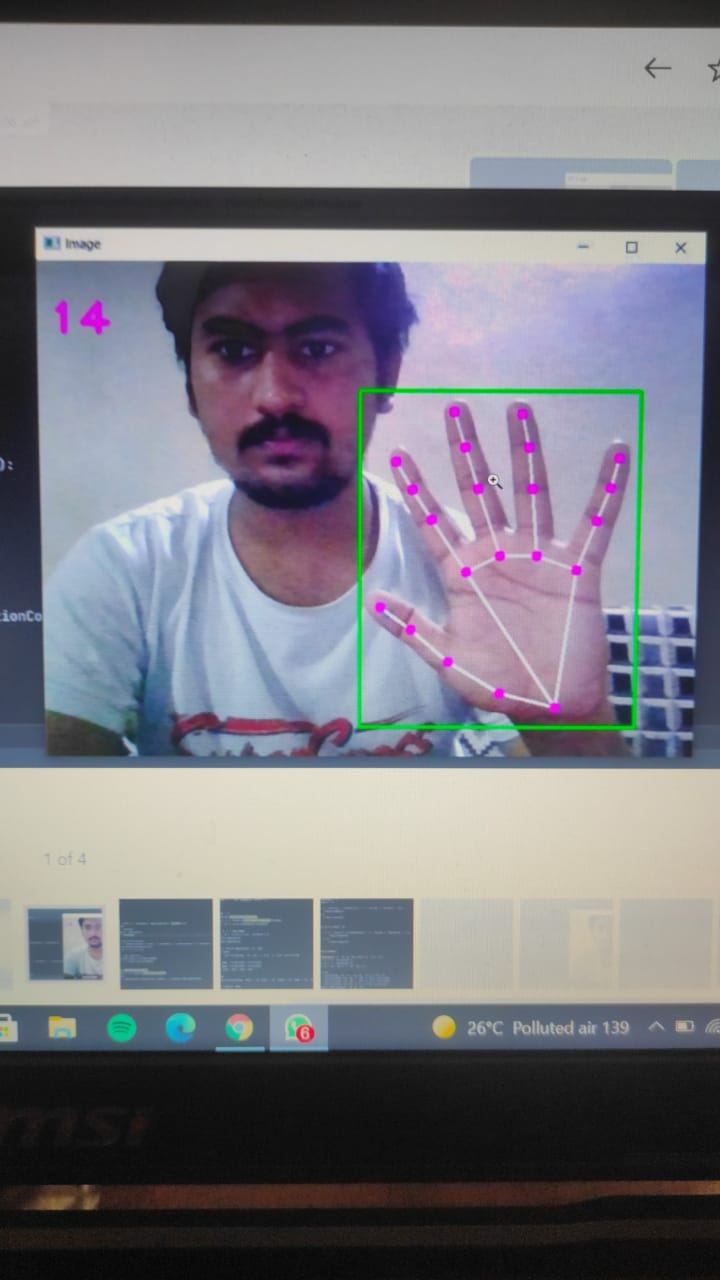
|  |  |  |  |
| --- | --- | --- | --- |
| Test scenario ID | Hand Tracking mouse 2 | Test case ID | Hand Tracking mouse |
| Test case description | Mouse using real time hand tracking | Test priority | High |
| Prerequisite | All the above mentioned software | Post- Requisite | NA |

Test Execution Steps:

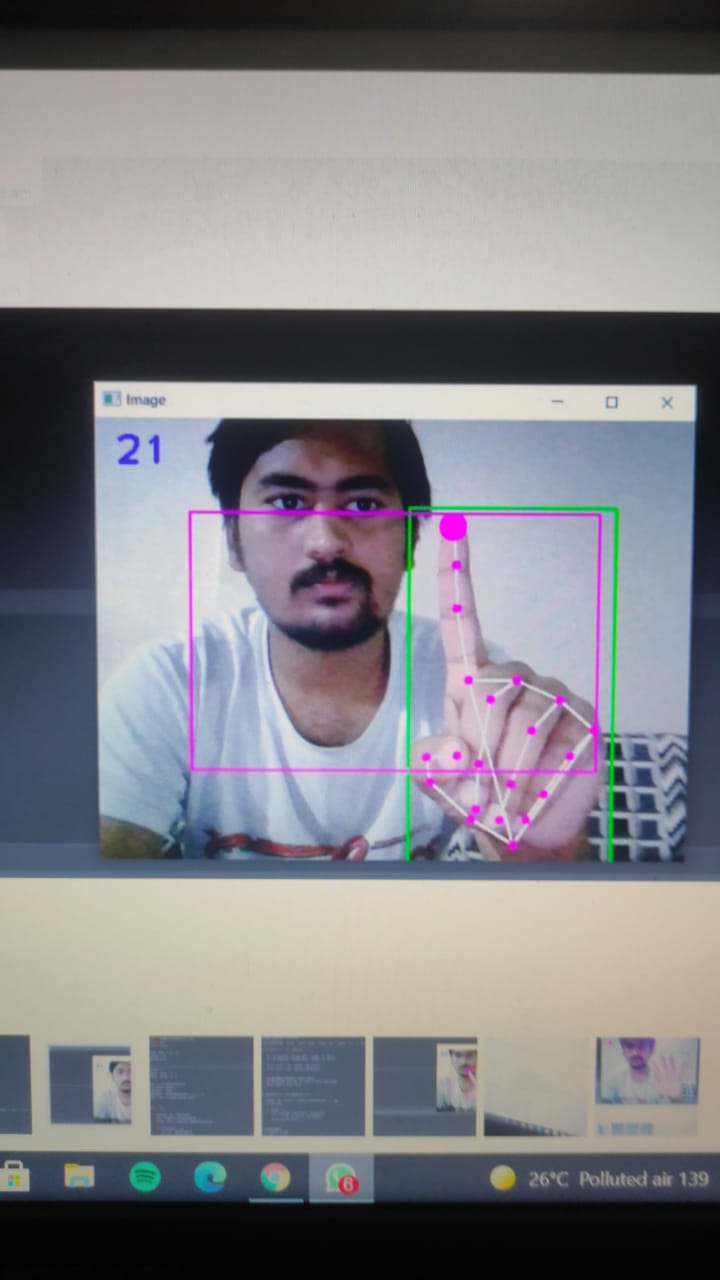
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.No | | Actions | Input | | Expected output | | Actual output | | | Test Browser | | Test Result | | Test Comments |
| 1. | | Launch Application | Run code in PyCharm | | Camera popped up | | Camera popped up | | | PyCharm | | Pass | | [Harshit 6/11/2021 09:14 AM]: Launch successful |
| 2. | | Moving index finger to control cursor | Index finger moving input | | cursor moving | | Cursor moving | | | PyCharm | | Pass | | [Harshit 6/11/2021 09:15 AM]: cursor movement successful |
| 3. | Using two finger to perform click operation of mouse | | | Joining Two fingers for performing click operation | | Click operation performed | | Click operation performed | PyCharm | | Pass | | [Harshit 6/11/2021 09:16 AM]: action perform successful | |

**5.1 Unit Testing:**

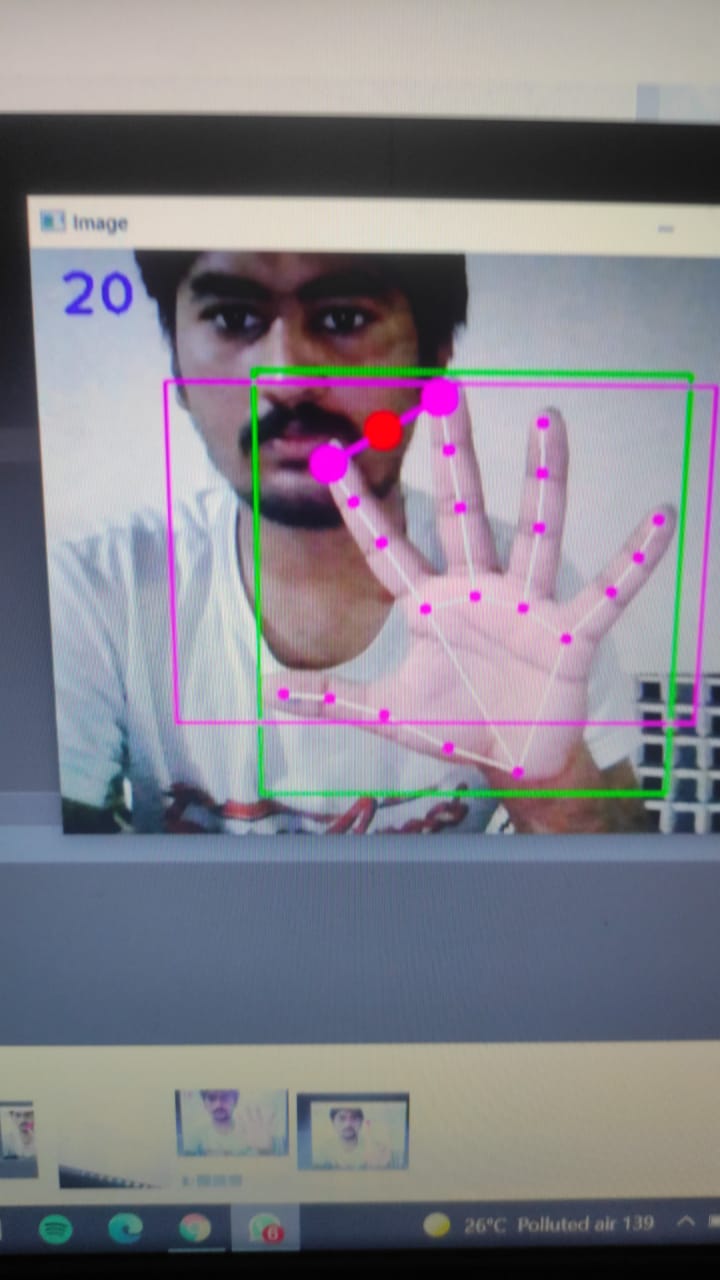
* In this testing, we tested individual modules against the corresponding specifications decided during the designing phase.
* Each module was tested for correct logic implementation, and for assurance of meeting the requirements, and the required changes were made.



**FIGURE5.1.1 CLICK**



**FIGURE5.1.2 CURSOR**



**FIGURE5.1.3 HAND DETECTION**

**5.2 Integration Testing:**

* Here all the modules, after unit testing, were combined into subsystems and then these subsystems were tested to check the correct integration and also to check, whether the subsystem is fulfilling its requirements.
* All the modules were successfully tested and proper integration was thus ensured.

**5.3 System Testing:**

* Here all the subsystems were properly integrated into complete software. And then whole software was tested for its correctness and to ensure that it meets all the requirements.
* The whole software passed these tests and required changes and modifications were made and the software is now ready for use.

**5.4 Validation Testing:**

* In this testing, we determined if the existing system complies with the system requirements and performs the dedicated functions for which it is designed along with meeting the goals and needs of the organization.
* All the dedicated function stated in the system requirement specification were available and working thus meeting the goals of the project.

**5.5 Performance Testing:**

* In this testing, we ensured software applications to perform properly under their expected workload.
* It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity and stability under a particular workload.

**5.5.1 Speed:**

* All the software applications were responding under 3 seconds after the request was made.

**5.5.2 Scability:**

* Software was able to handle and respond more than 10 simultaneous requests efficiently.

**5.5.3 Stability:**

* No major lag was noticed while uploading or fetching a large video or image file.

**5.5.4 Reliability:**

* Software function were found to be secure from any outside breach.

**Chapter-6**

**Result & Discussion**

The proposed model is capable of performing various mouse click events using hand gestures recognition. The main functionalities performed by this model include mouse movement, left mouse click event, double mouse click event, right mouse click event and folder dragging from one position to another. The user is allowed to use only green color paper which may vary in altered background.

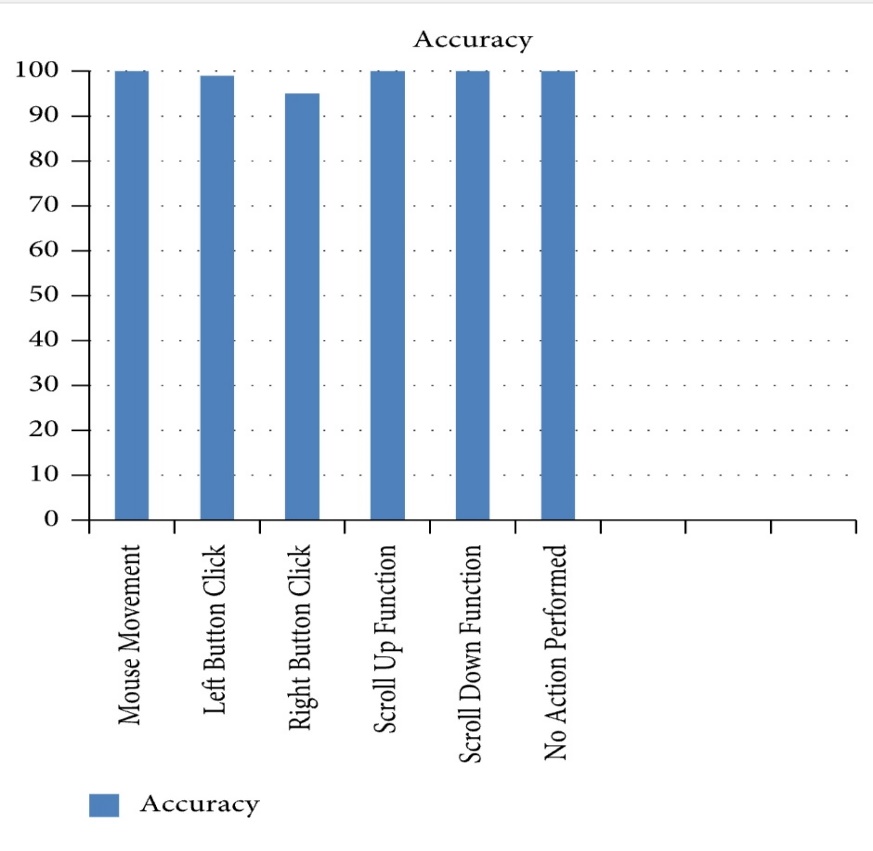
In the proposed virtual mouse system, the concept of advancing the human-computer interaction using computer vision is given.

Cross comparison of the testing of the virtual mouse system is difficult because only limited numbers of datasets are available. The hand gestures and fingertip detection have been tested in various illumination conditions and also been tested with different distances from the webcam for tracking of the hand gesture and hand tip detection. An experimental test has been conducted to summarize the results shown in Table 1. The test was performed 25 times by 4 persons resulting in 600 gestures with manual labelling, and this test has been made in different light conditions and at different distances from the screen, and each person tested the virtual mouse system 10 times in normal light conditions, 5 times in faint light conditions, 5 times in close distance from the webcam, and 5 times in long distance from the webcam, and the experimental results are tabulated in Table 1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Hand tip gesture | Mouse function performed | Success | Failure | Accuracy (%) | |  | | | | | | Tip ID 1 or both tip IDs 1 and 2 are up | Mouse movement | 100 | 0 | 100 | | Tip IDs 0 and 1 are up and the distance between the fingers is <30 | Left button click | 99 | 1 | 99 | | Tip IDs 1 and 2 are up and the distance between the fingers is <40 | Right button clicks | 95 | 5 | 95 | | Tip IDs 1 and 2 are up and the distance between the fingers is >40 and both fingers are moved up the page | Scroll up function | 100 | 0 | 100 | | Tip IDs 1 and 2 are up and the distance between the fingers is >40 and both fingers are moved down the page | Scroll down function | 100 | 0 | 100 | | All five tip IDs 0, 1, 2, 3, and 4 are up | No action performed | 100 | 0 | 100 | | Result |  | 594 | 6 | 99 | |  | | | | | |
| Fingertip ID for respective fingers: tip Id 0: thumb finger; tip Id 1: index finger; tip Id 2: middle finger; tip Id 3: ring finger; tip Id 4: little finger. |

.

From Table 1, it can be seen that the proposed virtual mouse system had achieved an accuracy of about 99%. From this 99% accuracy of the proposed virtual mouse system, we come to know that the system has performed well. As seen in Table, the accuracy is low for “Right Click” as this is the hardest gesture for the computer to understand. The accuracy for right click is low because the gesture used for performing the particular mouse function is harder. Also, the accuracy is very good and high for all the other gestures. Compared to previous approaches for virtual mouse, our model worked very well with 99% accuracy. The graph of accuracy is shown in Figure.



**Chapter-7**

**Summary & Conclusions**

Gesture recognition gives the best interaction between human and machine. Gesture recognition is also important for developing alternative human computer interaction modalities. It enables human to interface with machine in a more natural way. Gesture recognition can be used for many applications like sign language recognition for deaf and dumb people, robot control etc.

The Histogram-based and Explicitly threshold skin detection methods were evaluated and based on the results, the Histogram method was deemed as more accurate. The vision-based cursor control using hand gesture system was developed in the C++ language, using the OpenCV library. The system was able to control the movement of a Cursor by tracking the user's hand. Cursor functions were performed by using different hand gestures. The system has the potential of being a viable replacement for the computer mouse, however due to the constraints encountered; it cannot completely replace the computer mouse. The major constraint of the system is that it must be operated in a well-lit room. This is the main reason why the system cannot completely replace the computer mouse, since it is very common for computers to be used in outdoor environments with poor lighting condition. The accuracy of the hand gesture recognition could have been improved, if the Template Matching hand gesture recognition method was used with a machine learning classifier. This would have taken a lot longer to implement, but the accuracy of the gesture recognition could have been improved. It was very difficult to control the cursor for precise cursor movements, since the cursor was very unstable. The stability of the cursor control could have been improved if a Kalman filter was incorporated in the design. The Kalman filter also requires a considerable amount of time to implement and due to time constraints, it was not implemented. All of the operations which were intended to be performed using various gestures were completed with satisfactory results.

**Chapter-8**

**Future Scope**

We would improve the performance of the software especially hand tracking in the near future. And we also want to decrease the response time of the software for cursor movement so that it can completely be used to replace our conventional mouse. We are also planning to design a hardware implementation for the same in order to improve accuracy and increase the functionality to various domains such as a gaming controller or as a general-purpose computer controller. Other advanced implementation includes the hand gesture recognition stage to use the Template Matching method to distinguish the hand gestures. This method requires the use of a machine learning classifier, which takes a considerably long time to train develop. However, it would have allowed the use of lots more hand gestures which in turn would allow the use of more mouse functions such as zoom in and zoom out. Once the classifier is well trained, the accuracy of the Template Matching method is expected to be better than the method used in the proposed design. Another novel implementation of this technology would to use the computer to train the visually or hearing impaired.

#### **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. S.Mitra, and T. acharya, “Gesture recognition: A survey” Part C: Applications and reviews, vol. 37 (3), p.no. 311- 324,2007.
3. H.A Jalab,"Static hand Gesture recognition for human computer interaction”, p. no.1-7,2012
4. S. U. Dudhane, “Cursor control system using hand gesture recognition,” *IJARCCE*, vol. 2, no. 5, 2013.
5. K. P. Vinay, “Cursor control using hand gestures,” *International Journal of Critical Accounting*, vol. 0975–8887, 2016.
6. L. Thomas, “Virtual mouse using hand gesture,” *International Research Journal of Engineering and Technology (IRJET*, vol. 5, no. 4, 2018.