A Real-Time Human Computer Interaction using Hand Gestures in OpenCV

A PROJECT REPORT

Submitted by

BL.EN.U4AIE19028 K.Vishnu Sainadh

BL.EN.U4AIE19034 K.Satwik

BL.EN.U4AIE19066 V.Ashrith

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

DECEMBER – 2022

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, BANGALORE, 560035

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"A Real-Time Human Computer Interaction using Hand Gestures in OpenCV"** submitted by

BL.EN.U4AIE19028	K.Vishnu Sainadh
BL.EN.U4AIE19034	K.Satwik
BL.EN.U4AIE19066	V.Ashrith

in partial fulfillment of the requirements as part of **Bachelor of Technology** in "**COMPUTER SCIENCE AND ENGINEERING**" is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Bangalore.

Mr. Niranjan D K	Dr. Sriram Devanathan
Lecturer	Professor and Chairperson,
Dept. of CSE	Dept. of CSE

This project report was evaluated by us on ……

EXAMINER1 EXAMINER2

ACKNOWLEDGEMENTS

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people who made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranams at the lotus feet of **"AMMA," MATA AMRITANANDAMAYI DEVI** who showered her blessing upon us throughout the course of this project work.

We owe our gratitude to **Prof. Manoj P.**, Director, Amrita School of Engineering, Bangalore.

We thank **Dr. Sriram Devanathan,** Principal and Chairperson-CSE, Amrita School of Engineering, Bangalore for his support and inspiration.

It is a great pleasure to express our gratitude and indebtedness to our project guide **Mr. Niranjan D K,** Lecturer, Department of Computer Science and Engineering, Amrita School of Engineering, Bangalore for her/his valuable guidance, encouragement, moral support, and affection throughout the project work.

We would like to thank express our gratitude to project panel members for their suggestions, encouragement, and moral support during the process of project work and all faculty members for their academic support. Finally, we are forever grateful to our parents, who have loved, supported, and encouraged us in all our endeavors.

ABSTRACT

In today's world, with the development of new and emerging technologies and devices during the past ten years, the trends in human-computer interaction have undergone a significant transformation. Human-Computer Interaction (HCI) is the way that people and computers talk to each other. Generally, computers have been used with a mouse and a keyboard. Using hand gestures to communicate with computers is a new way to interact with computers. Computer Vision techniques will be used in our project to create a virtual mouse and keyboard that can read hand gestures.

People with different disabilities cannot use the computer with the way it is set up now. Our project aims to solve this problem. Our project is a GUI based application, and with the assistance of computer vision techniques, user hand motions (gestures) is implemented in real-time to operate mouse and keyboard of our computer and custom gestures are being used to make certain actions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER - 1
INTRODUCTION

1.1 INTRODUCTION TO HUMAN COMPUTER INTERACTION

A computer mouse detects two-dimensional surface movements. This movement controls the Graphical User Interface (GUI) on a computer. There are many varieties of mice in modern technology. One uses a firm rubber ball to identify movement. Years later, the optical mouse replaced the hard rubber ball with an LED sensor that detects table top movement and delivers the data to the computer. Laser mice were created in 2004 to increase precision with the tiniest hand movement and to overcome the optical mouse's inability to follow high-gloss surfaces. Despite its accuracy, the mouse has physical and technical limits. A computer mouse is a consumable hardware device since it must be replaced if its buttons degrade and create improper clicks or the computer no longer detects it.

Despite restrictions, computer technology and human-computer interactions evolve. Since the emergence of a mobile device with touch screen technology, the world wants the same technology on all devices, even desktops. Touch screen technology for desktops exists, but it's expensive.

As we see in Fig 1.1, the computer is being controlled by the user using gestures which are taken as input to the webcam.

An alternative to the touch screen might be a virtual HCI device. This would include using a webcam or some other device that captures images in place of a physical mouse or keyboard. This device, known as the camera, will be utilised by software that monitors the actions of the user in order to process those motions and convert them to mouse-like pointer motions.

Fig 1.1.1: Using webcam to control the mouse

In computer vision, static gestures which can be seen in the figure 1.1.2 refer to gestures that do not involve movement, such as a person holding up a peace sign or flashing a "V for victory" sign with their fingers. These gestures can be captured and analyzed using still images or video frames.

Fig 1.1.2: Static Gesture

Dynamic gestures, on the other hand, involve movement and can be captured and analyzed using video or other time-based media. Examples of dynamic gestures include waving, pointing, or making a sweeping motion with the hand. As shown in figure 1.1.3.

Fig 1.1.3: Dynamic Gesture

Both static and dynamic gestures can be used in computer vision systems to enable human-computer interaction, such as in virtual reality applications or gesture-based user interfaces. Techniques for detecting and recognizing gestures in computer vision systems typically involve using machine learning algorithms to analyze visual data and extract relevant features, such as the shape, position, and movement of the hands or other body parts.

Dynamic gestures are hand gestures that involve movement or change over time, such as waving, pointing, or drawing a shape in the air. Dynamic gestures can be used to communicate or convey meaning, such as indicating a direction or signaling a command. Dynamic gestures are typically distinguished from static gestures, which are hand gestures that involve a fixed pose or position, such as showing a peace sign or holding up a number of fingers.

In the context of computer vision and gesture recognition, dynamic gestures can be more challenging to recognize and classify accurately due to the complexity and variability of the hand movement and the potential for occlusions and other distractions. To improve the accuracy of dynamic gesture recognition, it may be necessary to use specialized hardware, such as a depth sensor or a glove with sensors, to capture more detailed information about the hand and finger movements.

1.2 MOTIVATION

People are moving toward a way of life in which all technological devices can be controlled and interacted with remotely, without the need for extra devices like remotes, keyboards, etc., so it's safe to say that the virtual mouse will soon replace the physical mouse. It is not only convenient, but it also saves money.

It is well known that in order to interact with a computer system, users must utilize a real, physical mouse. This mouse requires a specific area of surface to work, in addition to having cable length restrictions. Virtual Mouse doesn't need any of it; all it needs is a webcam to capture images of the user's hand position in order to figure out where the user wants the pointers to be.

1.3 PROBLEM STATEMENT

Problem: Current methods of controlling a computer mouse and keyboard, such as using a physical mouse and keyboard or touchscreens, can be cumbersome and require a fixed and stationary posture. This can be uncomfortable and inefficient, especially for users with disabilities or mobility impairments, or for users who prefer a more natural and intuitive way of interacting with computers.

Goal: To create a real-time virtually controlled mouse and keyboard using hand gestures that enables users to control a computer in a more natural and intuitive way, while achieving the ease and naturalness desired for human-computer interaction (HCI).

1.4 OBJECTIVE

The goal of this project is to make a programme with a virtual mouse and keyboard that focuses on a few important areas of growth. The first goal of this project is to get rid of the need for a hardware mouse and keyboard and let the computer system be controlled by a camera.

We can see in Fig 1.3.1, that the user is controlling the virtual keyboard using hand.

Fig 1.3.1: Virtual Keyboard

CHAPTER - 2

LITERATURE SURVEY

2.1 TRACKING AND RECOGNITION OF HAND GESTURES

Introduction:

The most common method of human-computer interaction (HCI) used in the past few decades is based on the typical input device, such as a keyboard and mouse. This method is effective, but the underlying flaw in it is that the results are unnatural and do not include a direct physical connection with the computer. Despite its effectiveness, the method has this essential flaw. This limitation is brought into sharper relief by the introduction of innovative display technologies such as virtual reality. As a result, just recently, a few innovative strategies have been created in order to eliminate the HCI bottleneck. This issue has received a lot of attention from researchers, and a lot of potential solutions have been identified. Because humans typically use their hands while interacting with their surroundings, the strategy known as gestural interaction stands out among the other ways. Because of this, consumers are provided with an experience that is authentic and immersive.

Methodology:

Every time, the suggested HCI system takes a picture using the depth camera. Following preprocessing, a hand detector is used to extract the hand area. The hand is tracked using the kalman filter, and motions are identified using apparent gestural properties. What should be said in response is decided by the recognized findings. In order to obtain the foreground depth picture, background removal must first be done on the depth image of an item obtained from the Kinect. then a hand area is extracted using a skin colour model for European TV transmission compression. then depth data is combined with the kalman filter. The system categorises the motions based on the characteristics.

Conclusion:

A system for human-computer interaction has been presented in this study. They've put in place an interactive framework that enables the player to manipulate the virtual item with just their hands. The suggested technology offers users a more immersive and natural experience than

traditional computer input methods. The system's capability for just static gestures is one of its limitations.

2.2 A VIRTUAL MOUSE AND KEYBOARD BASED ON GESTURES

Introduction:

The main point of this paper is to show how hand gesture recognition and image processing can be used to make a virtual mouse and keyboard. This will let the user move the mouse pointer with hand gestures and use keyboard functions that are set up for the user's needs. Getting the cost of hardware to go down. Some things will be harder and some will be easier to do with their method, like making 3D models and looking at the imagined parts of a patient's body while they are being operated on. One of the best things about it is that you don't have to touch anything for it to work in architectural designs and automated buildings.

Methodology:

The camera starts up and looks for anything that has a colour close to that of human skin. The captured video is then compared to the module and the code. If the code says that the object shown by the computer camera looks like a human hand, the process moves on to the next step, which is called "Convex Hull detection." The Convex hull process then checks to see if the object shown is a hand. If it is, it counts the number of flaws. After that, the centroid is found, and the gesture is classified as either a keyboard or mouse gesture based on how it matches up with the algorithm functions. All the process is shown in figure 2.2.1.

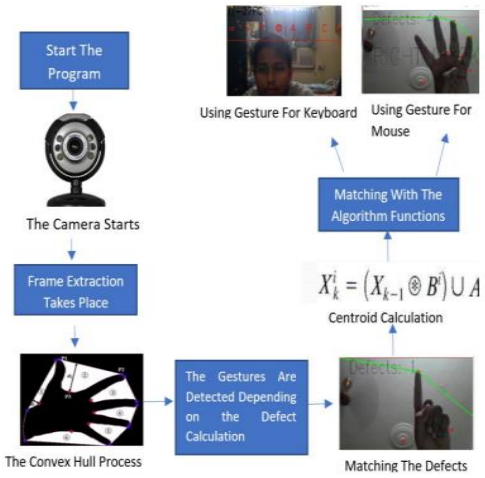

Fig 2.2.1: Flow Diagram

Conclusion:

This paper suggests a system that can read hand gestures and take the place of the mouse and keyboard. This includes moving the mouse cursor and using the keyboard to drag and click, print letters, and do other things with the keyboard. The skin segmentation process is used to separate the color/image of the hand from its background. Taking the whole body into the camera is a problem that can be solved by the "remove arm" method. Overall, the proposed algorithm can detect and recognise hand gestures so that it can control mouse and keyboard functions and also make a real-world user interface.

2.3 USING HAND GESTURES TO CONTROL MEDIA PLAYER

Introduction:

Technology has been changing a lot at the same time that society has been changing. Since they were first made, computers have grown and changed a lot over the years. HCI, or human-computer interaction, is the key to making interactive systems work well. It involves putting together knowledge of how people work and knowledge of how hardware and software technologies work. This paper is about how to use CNN hand gestures to control the media player. The proposed system is a web app that uses the device's camera to let users control any media player app without touching it or using a remote. This can be done without any special hardware.

Methodology:

The image data that is sent in is turned into a video by a webcam. In the video, there are also frames. OpenCV turns these frames into black-and-white pictures, which are then saved in certain directories. A CNN model is made with the help of Keras libraries. The ImageDataGenerator class is used to prepare the images in the directories so that only the parts that are needed can be taken out. To teach the model, images from the train dataset are used. The model is put together, and the test is checked to see how well it works. Keras libraries are used to save and load models and put gestures into one of several classes. Using the Pyautogui library, each gesture was linked to a function for controlling the programme. It is done with the prediction in charge. You can deploy a web app with all of the project files using Streamlit sharing.

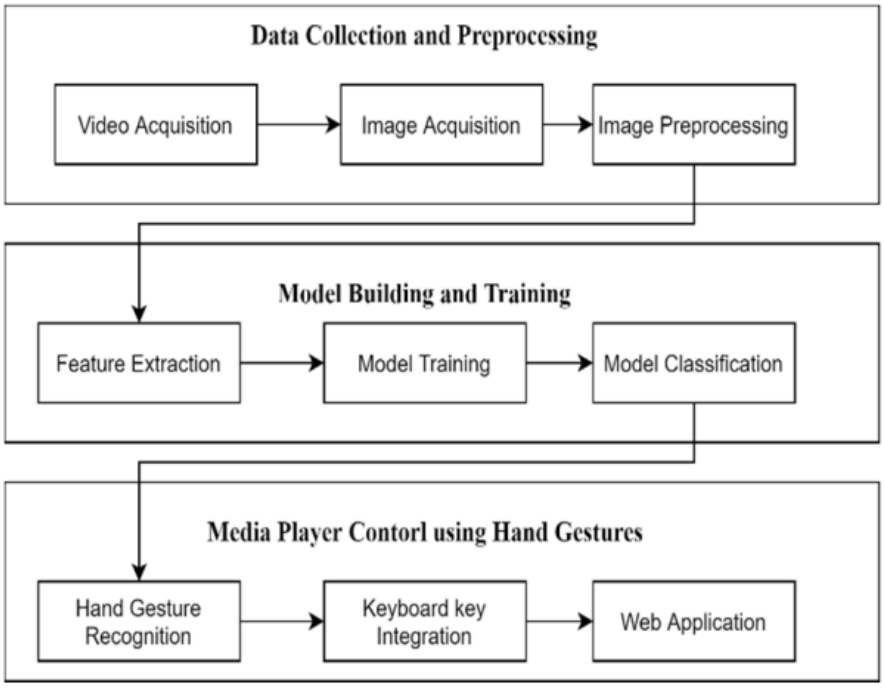

Fig 2.3.1: System design workflow

Conclusion:

The work that was done to create a hand gesture recognition system that can be used to operate the media player is detailed in this paper. OpenCV is used to take the photographs, a two-dimensional convolutional neural network is used to extract features and forecast movements, and PyAutoGUI is used to control the keyboard keys whenever a gesture is anticipated to go along with them. A specialised dataset consisting of seven different gestures is compiled in order to put their model to the test. In order to evaluate how successfully the suggested system operates, this model is also evaluated in real time using the aforementioned hand gestures. The proposed CNN model worked exceptionally well, with an accuracy of 98%.

2.4 RECOGNITION OF HAND TYPING USING CNN

Introduction:

The key component of a human interaction access point that initiates the beginning, running, and regulating activities of a possible system is the text entry user interface. Text entry user interfaces on conventional computer systems, however, nevertheless suffer from bottlenecks like the potential for breakage or infection transmission in public settings. A virtual keyboard is a projection of a full-size QWERTY keyboard onto any surface. Any key's picture can be touched

to provide the input for the system matching to that key's image. They looked at the common QWERTY keyboard in this essay.

Methodology:

The RGB picture must first be converted to a greyscale version before obtaining the difference map from the two photos that follow. and last, to combine all the difference maps to create a final difference map. It is difficult to recognise a button-clicking gesture with just one RGB camera. In this study, we developed a rudimentary CNN-based typing action detection system. CNN divides its output into touch and non-touch categories. The 5,180 data gathered by 12 individuals make up the dataset used in this study. CNN receives a grayscale picture as its input, and its output label is either 0 or 1, corresponding to a touch or no-touch action.

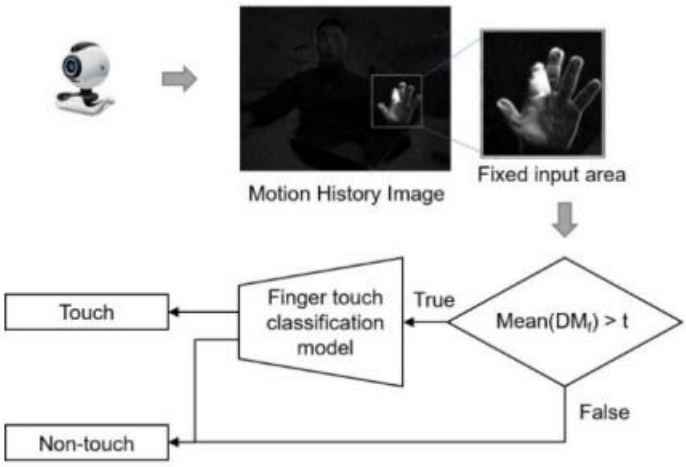

Fig 2.4.1: Proposed Architecture

Conclusion:

Using a single RGB camera, they showed in this study an effective hand typing motion identification system for Virtual Keyboard. The experienced group's top score in real-time testing is 92%. The score is 67% in the group of unexperienced people. Because the inexperienced group was attempting to type in the air for the first time, there was a significant discrepancy between the two scores. Push clicking gestures may be implemented in the future to enhance real-time functionality and apply to mobile devices.

2.5 AMBIDEXTROUS VIRTUAL KEYBOARD

Introduction:

Major firms have recently developed various commercial gadgets that offer AR/VR experience, which has piqued the public's interest in this emerging technology. These AR/VR gadgets often take orders via hand gesture detection. Considering this, research and development into a virtual keyboard interface based on hand gesture recognition have been conducted extensively and with great success utilising several types of sensors. This method is preferable to the one we just saw because it calls for nothing more than a cheap camera.

Methodology:

In order to boost the typing speed, a novel keyboard layout is presented in this study. The first approach is to employ motions that simultaneously activate the keyboard layer and the typing action. Choosing a layer now takes much less time due to this method.

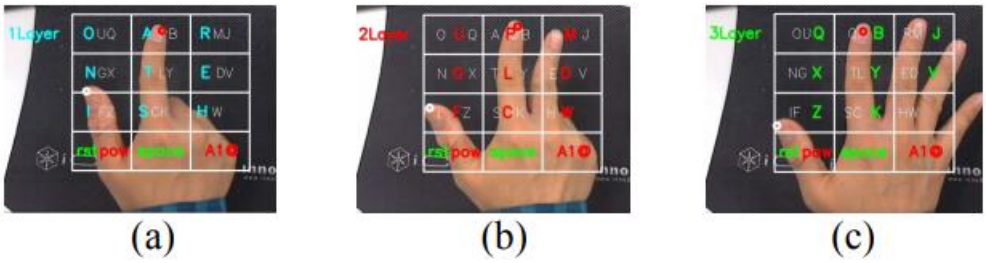

Fig 2.5.1: Layer-aware gestures (a) 1st layer (b) 2nd layer (c) 3rd layer

This method sets up new hand gestures that can be used to select the keyboard layer. Figure 2 shows the new hand moves. Figures 2.5.1(a), b), and c show the gestures for typing in the first, second, and third layers, respectively. In the first step of the typing gesture for the first layer in Figure 2.5.1 (a), the index finger is placed on the character. The second step is typing in, which is done by putting the thumb on the index finger. In Fig. 2.5.1 (b), both the index and middle fingers are stretched out to make the selection gesture. The type-in gesture is when you touch your thumb to your index finger while keeping your index finger and middle finger stretched out. When the contact is made, the character that the index finger points to is typed in. Figure 2.5.1 (c) shows the way to select the third layer by putting all your fingers out. Again, the touch of the thumb to the index finger, which points to the character to be typed, is the type-in gesture.

Conclusion:

This paper presents a novel design for a virtual keyboard that incorporates finger gesture recognition to speed up input by making use of the user's natural range of motion. Input speed is increased by 56.49 percent using the suggested keyboard design with new type-in actions, according to simulation data.

2.6 DEEP LEARNING BASED GESTURE RECOGNITION FOR SYSTEM APPLICATIONS

Introduction:

Due to its usefulness and the ease with which it allows humans to connect with machines, a hand gesture recognition system is a vital tool in the development of user-friendly interfaces. The proposed method is based on hand movement recognition since it is easy to implement and does not necessitate an intermediate medium. The current technique for accessing applications is rigid and time-consuming for those who are blind or have physical impairments in their hands.

Methodology:

The data set includes information on a total of 20 different individuals (10 men and 10 women). Several photographs were taken against a wide variety of backdrops. For the best possible classification performance, the ratio of the area of the feature map to the area that contains hand movements is carefully maintained at an appropriate level. They trained a Deep Convolutional Neural Network with a batch size of 64 and an epoch of 100 under these circumstances. The multi-convolutional layers that came before the max-pooling layer helped to extract appropriate features for classification, which is why the model is now able to recognise hand motions.

The deep convolutional neural network that has been proposed has a total of nine convolutional layers. These layers include four pooling layers and three fully connected layers, and they are all interlaced along the Rectified Linear Unit as well as other dropout layers. The output of the first layer, which consists of 128 neurons and is fully connected, is fed into a second layer that consists of 128 neurons and is also fully connected. The mapping from the intermediate classes to the ultimate classes for hand motion detection is then determined with the use of a third completely

connected layer. The final layer is a soft-max layer, which is specialised to receive the output of fully linked layers. It works by giving a probability to each class in the training set. The recognition is achieved by first classifying the input image in a stable state with a high probability. This is done in order to produce the recognition.

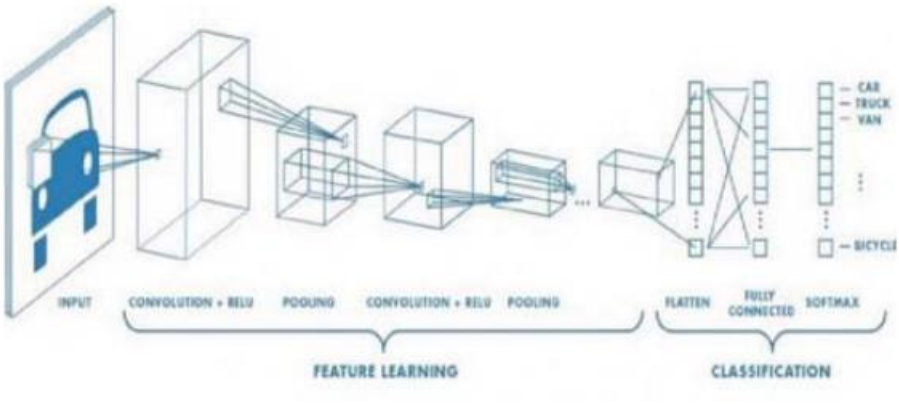

Fig 2.6.1: Proposed DCNN Architecture

Testing begins with frame detection, which involves selecting still images from a movie to be further processed. The picture in this frame is now in greyscale. Having determined the gesture class from the input, the trained model file is then used to do a class comparison.

Conclusion:

These devices are helpful for those who are paralysed in addition to those who are visually impaired. The suggested system would be able to recognise the hand gesture and accurately run the application of the system without the need for any external power needs. As a result, the requirement for a sensor to detect the gesture would be eliminated. In front of the webcam, users make gestures, which are then read, processed, and used to control the software running on the device. They presented a deep convolutional neural network (DCNN) in this paper to employ hand gesture recognition and quickly classify them by maintaining even the not-hand area without any detection or segmentation step. This was accomplished by recognising the hand movements. The constructed model has produced data that show its accuracy to be 92%, which is its top average accuracy with the least amount of test loss.

2.7 IMPROVED DYNAMIC TIME WARPING ALGORITHM FOR DYNAMIC HAND GESTURES RECOGNITION

Introduction:

Data gloves and computer vision-based techniques are the two primary categories into which research on gesture recognition can often be divided. In order to identify the position and contour of hand gestures, data gloves make use of specialised hardware components that include sensors into their designs. This paper presents an integrated model for the recognition of dynamic hand gestures based on an improved version of the Dynamic Time Warping (DTW) algorithm. This model has a significant impact on the effectiveness of dynamic trajectory analysis and was developed with the goal of recognising dynamic hand gestures in a way that is both efficient and intelligent.

Methodology:

In order to test and assess the suggested method for hand gesture recognition, this study makes use of the ChaLearn Gesture Data (CGD) dataset that was provided as part of the ChaLearn Gesture Recognition Challenge. Spread, Bumping, Down, Shaking, Zooming, and Holding are the six different dynamic movements that make up CGD. Each motion comes with one video template and twenty more videos to go along with it. First, the gesture motion area is extracted from three consecutive video frames, and then the Hu-moment for motion position centre of mass is used to define the trajectory of the gesture. In the final step, a comparison of the similarities between different gesture trajectories is made using an improved version of the DTW algorithm. DTW is able to evaluate the degree to which two time series are comparable to one another. Template construction and matchmaking make up the two processes that make up the DTW algorithm. The beginning and ending locations of a trajectory can be determined, in both stages, with the use of an algorithm called the endpoint algorithm.

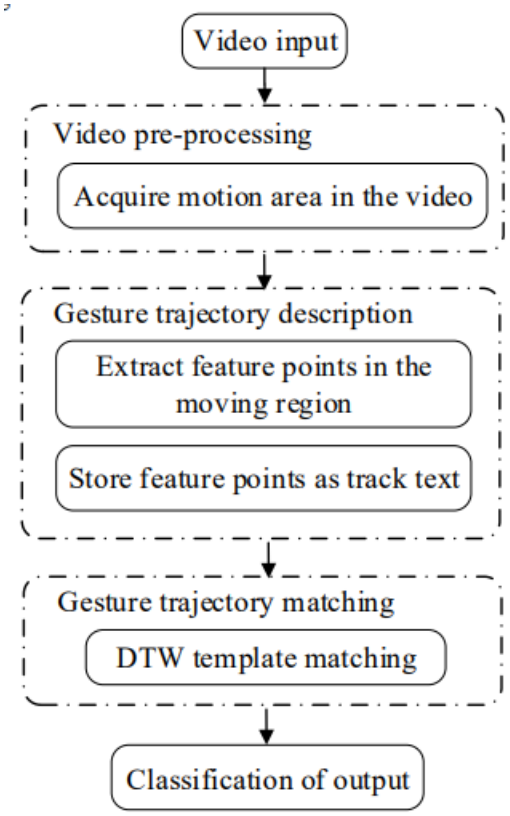

Fig 2.7.1: Dynamic gesture recognition model flowchart

Conclusion:

Even though numerous models for dynamic gesture detection have been presented, their efficiency and recognition rate have only marginally improved. To address this issue, the model described in this research extracts the dynamic gesture motion areas using an updated DTW (Dynamic Time Warping) method, they created a model for the recognition of dynamic hand gestures. Benchmarking trials on the identification of the six typical dynamic gestures produced satisfactory categorization outcomes.

S.No.	Gaps
1	Only static gestures are supported, restrictions on the multi-hand tracking and recognition, Custom gestures mapped to certain actions (shortcuts).
2	Mouse clicks are not intuitive, Custom gestures mapped to certain actions (shortcuts).
3	Only works for a specific application, Custom gestures mapped to certain actions (shortcuts).
4	Virtual mouse is absent, Custom gestures mapped to certain actions (shortcuts).
5	The keyboard pattern is frequently changed, Custom gestures mapped to certain actions (shortcuts).
6	Only works for a specific application, Custom gestures mapped to certain actions (shortcuts).
7	Only Dynamic gestures are supported, Custom gestures mapped to certain actions (shortcuts).

Table 2.1: Summary of Gaps

2.8 SUMMARY

In most of the papers they either used static gestures or dynamic gestures. Both are not being implemented together. The number of gestures is being limited due to the way they have been implemented i.e.; is they have provided limited number of gestures to the user because they have less gesture space. In our case we can change the actions mapped to the gestures and increase the number of gestures because in the way we have implemented we have large gesture space.

CHAPTER – 3

REQUIREMENTS AND ANALYSIS

3.1 SOFTWARE REQUIREMENTS

Python:

Python is a programming language that can be used for a wide range of purposes. It was designed by Guido van Rossum and released in 1991. One of the key features of Python is its emphasis on code readability, which is achieved through the use of whitespace and clear syntax. Python provides a variety of tools and features that allow developers to write efficient and effective code, both for small and large projects.

PyCharm:

PyCharm is an integrated development environment (IDE) for Python, developed by JetBrains. It provides a range of features to help you develop Python code more efficiently, such as code completion, error highlighting, and refactoring tools. PyCharm also includes a debugger and a testing tool, which allows you to find and fix errors in your code, and run and manage unit tests. Additionally, PyCharm integrates with version control systems like Git and includes support for various frameworks and libraries commonly used in Python development, such as Django and Flask.

OpenCV:

OpenCV (Open-Source Computer Vision) is a free and open-source library of computer vision and machine learning algorithms designed to help developers build applications that can analyze, understand, and manipulate visual data. It is written in C++ and can be used in various programming languages through bindings, such as Python, C#, and Java.

OpenCV provides a wide range of features for image and video processing, including basic image manipulation and processing (resizing, cropping, color space conversions), feature detection and extraction (edges, corners, blob detection), object tracking, and machine learning techniques for classification and clustering

OpenCV can be used in a variety of applications, such as robotics, security, medical imaging, and scientific research. It is also commonly used in fields such as computer vision, image processing, and artificial intelligence.

Windows 11:

Microsoft's Windows OS (Operating System) family of software runs on a range of gadgets, including desktops, laptops, tablets, and smartphones. The newest version is Windows 10, which was made available in July 2015. A graphical user interface, a web browser, a collection of apps, and support for hardware like keyboards, mouse, and gaming controllers are all included in the Windows OS. Additionally, the OS comes with several essential functions including networking, security, and system administration.

3.2 HARDWARE REQUIREMENTS

Camera:

Microsoft's Windows OS (Operating System) family of software runs on a range of gadgets, including desktops, laptops, tablets, and smartphones. The newest version is Windows 10, which was made available in July 2015. A graphical user interface, a web browser, a collection of apps, and support for hardware like keyboards, mouse, and gaming controllers are all included in the Windows OS. Additionally, the OS comes with a few essential functions including networking, security, and system administration.

CHAPTER – 4

SYSTEM DESIGN

4.1 HIGH LEVEL DESIGN

The program through webcam acquires frames of images of the user for recognizing the gestures and performs the actions to which the gesture is defined to generate. This process can be seen in the Figure 4.1.1

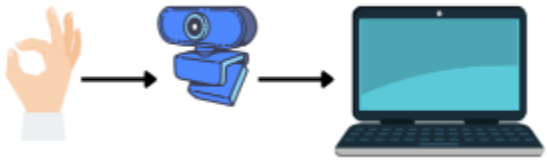

Fig 4.1.1: High Level Architecture

Initially when the application starts, the first thing that happens is it acquires an image from the camera and then the image is processed and the person's hand in the image will be recognized or detected. If the person is showing the custom gesture that we have defined then the action mapped to that gesture will be performed. Like one of the gestures is that if only the index finger is shown by the person, then the mouse of the computer can be controlled. This image acquisition is performed continuously and when the gesture is changed to another custom gesture hen the action performed will change. This process will continue till we end the application.

4.2 LOW LEVEL DESIGN

Using this project as application a user is able to interact with the computer without physically touching it by controlling it virtually through hand gestures by using the camera to recognize the number of hands the person is using if user is using 2 hands the application changes its mode to keyboard mode and allowing the user to use keyboard virtually using his index finger to choose between keys and to press a key on the virtual keyboard and if the person is using 1 hand then the application checks for gestures and performs operations as of a traditional mouse based on the gesture recognized after any gesture is been recognized the particular action is generated using the middle ware software interface we designed to control the system virtually. The gestures which are made by user using his hand or hands are detected using mediapipe hand detector which returns the landmarks of the hands recognized through which pre-defined gestures are matched to check if any gesture is made by user. The gesture if recognized then the action to which that gesture is mapped is performed and the image acquisition goes on till the user ends the program or application and checks for further gestures and this whole process of gesture recognition goes on till the user ends the program. A detailed flow chart is shown in figure 4.2.1

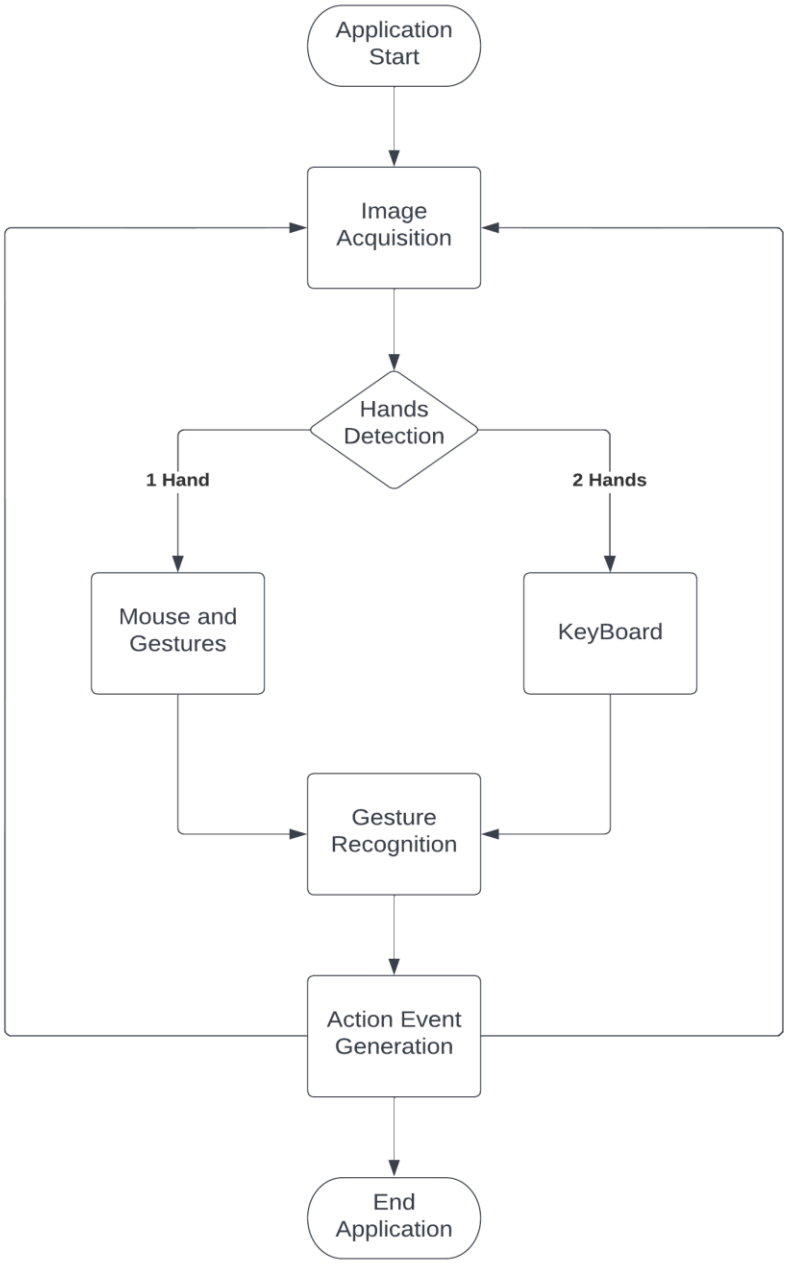

Fig 4.2.1: Low Level Architecture

CHAPTER – 5

SYSTEM IMPLEMENTATION

5.1 MODULES WITH DESCRIPTION

OS:

Python's OS module offers a set of functions that may be used to communicate with various operating systems. The OS module is categorized with Python's other basic utility modules. This module lets you use features that depend on the operating system even when you're not on that system. The os and os.path modules both include a large number of functions that may be used to interface with the file system.

Time:

The Python time module gives you many ways to show time in code, such as with objects, integers, and texts, among other things. It also includes features that is not related to the representation of time, such as waiting while code is being executed and assessing the efficiency of your code.

Datetime:

Although date and time do not constitute their own data type in Python, a module known as datetime may be loaded to facilitate the manipulation of both date and time information. There is no need to install the Python Datetime module from an external source because it is already a part of Python. Datetime is a module for Python that provides classes for working with dates and times.

Autopy:

Python's AutoPy is a user interface (GUI) automation module that is straightforward and compatible with several platforms. It has features for managing the keyboard and mouse, locating colors and bitmaps on the screen, and showing alarms, among other things.

Cv2:

OpenCV is a huge open-source library for machine learning, image processing, and computer vision. It is a big part of real-time operation, which is very important in systems today. This is

because systems today need to be able to work in real time. It can be used to analyse photos and videos to recognise things, faces, and even a person's handwriting. When paired with different libraries like NumPy, Python can analyse the OpenCV array structure. When Python is used, this ability can be used. We use vector space and do math on different parts of an image pattern to figure out what the image pattern is and all the parts that make it up. OpenCV can be used to solve a wide range of problems, such as object detection, face recognition, medical image analysis, etc.

Cvzone:

CVzone is a computer vision package that makes it simple for us to carry out tasks such as face identification, hand tracking, position estimation, and many more. It also does image processing and other AI-related operations. OpenCV and MediaPipe are two of the libraries that are utilised centrally.

Numpy:

Numerical Python, more commonly referred to as NumPy, is a library that comprises of multidimensional array objects as well as a set of functions for processing such arrays. NumPy is also referred to by its other name, Numerical Python. Numerical Python is what is meant to be abbreviated as NumPy. NumPy is a library for the Python programming language that gives users the ability to conduct logical and mathematical operations on arrays. In addition to that, it is equipped with tools for performing operations that are associated with linear algebra, the transform of Fourier, and matrices. According to the goal of the project, the array object that NumPy provides should be up to 50 times faster than the ordinary list data type that Python provides.

MediaPipe:

MediaPipe delivers platform-agnostic ML solutions for live and streaming video. MediaPipe Hands tracks hands and fingers accurately. It uses ML to determine 21 3D hand landmarks from a single image. MediaPipe Hands employ an ML pipeline with many cooperating models. An image-wide palm identification model that returns the hand's bounding box. A hand landmark model that uses the palm detector's cut region to generate accurate 3D hand keypoints. Providing a reduced hand image to the hand landmark model reduces the need for data augmentation (rotations, translations, and scaling) and allows the network to focus on coordinate prediction

precision. The hand landmark model localises 21 3D hand-knuckle coordinates through regression, which is direct coordinate prediction.

PyAutoGui:

PyAutoGUI gives Python scripts the ability to take control of the keyboard and mouse in order to automate user interactions with other apps. The application programming interface was developed with ease of use in mind. PyAutoGUI is compatible with Windows, macOS, and Linux, and it can run on both Python 2 and Python 3. PyAutoGUI is equipped with a number of functions, some of which include the ability to move the mouse and click in the windows of other apps, Sending or transmitting keystrokes to apps (for example, to fill out forms), Take screen captures, and if you're given an image of something (such a button or a checkbox, for instance), locate it on the screen, Find the window of a programme, and then relocate it, resize it, expand it, reduce it, or close it (Windows-only, currently), displaying alert and message boxes.

Screen_brightness_controller:

We are utilising the screen-brightness-control library in order to exercise control over the brightness of the screen. The screen-brightness-control library only has a handful of methods and a single class. The capabilities that are most helpful are the ones that allow you to get the brightness, change the brightness, fade the brightness to a given degree, and list all of the displays that are now accessible.

Appopener:

AppOpener is a module for PYPI that allows users to open any application without needing to know the program's absolute path. The functionality of the module relies on the usage of the App name and App Id. AppOpener is path agnostic, which means that it is not restricted in any way by the fact that we explicitly supply the application's path. It encompasses each and every programme that has been installed in the Windows operating system. AppOpener comes with a plethora of functions that might be of value. Because of these qualities, the module has increased levels of efficiency and productivity.

Pynput:

You can exert control over, as well as monitor and listen to, your input devices, such as your keyboard and mouse, with the help of the pynput library.

While the pynput.mouse module gives you the ability to operate and monitor the mouse, the pynput.keyboard module gives you the ability to do the same with the keyboard.

5.2 ACTIONS MAPPED TO GESTURES

No action will be performed if the user shows all the fingers which is referred to as neutral in our terminology.

To control the mouse the user needs to go from showing all the five fingers which is neutral mode to showing only index finger.

To use the virtual keyboard the user needs to show all the ten fingers which is two hands to the camera and if the user the keeps same gesture for 15 frames longer then the keyboard mode will be exited. In the keyboard mode to press a key the user needs to use index finger and middle finger which is ‘v’ gesture (symbol) as shown in figure 5.2.1. The keys in keyboard will get highlighted if the ‘v’ is placed on that key and if the distance between these two fingers is decreased then the key will be pressed. The virtual keyboard has all the English alphabets along with backspace, caps lock, space and enter.

Fig 5.2.1: V-Symbol

To perform left click using right hand the user needs to go from neutral mode to ‘v’ symbol which is opening index and middle finger and then if we close index finger then the left click is performed. For right click, middle finger should be closed from ‘v’ mode. And for the left double click the user needs to make the index finger and the middle finger come close and reduce distance between them. Similarly, if the user uses left hand, then for left click the user needs to go from neutral mode to ‘v’ symbol which is opening index and middle finger and then if we close middle finger then the left click is performed. For right click, index finger should be closed from ‘v’ symbol. Left double click action using left hand is same as left double click using right hand.

To take screenshot using gesture, the user needs to go from neutral mode to closing the fist (putting thumb inside the fist).
To drag and select files the user needs to go from ‘v’ symbol to closing the index and middle finger then the user can drag and select multiple files.

To control scroll up, down, left and right user needs to move right hand from neutral mode to opening only index, middle and ring finger and closing the pinky finger and thumb as shown in the figure 5.2.2. If the user moves this gesture vertically upwards then scroll up will be performed, and if the user moves this gesture vertically downwards then scroll down will be performed, and if the user moves this gesture horizontally left then scroll left will be performed, and if the user moves this gesture horizontally right then scroll right will be performed.

Fig 5.2.2: Three Finger Gesture

To control volume and brightness of the computer the user needs to move left hand from neutral mode to opening only index, middle and ring finger and closing the pinky finger and thumb same as above. In the same way as above, here the volume is controlled using horizontal movements and brightness is controlled using vertical movements. That is if the user moves this gesture vertically upwards then brightness will be increased, and if the user moves this gesture vertically downwards then brightness will be decreased, and if the user moves this gesture horizontally left then volume will be increased, and if the user moves this gesture horizontally right then volume will be decreased.

To open chrome using shortcut, the user must change from neutral mode to rock symbol which is opening index and pinky finger as shown in figure 5.2.3. If this gesture is performed google chrome will be opened. This is according to our code; we can also change the action mapped to certain gesture.

Fig 5.2.3: Rock - Symbol

To open a new tab in chrome, the user needs to change right hand from neutral mode to thumbs up symbol that is opening only thumb as shown in the figure 5.2.4. This gesture triggers ctrl+t command which is the keyboard shortcut for opening a new tab.

To close a tab in chrome, the user needs to change left hand from neutral mode to thumbs up symbol that is opening only thumb as shown in the figure 5.2.4. This gesture triggers ctrl+w command which is the keyboard shortcut for closing a tab.

Fig 5.2.4: Thumbs Up Symbol

To open WhatsApp, the user needs to change right hand from neutral mode to call me symbol that is opening pinky finger and thumb. This gesture helps to open WhatsApp application in the computer.

To open calculator, the user needs to change left hand from neutral mode to call me symbol that is opening pinky finger and thumb as shown in the figure 5.2.5. This gesture helps to open calculator application in the computer.

Fig 5.2.5: Call Me Symbol

5.3 CLASSES AND FUNCTIONS

Functions used in the code:

Class Button:

In this class no functions are created, this class is only for creating buttons of keyboard

- Button position that is where the button is to be placed, Size of the button and the text to be contained on the button. These are parameters required for creating a button of a keyboard. These should be given to the function.

drawKeyBoard is a global function used to draw keyboard on GUI screen when the user changes the mode to keyboard mode.

Set_KeyBoard is a global function which is written to change the GUI screen dimensions to fit the keyboard into it and the dimensions of GUI are length = 720, width = 1280.

Set_Mouse is a global function which is written to change the GUI screen dimensions when only single mouse, gestures are used and keyboard is not used. The dimensions of GUI dimensions are length = 480, width = 640.

Class GestureController:

In this class all the gestures are recognized and all actions performed are using the functions implemented in this class.
Functions implemented in Class GestureController:

- Action_listener function checks if any action is performed, if any action is taken place, then it stops processing the incoming video frames for specified delay which is up to certain frames and displays the name of the gesture recognized on the top left corner of the GUI.
- Change_factor function is used to control how much the brightness or volume should increase or decrease. if we stay in same gesture over some time the factor increases and in subsequent frames the brightness or volume increases or decreases more and more as the factor

increases. The function takes previous frame and current frame, if both frames are same and a gesture is same for a specified time i.e., number of frames. Using these two frames by how much the factor should be changed is calculated.

- Volume_change function is used to change volume depending upon the factor that is calculated using the above function (Change_factor).
- Brightness_change function is used to change brightness depending upon the factor that is calculated using the above function (Change_factor).
- Mouse_pointer function takes (x,y) coordinate as input i.e., the coordinate of the tip of index finger relative to GUI screen then the function changes the coordinate from GUI frame to whole screen dimensions and pace the pointer appropriately on the screen.
- Keyboard_press function is used to press a key on keyboard. Middle and index fingers are used to check which key is pressed when we are in keyboard mode by traversing all keys and checking if index finger tip falls inside a key area and if the distance between index finger and middle finger is decreased by bringing close then the key will be pressed.
- Dynamic_init function initializes co-ordinates when performing dynamic action gestures.
- Dynamic_distance function calculate distance between two pair of co-ordinates and checks how much distance the hand has moved in both x-axis and y-axis and finds out in which direction the hand moved vertically up or down and horizontally left or right.
- Start_controller function is used to recognize all the gestures and above-mentioned functions help to perform actions based on gesture recognized here. In this function we check if the user is in keyboard mode or mouse mode and control the mouse or keyboard virtually.

CHAPTER - 6

SYSTEM TESTING

6.1 TESTING

The purpose of this system testing report is to provide a summary of the results of the system testing conducted on our project application A Real-Time Human Computer Interaction using Hand Gestures in OpenCV. The system testing was performed to verify that the application meets the specified requirements and functions correctly in a production environment.

Test environment:

The system testing was performed on a Windows 10 operating system with the following hardware and software configurations:

Processor: Intel Core i5

Memory: 8GB

Storage: 1TB HDD

The 21 points on a palm refer to specific locations on the hand that can be tracked using computer vision algorithms. These points are typically defined based on the anatomy of the hand and are used to represent the shape and posture of the hand in a way that can be easily understood and analyzed by a computer.

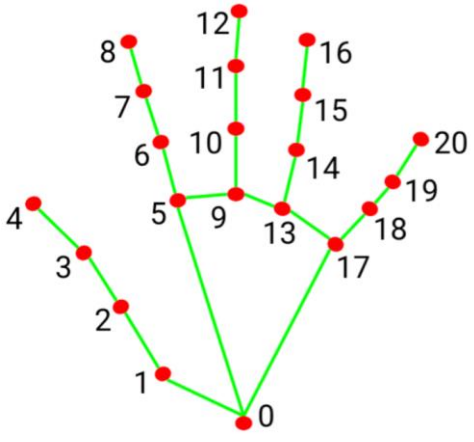

Fig 6.1: All 21 points on the palm

0. Wrist
1. Thumb_CMC
2. Thumb_MCP
3. Thumb_IP
4. Thumb_TIP
5. Index_Finger_MCP
6. Index_Finger_PIP
7. Index_Finger_DIP
8. Index_Finger_TIP
9. Middle_Finger_MCP
10. Middle_Finger_PIP
11. Middle_Finger_DIP
12. Middle_Finger_TIP
13. Ring_Finger_MCP
14. Ring_Finger_PIP
15. Ring_Finger_DIP
16. Ring_Finger_TIP
17. Pinky_MCP
18. Pinky_PIP
19. Pinky_DIP
20. Pinky_TIP

In this context, CMC refers to the carpometacarpal joint, MCP refers to the metacarpophalangeal joint, IP refers to the interphalangeal joint, PIP refers to the proximal interphalangeal joint, and DIP refers to the distal interphalangeal joint.

The specific points that are tracked may vary depending on the specific application or system being used, but some common points include the tips of the fingers, the joints of the fingers and thumb, the base of the thumb, and the wrist. These points can be used to detect and recognize a wide range of hand gestures, such as finger counting, thumb up/down, and fist clenching. They can also be used for other purposes, such as 3D hand pose estimation or virtual reality (VR) applications.

The testing was conducted manually and the results are noted down.

Test results:

Initially we tested our application to recognize left hand and right hand.

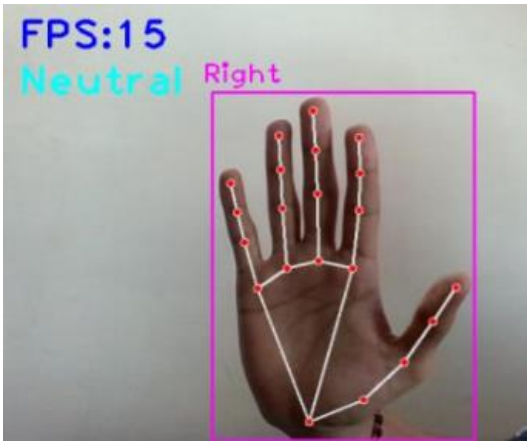

Fig 6.2: Right hand is recognized

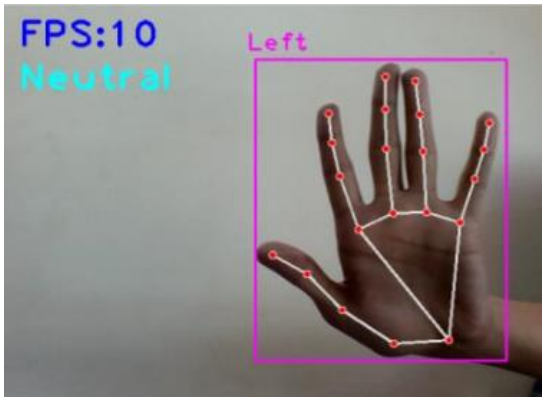

Fig 6.3: Left hand is recognized

The program has successfully recognized all 21 points on the palm which can be seen in the figures 6.2 and 6.3.

CHAPTER – 7
RESULTS AND ANALYSIS

7.1 RESULTS

The below figures are the results and outputs of our project:

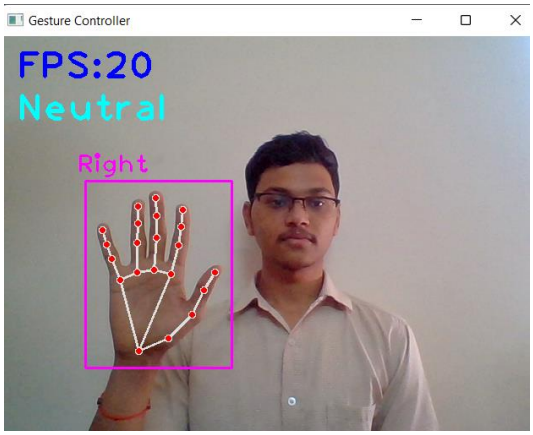

Fig 7.1: Neutral Action

In Figure 7.1 we can see that the system has recognized the action successfully and also identified the gesture as neutral with all the 21 points are visible on the palm. The FPS (Frames per Second) rate can also be seen on the screen. This gesture is not hand specific and it works for both hands.

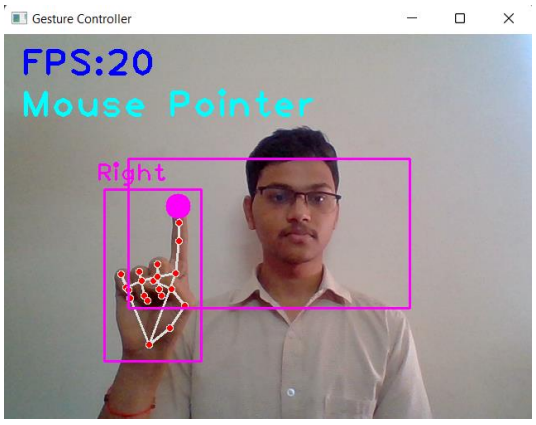

Fig 7.2: Mouse Pointer

Figure 7.2 reveals that the user has transitioned out of the neutral mode and is now revealing only their index finger, which is used to control the movement of the mouse cursor. This gesture is not

hand specific and it works for both hands. Additionally, another bounding box is created for the user inside which they can move the mouse. The fact that the user does not have to move his gesture outside the bounding box benefits the user because it provides him with a smaller workspace, which in turn decreases the amount of strain that is placed on his hand.

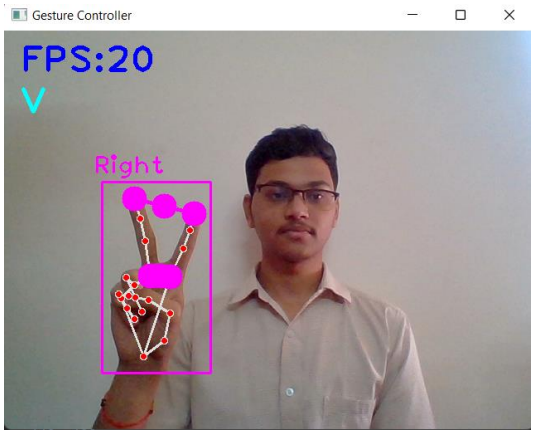

Fig 7.3: 'V' Symbol

Figure 7.3 shows that the user has switched from neutral mode to the V symbol, which opens just the index and middle fingers. This gesture is not hand specific and it works for both hands. The system has detected which hand has been exhibited here (either the left or the right), and the system has also highlighted the finger tips. This may be seen here. The system will indicate which gesture was identified in the top left corner of the display after it has finished recognizing the gesture. We can do actions such as right clicking, left clicking, left double clicking, and dragging by using this gesture.

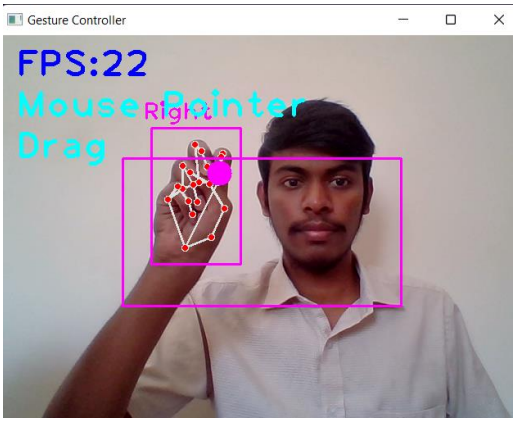

Fig 7.4: Drag Action

Figure 7.4 shows that the user has transitioned from the 'V' symbol to closing their index finger and middle finger, but not fully. This gesture is not hand specific and it works for both hands. The system has detected which hand has been exhibited here (either the left or the right), and the system has also highlighted the finger tips. This may be seen here. After recognizing the gesture, the system reveals which gesture was recognized by highlighting the mouse pointer drag on the upper left side of the screen. This gesture allows the user to pick many files at once, and those files can then be dropped wherever the user chooses.

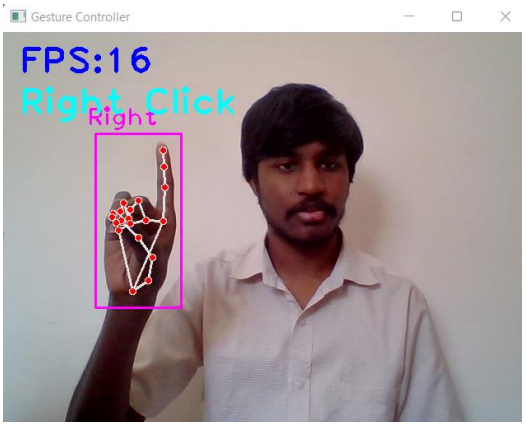

Fig 7.5: Right Click

In Figure 7.5 the user has changed from ‘V’ Symbol to closing only middle finger. This gesture is not hand specific and it works for both hands. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which

gesture is recognized on top left side of the screen which is right click. This gesture helps the user to perform the right click wherever the user stops the mouse and uses this gesture.

Fig 7.6: Left Click

In Figure 7.6 the user has changed from ‘V’ Symbol to closing only index finger. This gesture is not hand specific and it works for both hands. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is left click. When the user stops the mouse and executes this gesture, the computer automatically performs a left click wherever the user stopped the mouse.

Fig 7.7: Double Left Click

In Figure 7.7 the user has changed from ‘V’ Symbol to joining the middle finger and index finger i.e., bringing the two fingers closer. This gesture is not hand specific and it works for both hands. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen

which is double left click. This gesture helps the user to perform the double left click wherever the user stops the mouse and uses this gesture.

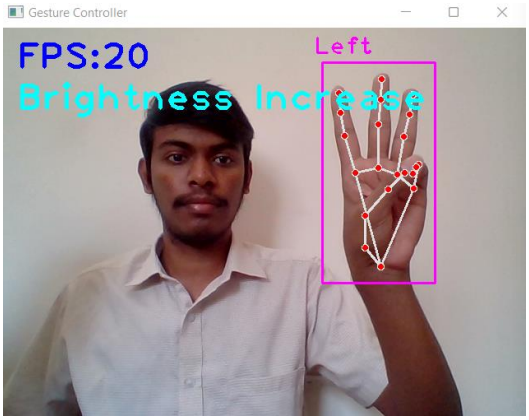

Fig 7.8: Brightness Increase

In Figure 7.8 the user has changed from neutral mode to opening only index finger, middle finger, and ring finger. This gesture is hand specific; it only works for left hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is brightness increase. Here the user moved this gesture vertically upward which performs the increase of brightness and this can be seen on the laptop screen.

Fig 7.9: Brightness Decrease

The user has transitioned from neutral mode to opening only their index finger, middle finger, and ring finger, as seen in Figure 7.9. Because of a hand restriction, you can only use this gesture with

your left hand. It is clear from this that the system can identify the hand that is being displayed (left or right). After identifying the gesture, the system displays the motion that was recognized on the upper left side of the screen; in this case, it is a decrease in the amount of brightness. This gesture was moved vertically downward by the user, which decreased the brightness of the display, as can be seen on the laptop screen.

Fig 7.10: Changing to Keyboard

In figure 7.10 the user has activated the Virtual keyboard option by showing two palms to the system. The palms have been detected, as evidenced by the fact that the system is now displaying a message instructing us to wait until the virtual mouse is replaced by a virtual keyboard.

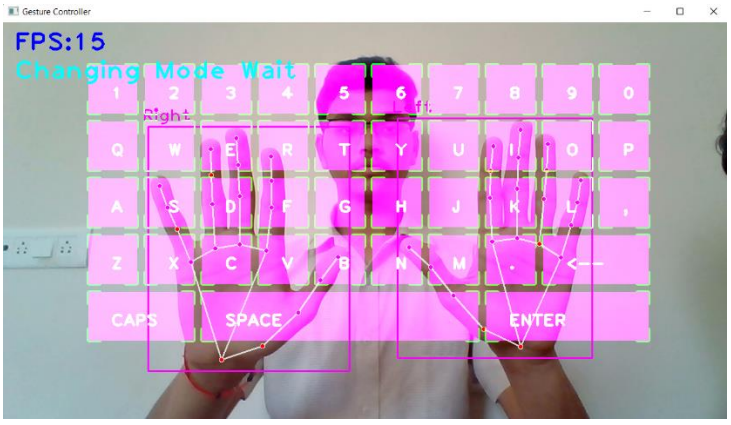

Fig 7.11: Virtual Keyboard

In figure 7.11 the user has shown two palms to the system to activate the Virtual keyboard option. As we can see the system has recognized the palms and is displaying a message stating to wait

until it changes to virtual keyboard from virtual mouse and the figure 7.11 shows a Virtual keyboard displayed on the screen.

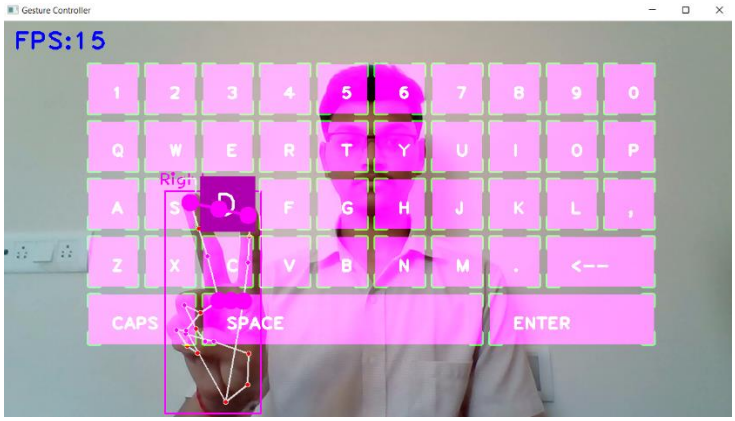

Fig 7.12: Selecting a Letter

In figure 7.12 the user has shown two palms to the system to activate the Virtual keyboard option. As we can see the system has recognized the palms and is displaying a message stating to wait until it changes to virtual keyboard from virtual mouse and the figure 7.12 shows a Virtual keyboard displayed on the screen and now the user has changed his gesture to 'V' symbol to navigate and select a key from keyboard.

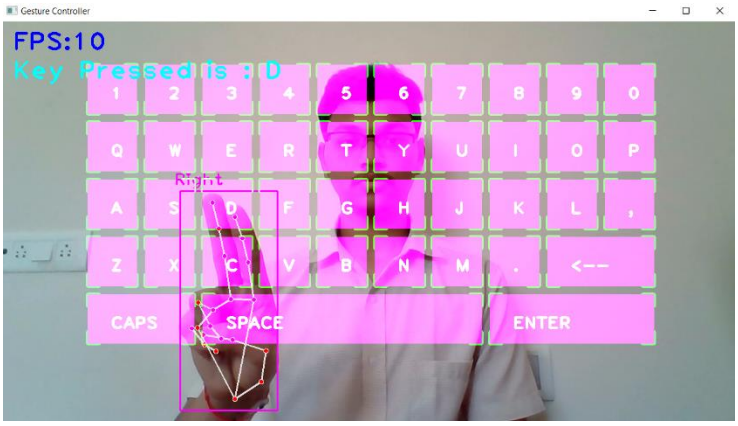

Fig 7.13: Pressing a Letter

In figure 7.13 the user has shown two palms to the system to activate the Virtual keyboard option. As we can see the system has recognized the palms and is displaying a message stating to wait until it changes to virtual keyboard from virtual mouse and the figure 7.13 shows a Virtual keyboard displayed on the screen and now the user has changed his gesture to 'V' symbol and joined the fingers to press a key from keyboard.

Fig 7.14: Volume Increase

In Figure 7.14 the user has changed from neutral mode to opening only index finger, middle finger, and ring finger. This gesture is hand specific; it only works for left hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is volume increase. Here the user moved this gesture in the right direction which performs the increase of Volume and this can be seen on the laptop screen.

Fig 7.15: Volume Decrease

In Figure 7.15 the user has changed from neutral mode to opening only index finger, middle finger, and ring finger. This gesture is hand specific; it only works for left hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is Volume decrease. Here the user moved this gesture in the left direction which performs the decrease of Volume and this can be seen on the laptop screen.

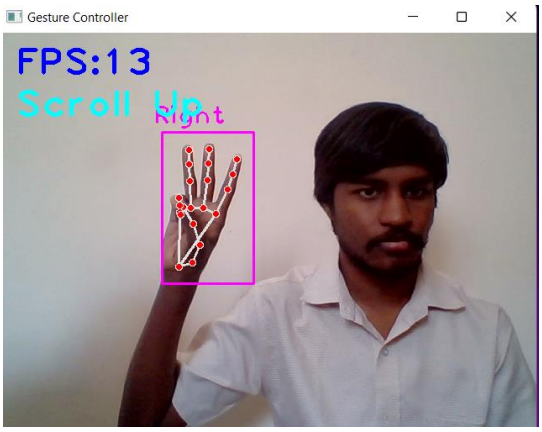

Fig 7.16: Scroll Up

In Figure 7.16 the user has changed from neutral mode to opening only index finger, middle finger, and ring finger. This gesture is hand specific; it only works for right hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is scroll up. Here

the user moved this gesture vertically upward which performs the scroll up action and this can be seen on the laptop screen.

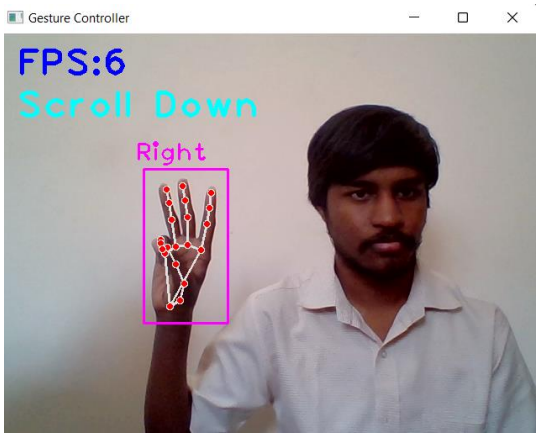

Fig 7.17: Scroll Down

In Figure 7.17 the user has changed from neutral mode to opening only index finger, middle finger, and ring finger. This gesture is hand specific; it only works for right hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is scroll down. Here the user moved this gesture vertically downward which performs the scroll down action and this can be seen on the laptop screen.

Fig 7.18: Scroll Left

In Figure 7.18 the user has changed from neutral mode to opening only index finger, middle finger, and ring finger. This gesture is hand specific; it only works for right hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is scroll left. Here the user moved this gesture in the left direction which performs the scroll left action and this can be seen on the laptop screen.

Fig 7.19: Scroll Right

In Figure 7.19 the user has changed from neutral mode to opening only index finger, middle finger, and ring finger. This gesture is hand specific; it only works for right hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is scroll right. Here the user moved this gesture in the right direction which performs the scroll right action and this can be seen on the laptop screen.

Shortcut gestures:

Fig 7.20 (a): Opening Calculator

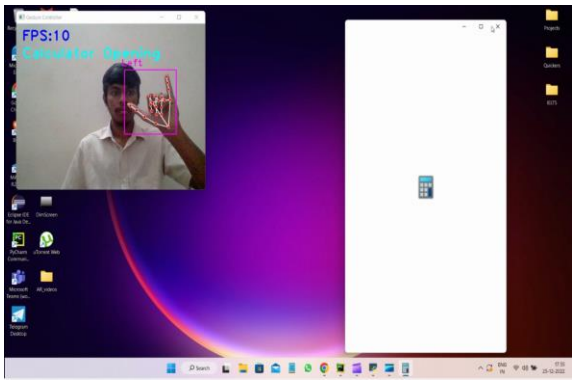

Fig 7.20 (b): Displaying Calculator

In Figure 7.20 (a) the gesture performed is hand specific, it only works for left hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is calculator opening. Here the user changed his gesture from neutral mode to call me symbol i.e, opening in the thumb and the pinky fingers which triggers the opening of calculator app. Displaying the calculator on the screen can be seen in figure 7.20 (b).

Fig 7.21 (a): Opening Browser

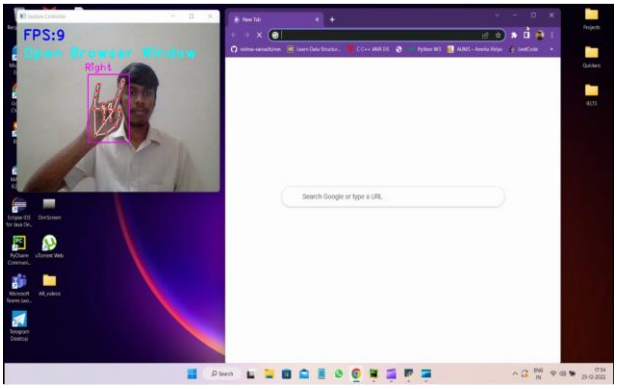

Fig 7.21 (b): Displaying Browser

In Figure 7.21 (a) the gesture performed is hand specific; it only works for right hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is Open Browser Window. Here the user changed his gesture from neutral mode to rock symbol i.e.,

opening the index and the pinky fingers which triggers the opening of browser Window. Displaying the browser on the screen can be seen in figure 7.21 (b).

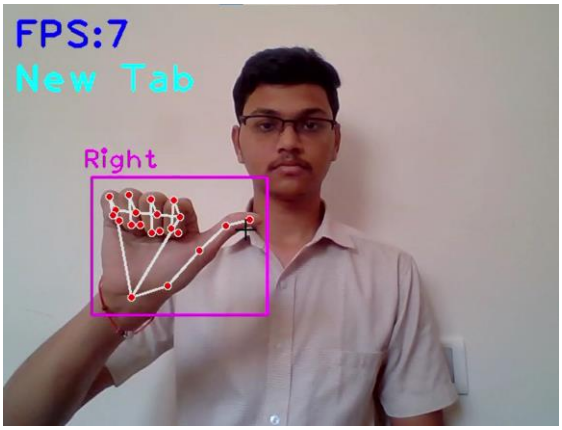

Fig 7.22: Opening New Tab

In Figure 7.22 This gesture is hand specific, it works only right hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is New Tab. Here the user changed his gesture from neutral mode to thumbs up i.e, opening only the thumb finger which triggers the opening of new tab in a browser.

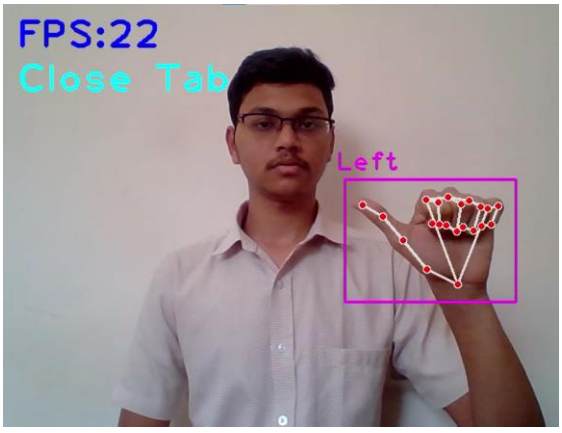

Fig 7.23: Closing Tab

In Figure 7.23 This gesture is hand specific, it works only left hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is Close Tab. Here the user changed his gesture from neutral mode to thumbs up i.e., opening only the thumb finger which triggers the closing of a tab in a browser.

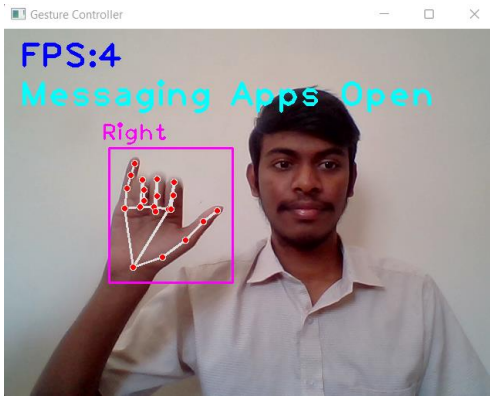

Fig 7.24 (a): Opening Messenger Application

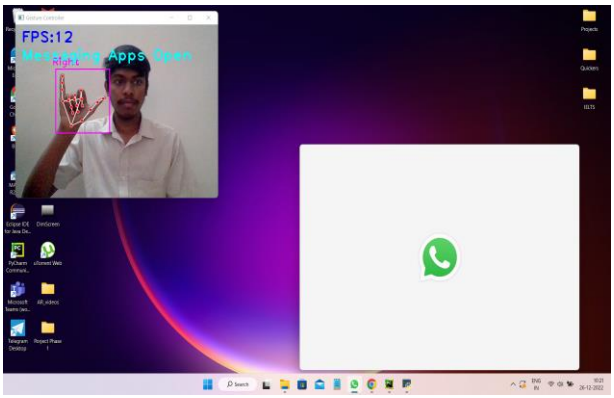

Fig 7.24 (b): Displaying Messenger Application

In Figure 7.24. This gesture is hand specific; it only works for right hand. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is messaging apps open. Here the user changed his gesture from neutral mode to call me symbol i.e., opening in the thumb and the pinky fingers which triggers the opening of messenger apps. Displaying the messaging app on the screen can be seen in figure 7.24 (b).

Fig 7.25: Screenshot

In Figure 7.21 This gesture is not hand specific, it works for both hands. Here we can see that the system has recognized which hand has been shown (left or right). After recognizing the gesture, the system shows which gesture is recognized on top left side of the screen which is Screenshot. Here the user changed his gesture from neutral mode to the closed fist symbol i.e., closing all the fingers which triggers the system to take a screenshot.

7.2 CHALLENGES

We faced the following challenges while implementing our project:

- Have faced difficulty recognizing hand gestures when the lighting conditions were poor or when the hand was occluded by other objects.
- It may be difficult to recognize hand gestures, particularly if the user's hand is not clearly visible, if there is a lot of background noise or movement, or both; additionally, if another person's hand enters the capturing region of the camera, this may also make it difficult to recognize hand gestures. This can lead to misrecognition of gestures and incorrect responses from the computer.
- Low FPS can also cause a delay in performing actions. There are several factors that can contribute to low frame rate (fps) when recognizing hand gestures, some of them are:
    - Processing power: Hand gesture recognition algorithms can be computationally intensive, especially if they involve tracking multiple points on the hand or

detecting complex gestures. This can require a lot of processing power, which can lead to low fps if the computer or device being used is not powerful enough.

- Camera quality: The quality of the webcam or other camera being used to capture the hand gestures can also affect fps. Higher resolution cameras or cameras with higher frame rates may produce better results, but they may also require more processing power to analyze the images.
- Lighting: Poor lighting conditions can make it difficult for the computer to accurately recognize hand gestures, which can lead to low fps.
- Gesture complexity: Complex gestures may be more difficult for the computer to recognize, which can also lead to low fps.
- Algorithm efficiency: The efficiency of the hand gesture recognition algorithm being used can also affect fps. Some algorithms may be more efficient and able to process images faster than others.

To improve fps when recognizing hand gestures, you may need to consider optimizing the processing power of the computer or device being used, using a higher quality camera, improving lighting conditions, simplifying the gestures being used, or using a more efficient algorithm.

- The distance observed between any two points differs as the camera view or distance from camera changes which effects our systems ability to recognize the gestures

CHAPTER – 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

In conclusion, the project on human computer interaction using hand gestures using OpenCV has demonstrated the feasibility and effectiveness of using hand gestures as a means of input for controlling a computer. We achieved a real-time human computer interaction using hand gestures by implementing virtual mouse and keyboard along with some custom gestures mapped to certain actions. We have implemented both static hand gestures and dynamic hand gestures. The use of OpenCV allowed for the real-time tracking and recognition of hand gestures, enabling a seamless and intuitive interaction with the computer.

Overall, the project has shown that hand gestures can be an effective and efficient alternative to traditional input methods such as a keyboard or mouse. It has the potential to be used in a variety of applications, including gaming, presentations, and accessibility for individuals with disabilities.

8.2 FUTURE SCOPE

The human computer interaction can be integrated with a hand glove which has sensors and helps to sense the gesture the user has used. In this case there will no need for the user to show the hand to the camera.

The system can be designed to be more adaptable and flexible to different users, environments, and contexts, by using personalized models or by adapting to the user's gestures and preferences over time.

The system can be applied to other domains and use cases, such as virtual reality, robotics, or assistive technology, where gesture recognition and control can enable more natural and intuitive human-computer interaction.

By exploring these and other opportunities, this project has the potential to make a significant impact and to pave the way for new and innovative ways of interacting with computers and other devices.

REFERENCES

[1] K. Li, J. Cheng, Q. Zhang, and J. Liu, "Hand Gesture Tracking and Recognition based Human-Computer Interaction System and Its Applications," 2018 IEEE International Conference on Information and Automation (ICIA), 2018, pp. 667-672, doi: 10.1109/ICInfA.2018.8812508.

[2] S. R. Chowdhury, S. Pathak and M. D. A. Praveena, "Gesture Recognition Based Virtual Mouse and Keyboard," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 585-589, doi: 10.1109/ICOEI48184.2020.9143016.

[3] G. D. Nagalapuram, R. S, V. D, D. D and D. J. Nazareth, "Controlling Media Player with Hand Gestures using Convolutional Neural Network," 2021 IEEE Mysore Sub Section International Conference (MysuruCon), 2021, pp. 79-86, doi: 10.1109/MysuruCon52639.2021.9641567.

[4] Enkhbat, T. K. Shih, T. Thaipisutikul, N. L. Hakim and W. Aditya, "HandKey: An Efficient Hand Typing Recognition using CNN for Virtual Keyboard," 2020 - 5th International Conference on Information Technology (InCIT), 2020, pp. 315-319, doi: 10.1109/InCIT50588.2020.9310783.

[5] T. -H. Lee and H. -J. Lee, "Ambidextrous Virtual Keyboard Design with Finger Gesture Recognition," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1-4, doi: 10.1109/ISCAS.2018.8351485.

[6] V. Niranjani, R. Keerthana, B. Mohana Priya, K. Nekalya and A. K. Padmanabhan, "System application control based on Hand gesture using Deep learning," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 1644-1649, doi: 10.1109/ICACCS51430.2021.9441732.

[7] Y. Li et al., "A Dynamic Hand Gesture Recognition Model Based on the Improved Dynamic Time Warping Algorithm," 2019 25th International Conference on Automation and Computing (ICAC), 2019, pp. 1-6, doi: 10.23919/IConAC.2019.8895002.