MULTIMEDIA CONTROL USING HAND GESTURES

MEERA PRAKASH (32020108024)

MIDHUN. R (32020108025)

SANDRA S SANTHOSH (32020108030)

ALEN MATHEW (32020108002)

# GOVERNMENT COLLEGE KARIAVATTOM

*(Affiliated to the University of Kerala)*

KARIAVATTOM, THIRUVANANTHAPURAM-695581



# PROJECT REPORT

SUBMITTED IN PARTIAL FULLFILMENT OF THE REQUIREMENT FOR THE

AWARD OF

BSc COMPUTER SCIENCE

DEGREE OF UNIVERSITY OF KERALA 2023

DEPARTMENT OF COMPUTER SCIENCE

GOVT.COLLEGE KARIAVATTOM

(Affiliated to the University of Kerala)

Kariavattom, Thiruvananthapuram-695581



# CERTIFICATE

This is to certify that the report titled “MULTIMEDIA CONTROL USING HAND GESTURES” is a bonafied record of the project done by MEERA PRAKASH (32020108024), MIDHUN R (32020108025), SANDRA S SANTHOSH (32020108030) and ALEN MATHEW (32020108002) under the supervision and guidance of Dr. SREELA S R, Assistant Professor, towards the partial fulfilment of the requirements of SIXTH SEMESTER for the award of the degree BSc Computer Science under the University of Kerala.

Dr. SREELA S R Dr. TINA ELIZABETH MATHEW

Internal project Guide Head of department

Dept. of Computer Science Dept. of Computer Science

Submitted for the viva-voice examination held on

Examiner 1)

2)

## DECLARATION

We hereby declare that the project report titled “MULTIMEDIA CONTROL USING HAND

GESTURES” submitted under the guidance and supervision of Dr. SREELA S R, Assistant Professor in Computer Science is an outcome of our work. This project has not been produced previously for the award of any Degree, Diploma or any other similar title in any other college, University or Institution.

MEERA PRAKASH

MIDHUN R

SANDR S SANTHOSH

ALEN MATHEW

Place: Kariavattom

Date:

### ACKNOWLEDGEMENT

Firstly, we thank God almighty for successfully completion of the project. It is great pleasure and gratitude to express our sincere thanks to those who contribute their support encouragement. We are very much indebted to our principal, government college kariavattom who gave the source of inspiration for achieving great height in the pursuit of excellence.

The project report entitled “MULTIMEDIA CONTROL USING HAND GESTURES” has been done under the Guidance and supervision of Dr. SREELA S R, Assistant Professor in Computer science and Mrs. Tina Elizabeth Mathew, Head of Department. With great respect, we express our sincere and heartfelt thanks for their guidance for preparing this Project.

Last but not the least; we express our gratitude to our parents and friends who have given us Inspiration, mental support and lot of help and encouragement for doing this Project Successfully, who helped us directly or indirectly in the completion of this work. This includes staffs and students of Govt. College Kariavattom.

Sincerely,

MEERA PRAKSH

MIDHUN R

SANDRA S SANTHOSH

ALEN MATHEW

ABSTRACT

In this Project we try to control multi medias like video, PowerPoint, image, word document etc. using hand gestures with the help of OpenCV, media pipe and Python. Computer applications require interaction between human and computer. This interaction needs to be unrestricted and it has made it challenging to traditional input devices such as keyboard, mouse, pen etc. Hand gesture is an important component of body languages in linguistics. Human computer interaction becomes easy with the use of the hand as a device. Use of hand gestures to operate machines would make interaction interesting. Gesture recognition has gained a lot of importance. Hand gestures are used to control other various applications like windows media player, robot control, gaming etc. Use of gesture makes interaction easy, convenient and does not require any extra device. Vision and audio recognition can be used together. But audio commands may not work in noisy environments.

### INDEX

CHAPTER NO TITLE PAGE NO.

DECLARATION

ACKNOWLEDGEMENT

ABSTRACT

1. INTRODUCTION 1

2. SYSTEM REQUIREMENTS 2

2.1 HARDWARE REQUIREMENTS 2

2.2 SOFTWARE REQUIREMENTS 2

2.3 TECHNOLOGIES USED 3

2.3.1 PYTHON 3

2.3.2 OPENCV 5

2.3.3 PYAUTOGUI 5

2.3.4 MEDIAPIPE 6

2.3.5 STREAMLIT 6

2.3.6 PIL  6

2.3.7 MATH 7

|  |  |  |
| --- | --- | --- |
|  | LITERATURE REVIEW | 7 |
| 3.1 | A DYNAMIC HAND GESTURE RECOGNITION | 7 |
|  | SYSTEM FOR CONTROLLING VLC MEDIA |  |
|  | PLAYER |  |
| 3.2 | MEDIA CONTROL USING HAND GESTURES | 7 |
| 3.3 | CONTROLLING POWER POINT USING HAND | 7 |
|  | GESTURES IN PYTHON |  |
| 3.4 | USING REAL-TIME GESTURE TO  AUTOMOTIVE CONTROL | 8 |

3.

2.3.7 MATH  7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3.3 | | A-FAST-SIGHTED HAND-BASED TOUCH | | 8 | |
|  | | ALGORITHM RECOGNITION FOR ROBOT | |  | |
|  | | CONTROL | |  | |
| 4 |  | | **PROBLEM DEFINITION** | | 9 | |
|  | 4.1 | | INTRODUCTION | | 9 | |
|  | 4.2 | | EXISTING SYSTEM | | 9 | |
|  |  | | 4.2.1 LIMITATIONS OF EXISTING SYSTEM | | 9 | |
|  | 4.3 | | PROPOSED SYSTEM | | 9 | |
|  |  | | 4.3.1. ADVANTAGES OF PROPOSED SYSTEM | | 10 | |
|  | 4.4 | | FEASIBLITY STUDY | | 10 | |
|  |  | | 4.4.1 TECHNICAL FEASIBILITY | | 10 | |
|  |  | | 4.4.2 ECONOMICAL FEASIBILITY | | 10 | |
|  |  | | 4.4.3 OPERATIONAL FEASIBILITY | | 11 | |
| 5 |  | | **SYSTEM DESIGN** | | 12 | |
|  | 5.1 | | INTRODUCTION | | 12 | |
|  | 5.2 | | ADOPTION OF MODULES | | 12 | |
|  |  | | 5.2.1 OPEN WEB APPLICATION | | 12 | |
|  |  | | 5.2.2 VIDEO CAPTUREING | | 12 | |
|  |  | | 5.2.3 HAND TRACKING AND HAND  LANDMARK | | 12 | |
| 5.2.4 GESTURE IDENTIFICATION AND | | | | | 13 | |
|  |  | | PERFORMING ACTION | |  | |
| 5.2.5 FLOW CHART | | | | | 14 | |
| 5.2.5.1 VIDEO CONTROL USING HAND | | | | | 14 | |
|  |  | | GESTURE | |  | |
| 5.2.5.2 POWERPOINT CONTROL USING HAND | | | | | 15 | |
|  |  | | GESTURE | |  | |
| 5.2.5.3 IMAGE CONTROL USING HAND | | | | | 16 | |

GESTURE

* 1. DATA FLOW DIAGRAM 17
     1. RULES FOR CONSTRUCTING A DATA 17

FLOW DIAGRAM 17

* + 1. COMPONENTS OF DATA FLOW 17

DIAGRAM

* + 1. CONTEXT LEVEL DIAGRAM 19
       1. LEVEL 1: MAIN MODULE IN MULTI 19

MEDIA CONTROL USING HAND GESTURES

* + - 1. LEVEL 1.1: OPEN WEB APPLICATION 20
      2. LEVEL 1.2: VIDEO CAPTURING
      3. LEVEL 1.3: HAND LANDMARKING AND 20

GESTURE DETECTION

* 1. UML DIAGRAM 21
     1. USE CASE DIAGRAM 21
        1. USE CASE DIAGRAM FOR VIDEO 21

CONTROL

* + - 1. USE CASE DIAGRAM FOR POWER 22

POINT CONTROL

* + - 1. USE CASE DIAGRAM FOR IMAGE 23

CONTROL

* + 1. SEQUENCE DIAGRAM 24
       1. SEQUENCE DIAGRAM FOR VIDEO 24

CONTROL

* + - 1. SEQUENCE DIAGRAM FOR POWER 25

POINT CONTROL

* + - 1. SEQUENCE DIAGRAM FOR IMAGE 25

CONTROL

* + 1. MODULAR DIAGRAM 26
       1. MODULAR DIAGRAM FOR VIDEO 26

CONTROL

* + - 1. MODULAR DIAGRAM FOR POWER 27

POINT CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| * + - 1. MODULAR DIAGRAM FOR IMAGE   CONTROL | | | 27 |
| 6 |  | **SYSTEM IMPLEMENTATION** | 29 |
|  | 6.1 | INTRODUCTION | 29 |
|  | 6.2 | ALGORITHM FOR THE SYSTEM | 30 |
|  |  | 6.2.1 VIDEO CONTROL | 30 |
|  |  | 6.2.2 POWER POINT CONTROL | 31 |
|  |  | 6.2.3 IMAGE CONTROL | 31 |
| 7 |  | **SYSTEM TESTING** | 33 |
|  | 7.1 | INTRODUCTION | 33 |
|  | 7.2 | LEVELS OF TESTING | 33 |
|  |  | 7.2.1 UNIT TESTING | 33 |
|  |  | 7.2.2 INTEGRATION TESTING | 33 |
|  |  | 7.2.3 ACCEPTANCE TESTING | 34 |
|  |  | 7.2.4 SYSTEM TESTING | 34 |
|  | 7.3 | TEST CASE & OUTPUT | 35 |
|  |  | 7.3.1 VIDEOPLAYER | 35 |
|  |  | 7.3.2 POWEREPOINT | 35 |
|  |  | 7.3.3 IMAGE | 36 |
| 8 |  | **CONCLUSION AND FUTURE** | 37 |
|  |  | **ENHANCEMENT** |  |
|  | 8.1 | CONCLUSION | 37 |
|  | 8.2 | FUTURE ENHANCEMENT | 37 |
|  |  | **APPENDICES** | 38 |
|  |  | SCREENSHOTS | 38 |
|  |  | **REFERENCES**  CODING | 43 |

### 1.INTRODUCTION

In today’s world, everyone is surrounded by machines and gadgets at all times. The use of mobile phones, tablets, computers and smartwatches has increased manifold due to creative new technology and Internet use. The system that contains a combinationof hardware and software for communication between the user and the computer or machine, makes up the human computer interface (HCI). Normally, switches, touch screens and control elements are used. An easier way of communication is through the touchless user interface, which could be through voice or gesture input. This project focuses on gesture input and create a web application for controlling multimedia using hand gestures. Instead of using a mouse or keyboard, users can show hand gestures to perform operations or control certain functions of the computer. This could include playing a video and controlling power point presentations and images.

This project aims to develop a web application which focus on the field of gesture recognition. Through this application, users will be able to control video, Microsoft power point, Microsoft Word and images in the system using hand gestures. The input is taken through the webcam of the system. The programming languages used is Python.

A common scenario where this application will be able to help users is in the kitchen, where users may not be able to touch device screens or press buttons with their hands while cooking. Another scenario is devices used at the beach or poolside, where users will have their hands too wet or sandy to type. Some of the other common applications are in automated homes, driving safety, television control and much more. With upcoming IoT (Internet of Things) devices, it is possible to switch off lights, control surround sound systems, access devices and change room temperature through the connected devices. Many of these devices use gesture recognition and voice input.

### 2.SYSTEM REQUIREMENTS

**2.1 HARDWARE REQUIREMENTS**

The selection of hardware configuration is a very important task related to the software development. Random access memory may affect adversely on the speed and correspondingly on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should have sufficient capacity to store the database and the application. The network should be efficient to handle the communication fast.

* Processor : Intel i3 or above
* RAM : 8 GB
* Hard Disk Memory : SSD 512 GB
* Keyboard : Normal or Multimedia Web Camera

2.2 SOFTWARE REQUIREMENTS

* Operating System: Windows 10
* Language: Python

**2.3 TECHNOLOGIES USED**

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system which enables the user to interact with several physical hardware devices. The minimum software requirement specification for developing this project are as follows:

• Operating System: Windows 10 • Documentation Tool: MS Word

**2.3.1 PYTHON**

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is openSource software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation (PSF). Python’s development is conducted largely through the Python Enhancement Proposal (PEP) process. The PEP process is the primary mechanism for proposing major new features, for collecting community-input on an issue, and for documenting the design decisions that have gone into Python.

FEATURES OF PYTHON:

There are many features in Python, some of which are discussed below –

1.Easy to code:

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, JavaScript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developerfriendly language.

2. Free and Open Source:

Python language is freely available at the official website and you can download it. Since it is open-source, this means that source code is also available to the public. So, you can download it as, use it as well as share it. 3. Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object-

oriented language and concepts of classes, objects encapsulation, etc.

1. GUI Programming Support:

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

1. High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

1. Extensible feature:

Python is an Extensible language. We can write us some Python code into C or C++ language and also, we can compile that code in C/C++ language.

1. Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

1. Python is Integrated language:

Python is also an integrated language because we can easily integrated python with other

languages like c, c++, etc.

1. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time like other languages C, C++, Java, etc.

1. Large Standard Library:

Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

1. Dynamically Typed Language:

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don’t need to specify the type of variable.

**2.3.2 OPENCV**

OpenCV (Open-source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow George then Itseez. The library is cross-platform and free for use under the open-source BSD license. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, exact 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of entire scene. Human eyes provide lots of information based on what they see. Machines are facilitated with seeing everything, convert the vision into numbers and store in the memory. Here the question arises how computer convert images into numbers. So, the answer is that the pixel value is used to convert images into numbers. A pixel is the smallest unit of a digital image or graphics that can be displayed and represented on a digital display device. The picture intensity at the particular location is represented by the number.

**2.3.3 PYAUTOGUI**

PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be as simple. PyAutoGUI works on Windows, macOS, and Linux, and runs on Python 2 and 3.

PyAutoGUI has several features:

* Moving the mouse and clicking or typing in the windows of other applications

Sending keystrokes to applications (for example, to fill out forms).

* Take screenshots, and given an image (for example, of a button orcheckbox), find it on the screen.
* Locate an application’s window, and move, resize, maximize, minimize, or close it

(Windows-only, currently)

* Display message boxes for user interaction while your GUI automation script runs.

The x, y coordinates used by PyAutoGUI has the 0, 0 origin coordinates in the top left corner of the screen. The x coordinates increase going to the right (just as in mathematics) but the y coordinates increase going down (the opposite of mathematics). On a screen that is 1920 x 1080 pixels in size, coordinates 0, 0 are for the top left while 1919, 1079 is for the bottom right. Currently, PyAutoGUI only works on the primary monitor. PyAutoGUI isn't reliable for the screen of a second monitor (the mouse functions may or may not work on multi-monitor setups depending on your operating system and version). All keyboard presses done by PyAutoGUI are sent to the window that currently has focus, as if you had pressed the physical keyboard key.

**2.3.4 MEDIAPIPE**

Media-Pipe is a framework for building cross-platform (i.e., Android, iOS, web, edge devices) multimodal (e.g., video, audio, any time series data) applied Machine Learning pipelines that consist of fast ML inference, classic computer vision, and media processing (e.g., video decoding). Media-pipe has released various prebuilt python and other language packages like:

* Object detection
* Face detection
* Hand tracking
* Pose estimatimation
* Multi-hand tracking
* Hair segmentation

**2.3.5 STREAMLIT**

Stream lit is an opensource app framework in python language. It helps us create beautiful web-apps for data science and machine learning in a little time. It is compatible with major python libraries such as scikit-learn, keras, pytorch, latex, NumPy, pandas, matplotlib, etc.

* It embraces python-scripting.
* Less code is needed to create amazing web-apps.
* No callbacks are needed since widgets are treated as variables.
* Data caching simplifies and speeds up computation pipelines.

**2.3.6 PIL**

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

**2.3.7 Math**

This module provides access to the mathematical functions defined by the C standard .These functions cannot be used with complex numbers; use the functions of the same name from the cmath module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don’t is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

### 3.LITERATURE REVIEW

**3.1 A DYNAMIC HAND GESTURE RECOGNITION SYSTEM FOR CONTROLLING VLC MEDIA PLAYER**

Authors: Manuj Paliwal, Gaurav Sharma (2020, IEEE)

A low-value device that makes use of dynamic hand gesture reputation method to govern the VLC media player. This software contains a crucial computation module that segments the foreground a part of the frame, the use of pores and skin detection, and the approximate median method. The popularity of gestures is done by growing a Decision Tree that uses various capabilities extracted from the segmented component. This hand gesture popularity approach introduces a new, herbal way to interact with computers. With the increase in the interaction of computer systems in our everyday existence, it'd be worth enough to get a Perceptual User Interface (PUI) [1, 2] to interact with computers as humans interact with each other. Vision based gesture popularity is an important generation for the friendly human-pc interface and has received more and more attention in current years.

**3.2 MEDIA CONTROL USING HAND GESTURES:**

Authors: Vallabh Chapalgaonkar, Atharva Kulkarni, Amey Sonawale

Gesture-based real-time gesture recognition systems received great attention in recent years because of their ability to interact with systems efficiently through human-computer interaction. Human-Computer Interaction can gain several advantages with the establishment of different natural forms of device-free communication. Gestures are a natural form of action that we often use in our daily lives to interact, so to use them as a way of communicating with computers generates a new paradigm of computing interaction. This project implements computer vision and gesture recognition techniques and develops a vision based low-cost input software for controlling the media player through gestures.

**3.3 CONTROLLING POWER POINT USING HAND GESTURES IN PYTHON**

Authors: Muhammad Idrees, Ashfaq Ahmad, Muhammad Arif Butt, and Hafiz Muhammad

Danish

Presentations have a significant role in different fields of life. Whether you are a student, an entrepreneur, businessman, or a corporate worker, you must have had given presentations at some point in your life. PowerPoint presentations sometimes become less lively because either you have to use the keyboard to change and operate the slides or use a dedicated gadget to perform these tasks. We aimed to enable people to control the slideshow with the gestures of hands. The applications of gestures in human-computer interaction have massively risen in the past few years. The research has tried to control different operations of the PowerPoint slideshow through gestures. This research has used Machine Learning to detect gestures with subtle differences and tried to map them with some fundamental PowerPoint slideshow controlling functions using Python.

**3.4 USING REAL-TIME GESTURE TO AUTOMOTIVE CONTROL**

Authors: Shilpa Chaman, Jay Jani, Henson Fernandes, Rahila Dhuka, Dhanvin Mehta (2018, IEEE)

The system Real-Time Gesture to Automotive Control (G2AC) system is developed in this paper, using hand gesture recognition to handle the media player in automotive. The system presents fast gesture recognition with low complexity algorithms for controlling real-time media in automotive systems. In the proposed vision-based system two modules are interconnected: one module recognizes the hand gesture in the region of interaction and another module does the task of selecting music from the media player using Raspberry pi. The proposed G2AC system can recognize real-time hand gestures with 98 percent accuracy which is demonstrated using a hand gesture dataset collected under different settings of illumination variation, hand orientation, and occlusion.

**3.4 A-FAST-SIGHTED HAND-BASED TOUCH ALGORITHM RECOGNITION FOR**

**ROBOT CONTROL**

Authors: Erol Ozgur, Asansarabi Malima

In 2006, it formed Erol Ozgur and Asansarabi Malima "A fast-sighted hand-based touch algorithm Recognition for Robot Control” which controlled the robot using hand gestures but with limited touch. First the division of the hand circuit was followed by pointing fingers and finally separating the gestures. The algorithm used is consistent in translation, rotation and hand scale. This program works on a robot control app with reliable performance.

### 4.PROBLEM DEFINITION

**4.1 INTRODUCTION**

In my project I introduce a web app for multimedia control using hand gestures. Gesture recognition is the process by which systems can see, recognize andrespond to gestures shown by the user. This perceptual user interface provides a way forusers to communicate with the system without the need for clicking and typing. Gesturescan come from any bodily motion or state. Usually, they come from the hand or face. During my project I went through the different system development life cycle. First of all, I started with system study which helped me understand scope of the system. During this phase I am able to understand the limitations of the existing system and it also helped me in realizing the requirements from the client's perspective.

**4.2 EXISTING SYSTEM**

In the existing system we control video, power point presentations, images by using mouse and keyboard for all inputs. The problem with traditional systems can arise in many ways. Firstly, for people with disabilities, it is not possible to click buttons or type for every operation. Secondly, ifsome part of your computer is not working properly, it is not possible to rely on the traditional use of mouse and keyboard for all inputs. Thirdly, even if people are able and have systems that are usable, nowadays, every part of life is being automated and simplified. The hardware required also costs a lot of money.

**4.2.1 LIMITATIONS OF THE EXISTING SYSTEMS:**

* Existing systems lack natural communication between humans and machines
* Currently available gesture recognition systems are expensive

**4.3 PROPOSED SYSTEM**

In proposed system we are using the web app for muti media control using hand gestures we can easily control multi medias like video, PowerPoint, image by using the different hand gestures. My project uses systems web camera as an input device to capture gestures performed by the users. The vision-based gesture recognition seems to be a better option due to its advantages over non-vision-based method. With the help of image processing preprocessing the images and which is easy to find the hand landmarks and track the gesture movement. The major aim behind this project is to enhance user experience while using any computer system. By providing an easier way to control multimedia applications, users will find more interest in using computers altogether. This project aims to provide such a touchless user interface for computer systemswith the required hardware and software combination. This application can be used to control applications like media player, Microsoft PowerPoint, Microsoft word, image management. by using hand gestures instead of clicking the buttons on the screen or on the keyboard.

**4.3.1 ADVANTAGES OF PROPOSED SYSTEM:**

* Since it uses only systems webcam or external webcam as hardware, it is cost efficient.
* Maintenance is much easier than the existing gesture recognition system.
* User friendly and easily adaptable web app which explain the working of the application.
* Easy to implement.

**4.4 FEASIBILITY STUDY**

A feasibility study is made to see the proposed system in the light of the workability, meeting user’s requirements, effective use of resources, effort and the time that is spend and of course, the cost effectiveness. An important outcome of the preliminary investigation determining whether the system required is feasible. There are three aspects in the feasibility study portion of the preliminary investigation.

* Economic Feasibility
* Technical Feasibility
* Operational Feasibility

**4.4.1 TECHNICAL FEASIBILITY**

The web app must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Technical feasibility centers around computer system and to what extend it can support the proposed addition. This involves the financial considerations to accommodate the additional technical enhancements. If budget is not a serious constraint, then the project is judged technically feasible. Since the project uses python as programming language and stream lit and html for web app designing which is easy to understand and the packages used for image processing and GUI automation are freely available. The systems webcam is used to fetch the data, so the proposed multimediacontrolled web app is technically feasible.

**4.4.2 ECONOMICAL FEASIBILITY**

The developing system must be justified cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. This deals with whether expected cost saving, increase the profits and reductions in required investment, and other benefits exceed the cost of developing and operating the proposed system. Its preliminary investigation is concentrated on costs of hardware and software. The project uses python as programming language and the packages used for image processing and stream lit and html are used to web page designing which are freely available. The only external hardware required is a webcam and it’s not mandatory.

**4.4.3 OPERATIONAL FEASIBILITY**

The operational feasibility depends up on whether system performed in the expected way or not. The project uses python as programming language which is easy to understand and the packages used for image processing and stream lit and html for web app designing are freely available. The web app developed is so simple and user friendly there is no special user training is required. The language used in is English and every people can operate it reading options. The type of different gestures, buttons, and about the project are written in the home page of the web app. Here we use separate pages for media player control and power point control. There is no need of special training required for the users.

### 5.SYSTEM DESIGN

**5.1 INTRODUCTION**

System Design develops the architectural details required to build system or product. The system design process encompasses the following activities:

* Partition the analysis model into subsystems.
* Identify concurrency that is dictated by the problem.
* Develop design for the user interface.
* Choose a basic strategy or implementing data management.
* Identify global resources and the control mechanisms required to access them.
* Design an appropriate control mechanism for the system, including task management.

**5.2 ADOPTION OF MODULES**

The application consists of Four Modules • Open web application.

* Video capturing.
* Hand tracking and Hand Landmark
* Gesture identification and performing action
  + 1. **OPEN WEB APPLICATION**

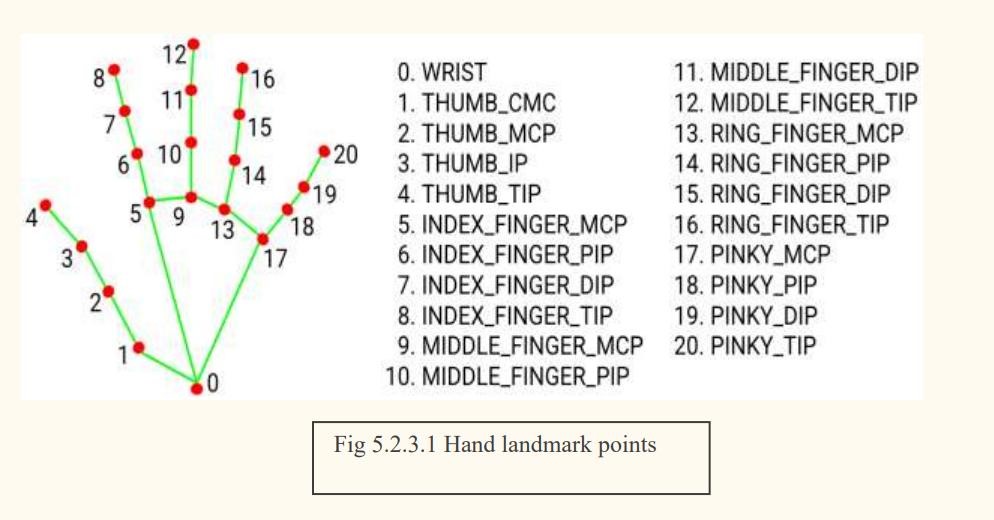
Stream lit and html is used to design the web application. In the application 3 subpages select the type of media (video, image, power point) a to perform.

* + 1. **VIDEO CAPTUREING**

The web cam is open by using open cv and capture the different hand gestures of the user.

* + 1. **HAND TRACKING AND HAND LANDMARK**

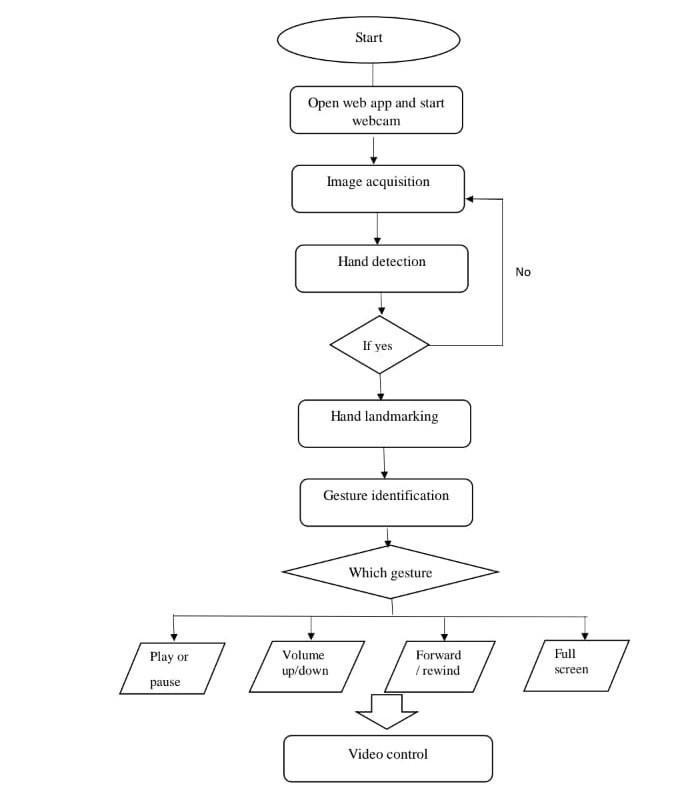
Achieves precise key point localization of 21 key points with a 3D hand-knuckle coordinate which is conducted inside the detected hand regions through regression which will produce the coordinate prediction directly which is a model of the hand landmark in Media Pipe. Each hand-knuckle of the landmark has coordinate is composed of x, y, and z where x and y are normalized to [0.0, 1.0] by image width and height, while z representation the depth of landmark. The depth of landmark that can be found at the wrist being the ancestor. The closed the landmark to the camera, the value becomes smaller.



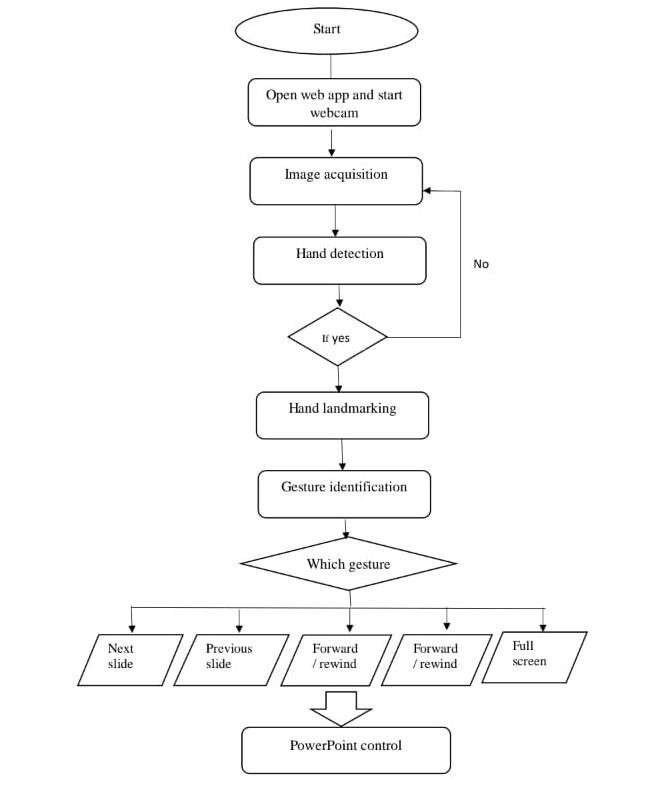
* + 1. **GESTURE IDENTIFICATION AND PERFORMING ACTION**

Based on the handland mark and algorithm detect the type of hand gesture and pyautogui module is used to perform the corresponding keyboard action.

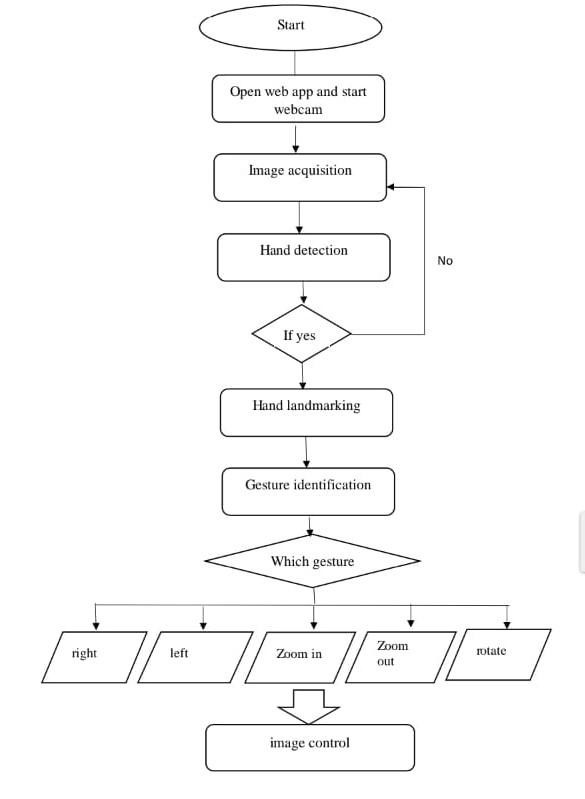
* + 1. **FLOW CHART**
       1. *VIDEO CONTROL USING HAND GESTURE*



* + - 1. *POWERPOINT CONTROL USING HAND GESTURE*



* + - 1. *IMAGE CONTROL USING HAND GESTURE*



**5.3 DATA FLOW DIAGRAM**

A dataflow diagram is a graphical technique that depicts information and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A level-0 DFD is also called a fundamental system model represents the entire software elements as a single bible with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level, i.e., level 1 DFD. Each of the process represented at level 1 are sub functions of overall system depicted in the context model.

ADVANTAGES

* Users easily understood these simple notations.
* Users can make suggestions for modifications.
* They can also spot problem quickly.
* If analyst wants to overview the overall system late, they use the higher overview of the system.

**5.3.1 RULES FOR CONSTRUCTING A DATA FLOW DIAGRAM**

* Arrows should not cross each other.
* Squares, circles and files must bear names.
* Decomposed data flow squares and circles can have same names.
* Choose meaningful names for data flow.
* Draw all data flows around the outside of the diagram.

**5.3.2 COMPONENTS OF DATA FLOW DIAGRAM**

Data Flow Diagram (DFD) is an important tool used by system analyst. DFD provide an overviewof what data a system would process, what transformation of data are done, what files are used and where the results flow. The graphical representation of the system makes it a good communication tool between the user and the analyst. Analysis model help us to understand the relationship between different components in the design. Analysis model shows the user clearly how a system will function. This is the first technical representation of the system. The analysis modeling must achieve three primary objectives.

* To establish a basis for creation of software design.
* To describe what the user requires.
* To define set of requirements that can be validated once the software us build.

A data flow diagram is a graphical technique that depicts information flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A level 0 DFD also called fundamental system model represents the entire software elements as single bubble with input and output indicated by incoming and outgoing arrow respectively. To construct the data flow diagram we use arrows, circle, and rectangles.

`A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD show what kind of information will beinput to and output from the system where the data will come from and go to and where the datawill be stored. The symbols used to draw DFD are:

External Entity represents the source of data that enter in to the system or the recipients of data that leave the system.

* Processes represent activities in which data that are manipulated by being stored or transformed in some way.
* A data flow shows the flow of information from its source to its destination. A linerepresents a data flow, with arrow heads showing the direction of flow.
* Data stores represent stores of data within the system. Data stores may be long-term filessuch as sales ledgers, or may be short-term accumulations

*5.3.3 CONTEXT LEVEL DIAGRAM*

Request

Response



Multimedia

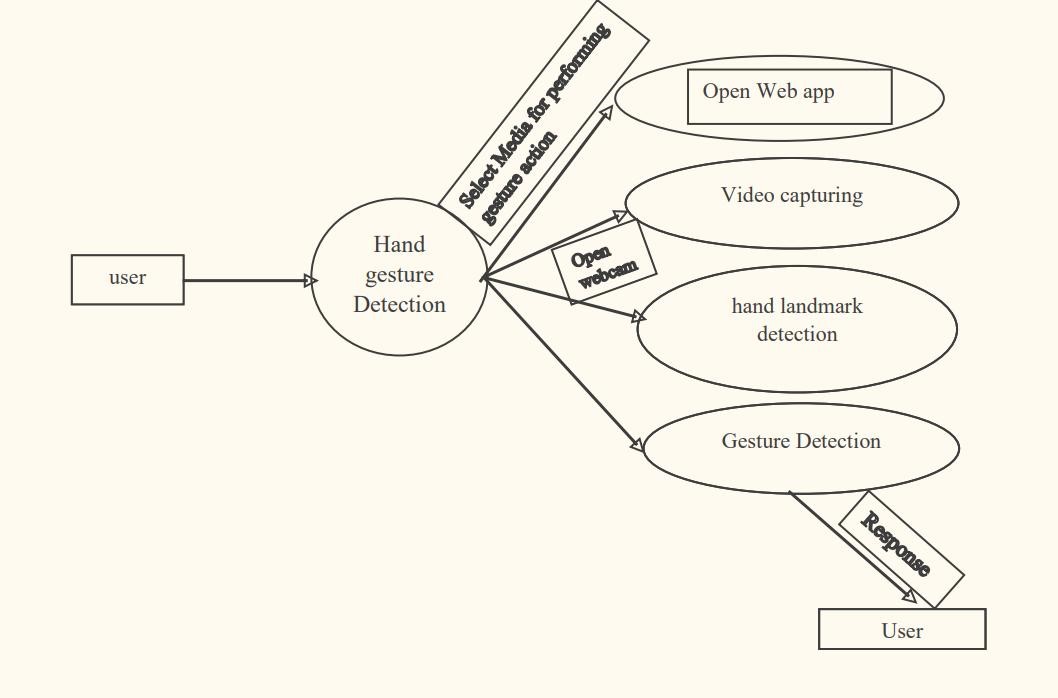
control

using

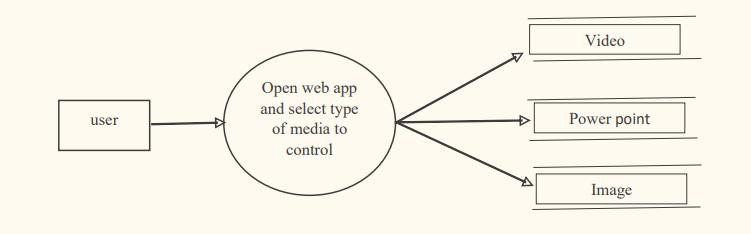
hand

gestures

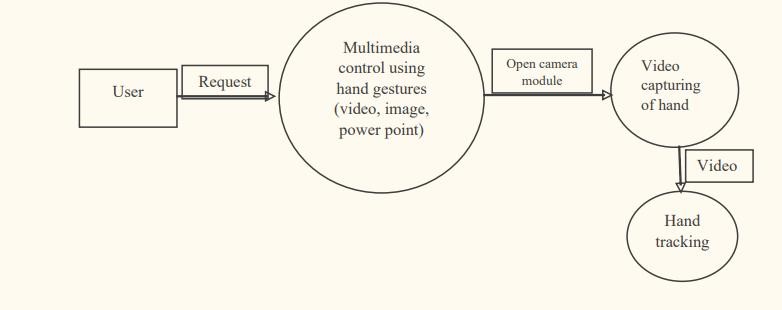
*5.3.3.1 LEVEL 1: MAIN MODULE IN MULTI MEDIA CONTROL USING HAND GESTURES*



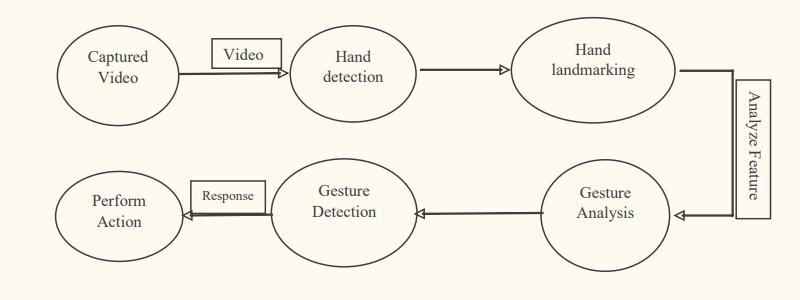
*5.3.3.2 LEVEL 1.1: OPEN WEB APPLICATION*

**

*5.3.3.3 LEVEL 1.2: VIDEO CAPTURING*



*5.3.3.4 LEVEL 1.3: HAND LANDMARKING AND GESTURE DETECTION*



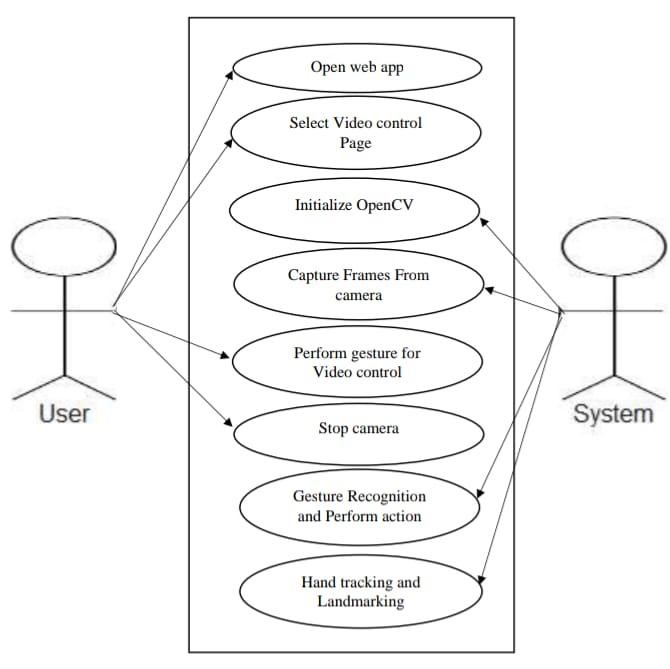
**5.4 UML DIAGRAM**

The Unified Modelling Language is a standard visual modelling language intended to be used formodelling, analysis, design, and implementation of software-based systems.

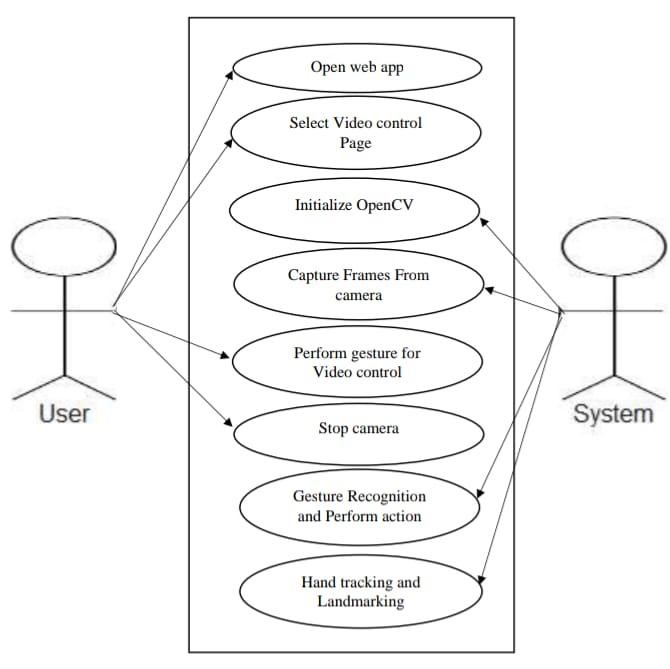
**5.4.1 USECASE DIAGRAM**

A use case is the set of scenarios that describing an interaction between a user and a system. A use diagram displays relationship among actors and use cases. The two main components of a use case diagram are use case and actors. An actor is representing a user or another system that will interact with the system you are modeling. A user is an external view of the system that represents some action the user might perform in order to complete a task.

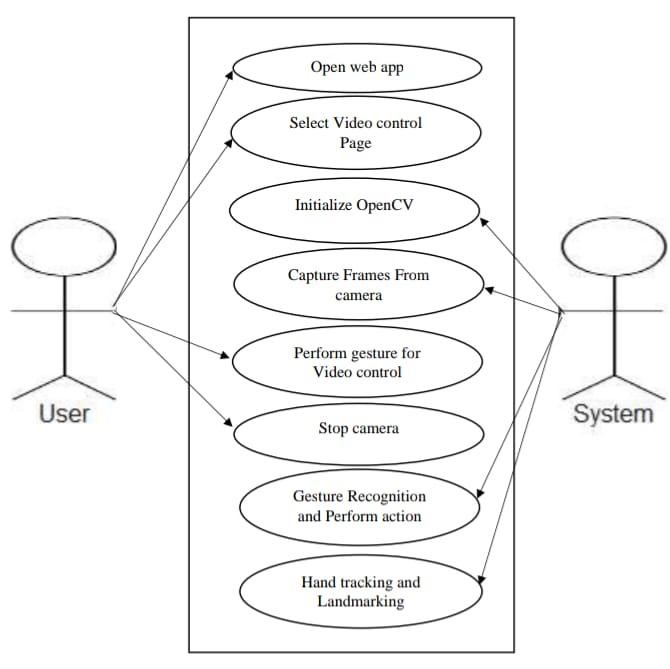
*5.4.1.1 USE CASE DIAGRAM FOR VIDEO CONTROL*



*5.4.1.1 USE CASE DIAGRAM FOR VIDEO CONTROL*



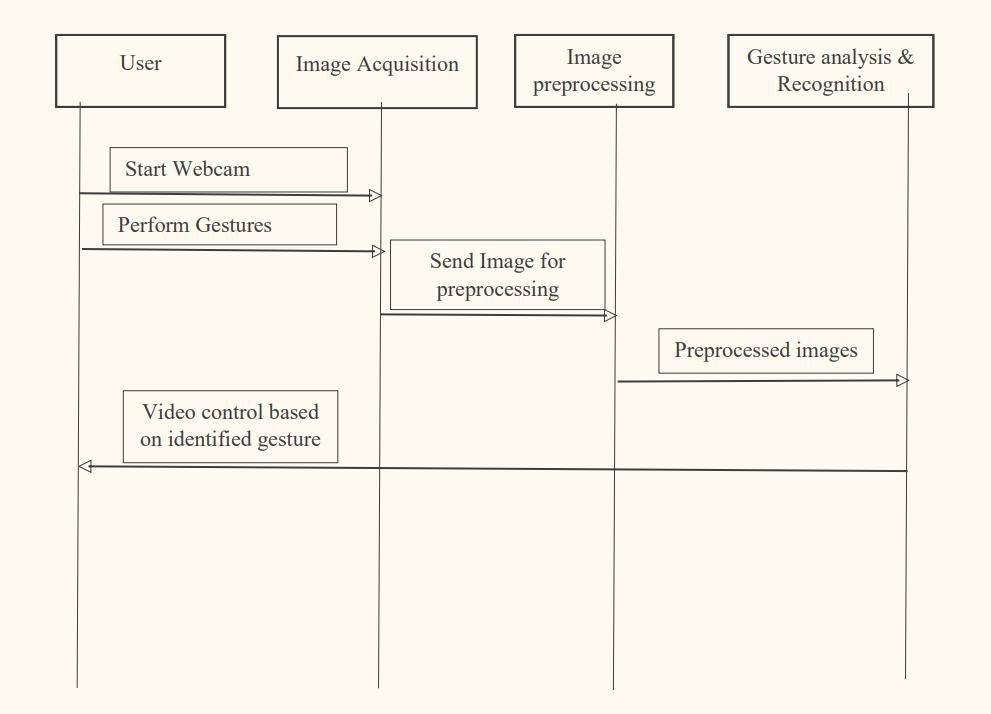
*5.4.1.3 USE CASE DIAGRAM FOR IMAGE CONTROL*



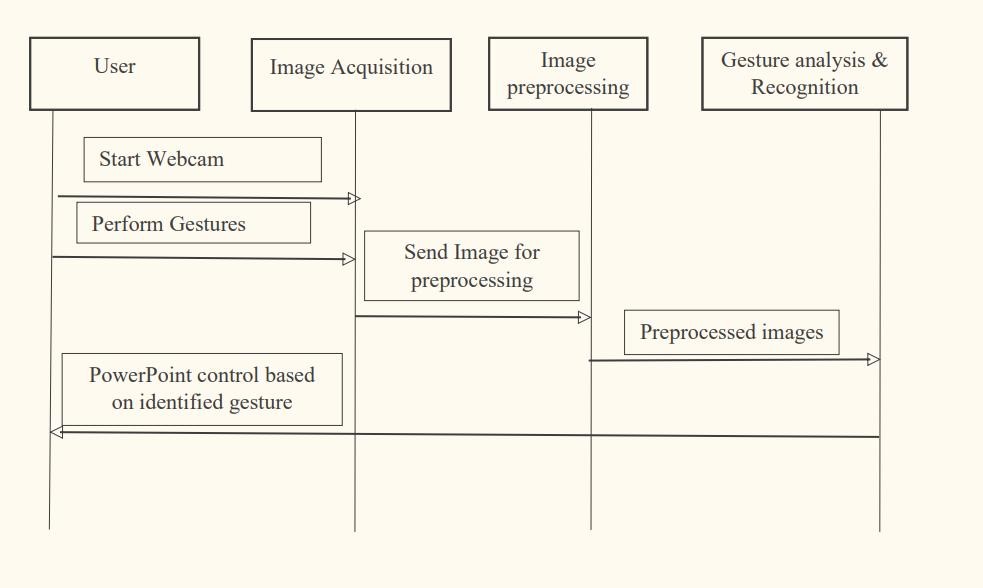
**5.4.2 SEQUENCE DIAGRAM**

A Sequence diagram shows interaction among arranged in a time sequence. It shows the objects participating in the interaction by their life lines and the messages they exchange, arranged in a time sequence. The sequence diagram has two dimensions; the vertical dimension represents time; the horizontal dimension represents different objects. The vertical line is called objects lifeline. The life line represents the objects existence during the interaction.

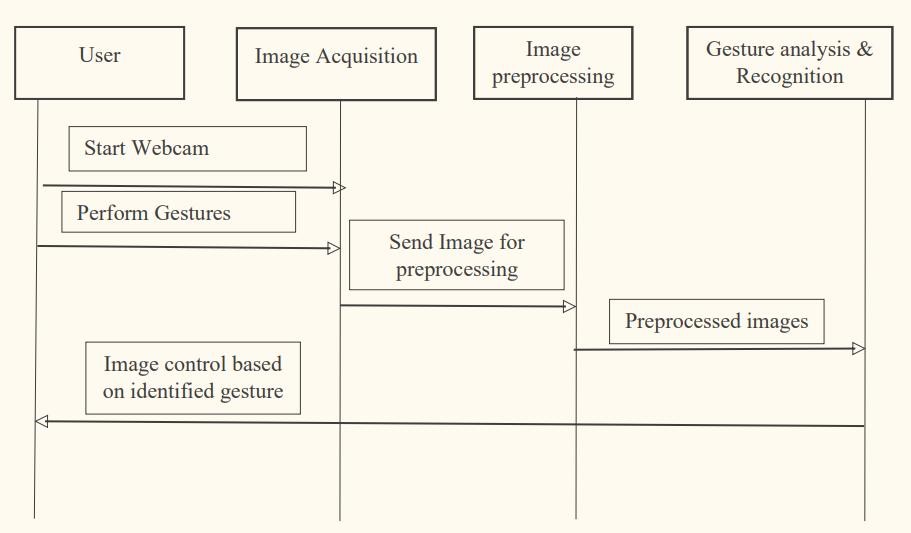
*5.4.2.1 SEQUENCE DIAGRAM FOR VIDEO CONTROL*



*5.4.2.2 SEQUENCE DIAGRAM FOR POWER POINT CONTROL*



*5.4.2.3 SEQUENCE DIAGRAM FOR IMAGE CONTROL*



**5.4.3 MODULAR DIAGRAM**

Modular diagram is used to represent the modules in the software. In the diagram rectangles and arrows are used to represent the working of modules.

*5.4.3.1 MODULAR DIAGRAM FOR VIDEO CONTROL*

User

video

control

using

hand

gesture

System

Initialize

OpenCV

Capture

frames

from

camera

convert

the

BGR

image

to

RGB

and

Hand

Detection

convert

the

RGB

image

to

BGR

Gesture

detection

Perform

action

Open

Web

App

And

select

Video

control

page

Start

camera

Select

a

video

Perform

Gestures

for

video

control

Stop

camera

*5.4.3.1 MODULAR DIAGRAM FOR VIDEO CONTROL*

User

Powerpoint

control

using

hand

gesture

System

Initialize

OpenCV

Capture

frames

from

camera

convert

the

BGR

image

to

RGB

and

Hand

Detection

Draw

hand

annotations

on

image

Gesture

detection

and

perform

action

Perform

action

Open

Web

App

And

select

Video

control

page

Start

camera

Select

a

Power

Point

Perform

Gestures

for

Power

Point

control

*5.4.3.3 MODULAR DIAGRAM FOR IMAGE CONTROL*

User

Image

control

using

hand

gesture

System

Initialize

OpenCV

Capture

frames

from

camera

convert

the

BGR

image

to

RGB

and

Hand

Detection

Draw

hand

annotations

on

image

Gesture

detection

and

perform

action

Open

Web

App

And

select

Video

control

page

Start

camera

Select

an

image

Perform

Gestures

for

image

control

### 6.SYSTEM IMPLEMENTATION

**6.1 INTRODUCTION**

The purpose of System Implementation can be summarized as follows: making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the Performing Organizing (the transition). At a fine level of detail, deploying the system consists of executing all the steps necessary to educate the consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support and maintenance mode of operation, with ownership of the new system moving from the Project Team to the Performing Organization.

A key difference between System Implementation and all other phases of the lifecycle is that all project activities up to this point have been performed in safe, protected, and secure environments, where project issues that arise have little or no impact on day-to-day business operations. Once the system goes live, however, this is no longer the case. Any miscues at this point will almost certainly translate into direct operational and/or financial impacts on the Performing Organization. It is through the careful planning, execution, and management of System Implementation activities that the Project Team can minimize the likelihood of these occurrence, and determine appropriate contingency plans in the event of a problem.

This phase consists of the following processes:

Prepare for the System Implementation: where all steps needed in advance of actually deploying the

application is performed, including preparation of both the production environment and the consumer communities.

Deploy System: where the full deployment plans are initially developed during System Design and evolved throughout subsequent lifecycle phases, is executed and validated.

Transition to Performing Organization: where the responsibility for and ownership of the application are transitioned from the Project Team to the unit in the Performing Organization that will provide system support and maintenance.

**6.2 ALGORITHM FOR THE SYSTEM**

**6.2.1 VIDEO CONTROL**

START: Start Web application and start webcam.

STEP 1: Detect the user’s hand.

STEP 2: Capture the image

STEP 3: Detect the hand landmark and lmList of hand and identify the specific hand gesture.

STEP 4: If gesture for play/pause is detected go to step 9.

STEP 5: If gesture for volume up/volume down is detected go to step 10.

STEP 7: If gesture for forward/rewind is detected go to step 11.

STEP 8: If gesture for full screen is detected go to step 12.

STEP 9: Pyautogui.press(‘space’) and go to step 3.

STEP 10: Pyautogui.press(‘up’) / Pyautogui.press(‘down’) and go to step 3.

STEP 11: Pyautogui.press(‘right) / Pyautogui.press(‘left) and go to step 3.

STEP 12: Pyautogui.press(‘f’) and go to step 3.

STEP 13: Stop.

Here we have coordinates for all of the hand landmarks, we will use them to detect different hand gestures and the first of them is detecting whether the fist is open or closed. For that, we will compare the coordinates of tips of fingers [8, 12, 16, 20] and middle points [6, 10, 14, 19] and if the fingertips are below the middle points, then the first is closed and vice versa. Next get the total number of fingers counted and save it in a variable. for id in range (1, 5): if lmList[tipIds[id]][2] < lmList[tipIds[id] - 2][2]:

fingers.append(1)

if (lmList[tipIds[id]][2] > lmList[tipIds[id] - 2][2]): fingers.append(0)

**6.2.2 POWER POINT CONTROL**

START: Start Web application and start webcam.

STEP 1: Detect the user’s hand.

STEP 2: Capture the image

STEP 3: Detect the hand landmark of hand gesture and identify the specific hand gesture.

STEP 4: If gesture for next slide is detected go to step 10.

STEP 5: If gesture for Previous slide is detected go to step 11.

STEP 7: If gesture for first slide is detected go to step 12.

STEP 8: If gesture for full screen is detected go to step 13.

STEP 9: If gesture for exit full screen is detected go to step 14.

STEP 10: Pyautogui.press(‘right) and go to step 3.

STEP 11: Pyautogui.press(‘left) and go to step 3.

STEP 12: Pyautogui.press(‘home) and go to step 3.

STEP 13: Pyautogui.press([‘fn’,’f5’]) and go to step 3.

STEP 14: Pyautogui.press(‘esc’) and go to step 3.

STEP 15: Stop.

**6.2.3 IMAGE CONTROL**

START: Start Web application and start webcam.

STEP 1: Detect the user’s hand.

STEP 2: Capture the image

STEP 3: Detect the hand landmark of hand gesture and identify the specific hand gesture.

STEP 4: If gesture for next image is detected go to step 10.

STEP 5: If gesture for Previous image is detected go to step 11

STEP 7: If gesture for zoom in is detected go to step 12.

STEP 8: If gesture for zoom out is detected go to step 13.

STEP 9: If gesture for rotate is detected go to step 14.

STEP 10: Pyautogui.press(‘right) and go to step 3.

STEP 11: Pyautogui.press(‘left) and go to step 3.

STEP 12: Pyautogui.press([‘ctrl’,’=’]) and go to step 3.

STEP 13: Pyautogui.press([‘ctrl,’-’]) and go to step 3.

STEP 14: Pyautogui.press([‘ctrl,’r’]) and go to step 3.

STEP 15: Stop

### 7.SYSTEM TESTING

**7.1 INTRODUCTION**

System testing is the major quality control measure during software development. Testing is a set

activity that can be planned and conducted schematically. Testing begins at the module level and work towards the integration of entire computer-based system. Testing is a process of executing a program with the intention of finding an error. A good test case is one that has a higher probability of finding an undiscovered error. A successful test case is one that uncovers an undiscovered error. Testing phase in the “Multimedia control using hand gestures” is supposed to verify that the system does exactly what it is designed to do. The system is to be tested with the data at the extremes of the input range. This system is also to be tested for various values outside the input range. In the system that provides different validity test strategies to validate the textboxes, entries in the system. Also, it can check the system efficiency in terms of their input and output data's.

**7.2 LEVELS OF TESTING**

**7.2.1 UNIT TESTING**

A level of the software testing process where individual units of a software are tested. The purpose is to validate that each unit of the software performs as designed. The first level of testing, unit testing, is the most micro-level of testing. It involves testing individual modules or pieces of code to make sure each part or “unit” is correct. A “unit” can be a specific piece of functionality, a program, or a particular procedure within the application. Unit testing helps verify internal design and internal logic, internal paths, as well as error handling. The unit testing level includes a single type of testing; unit testing. Unit tests are done by the developer who wrote the code.

**7.2.2 INTEGRATION TESTING**

A level of the software testing process where individual units are combined and tested as a group.

The purpose of this level of testing is to expose faults in the interaction between integrated units.

Integration testing is done after unit testing. This level tests how the units work together. Individual modules are combined and tested as a group. It’s one thing if units work well on their own, but how do they perform together? Integration testing helps you determine that, and ensures your application runs efficiently. It identifies interface issues between modules. There are a few techniques that can be used for conducting integration testing:

* Big Bang Testing
* Top-Down Approach
* Bottom-Up Approach

The Big bang testing involves testing the entire set of integrated components together simultaneously. Because everything is integrated together and being tested at one time, this approach makes it difficult to identify the root cause of problems. The top-down approach starts by testing the top-most modules and gradually moving down to the lowest set of modules oneby-one. The bottom-up approach starts with testing the lowest units of the application and gradually moving up one-by-one.

**7.2.3 ACCEPTANCE TESTING**

A level of the software testing process where a system is tested for acceptability. The purpose of this

test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. The final level of testing, acceptance testing, or UAT (user acceptance testing), determines whether or not the software is ready to be released. Let’s face it, requirements change throughout the development process. It’s important that the user verifies the business needs are met before the software is released into production.

Are the functional requirements met? Are the performance requirements met? These are the questions that are answered during acceptance testing level. UAT is the final say as to whether the application is ready for use in real life or not. This phase also involves change control managing requested modifications and new feature requests. Acceptance testing should be done by the business user / end-user.

**7.2.4 SYSTEM TESTING**

System testing validates the “Multimedia control using hand gestures” once it has been incorporated

into a large system. System testing is actually a series of different tests whose primary purpose is fully exercise the computer-based system. All work to verify that “Multimedia control using hand gestures” elements have been properly integrated and perform allocated function. They can check the functioning of processes with respect to their input data. Also “Multimedia control using hand gestures” that test the system validity in a user-friendly manner. Recovery testing is a system test that for the software to fail a variety of ways and verifies that recovery testing is properly performed. Security testing attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper penetration. Stress testing executes a system in a manner that demands resources in abnormal quantity, frequency or volume. Performance testing is designed to test the run time performance of software within the context of an integrated system.

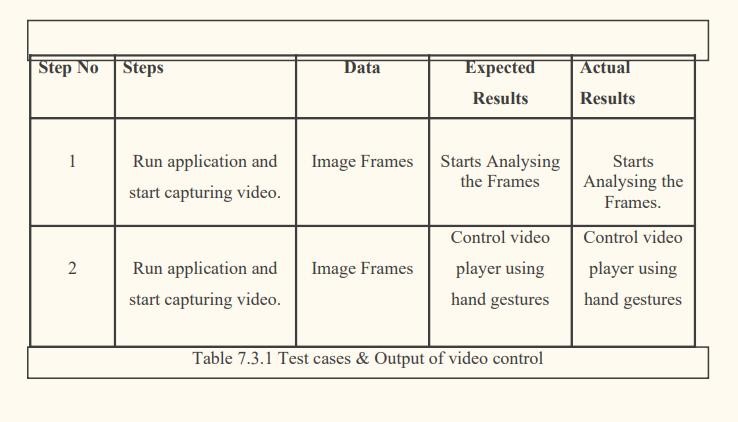
**7.3TEST CASES**

**7.3.1 VIDEOPLAYER**

|  |
| --- |
| To check whether the video player is controlled by using hand gestures or not |

Test objectives:

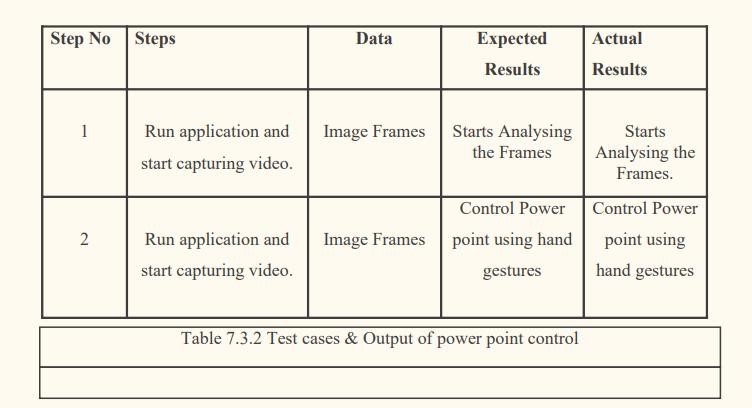
Test Data: Webcam Live Video



**7.3.2 POWEREPOINT**

Test Objectives: To check whether power point is controlled by using hand gestures or not

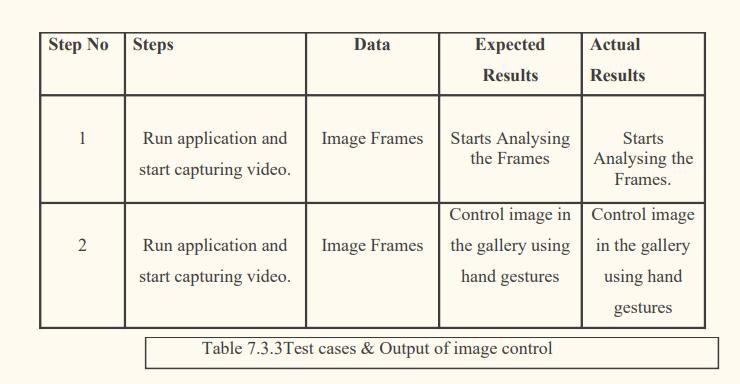
Test Data: Webcam Live Video



**7.3.3 IMAGE**

Test Objectives: To check whether the image is controlled by using hand gestures or not

Test Data: Webcam Live Video



### 8.CONCLUSION

The new system has overcome most of the limitations of the existing system and works according to the design specification given. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information. All the requirements projected by the user have been met by the system.

The project describes a Web application that controls Multi medias like video, power point, image etc. with the help of hand gestures. The method proposed here successfully created a hand gesture recognition system, that is able to recognise which gesture is performed by the user and accurately perform the functionality associated with it. The project uses only webcam, would completely eliminate the keyboard and mouse. Also, this would lead to a new era of Human Computer Interaction (HCI) where no physical contact with the device is required.

### 9.FUTURE ENHANCEMENT

To overcome the drawbacks of the current system, we can modify it for better. we can integrate iris detection to this project to make it run more smoothly. this project can be extended to other public service technical systems to avoid direct contact. ATM machines, Ticket Counters, etc can make use of the extended version.

### 10. REFERENCES

1. P. Premaratne, “Historical development of hand gesture recognition”, in Human Computer Interaction Using Hand Gestures. Cognitive Science and Technology. Singapore: Springer, 2014, pp. 5–29.
2. A. Joshi, S. Ghosh, M. Betke, S. Sclaro, H. Pfister, “Personalizing gesture recognition using hierarchical bayesian neural networks”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6513–6522.

[3]. Ruize Xu, Shengli Zhou, Wen J. Li., “MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition”. IEEE, 2012. Vol: 12, 1166–1173.

[4]. N. Krishna Chaitanya, R. Janardhan Rao, “Controlling of Windows Media Player Using Hand Gesture Recognition System”, IJES, 2014, Vol: 3, 2319–1.

[5] Hand in Hand: Automatic Sign Language to English Translation, by Daniel Stein, Philippe Dreuw, Hermann Ney and Sara Morrissey, Andy Way

**WEBSITES:**

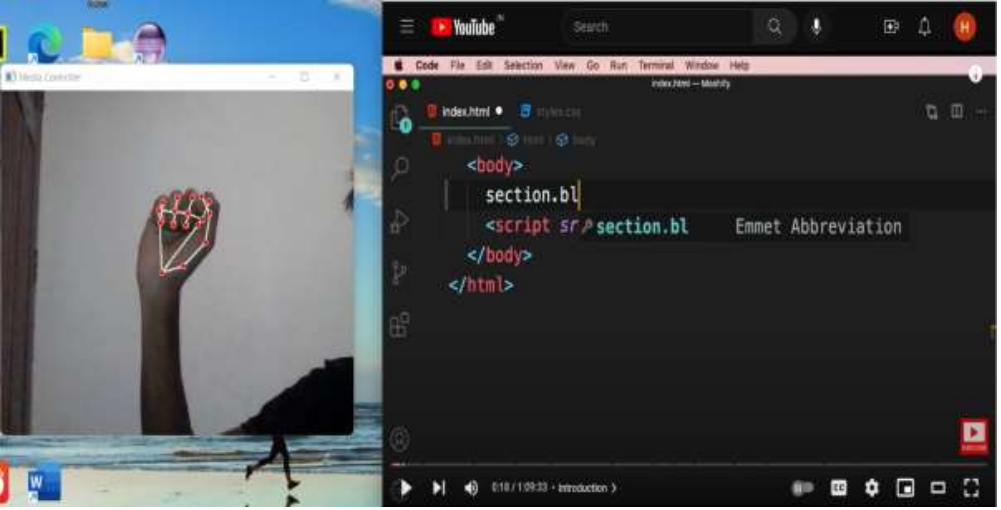
[1] <https://doi.org/10.1109/CVPR.2017.56>

[2] [www.learnpython.org](http://www.learnpython.org)

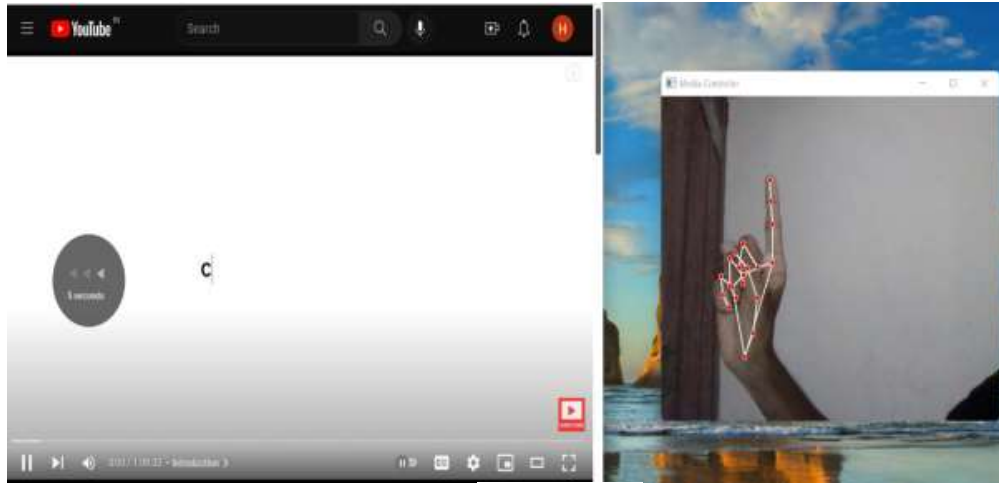
[3] https://doi.org/10.1007/978-981-4585-69-9\_2

## 11.APPENDIX

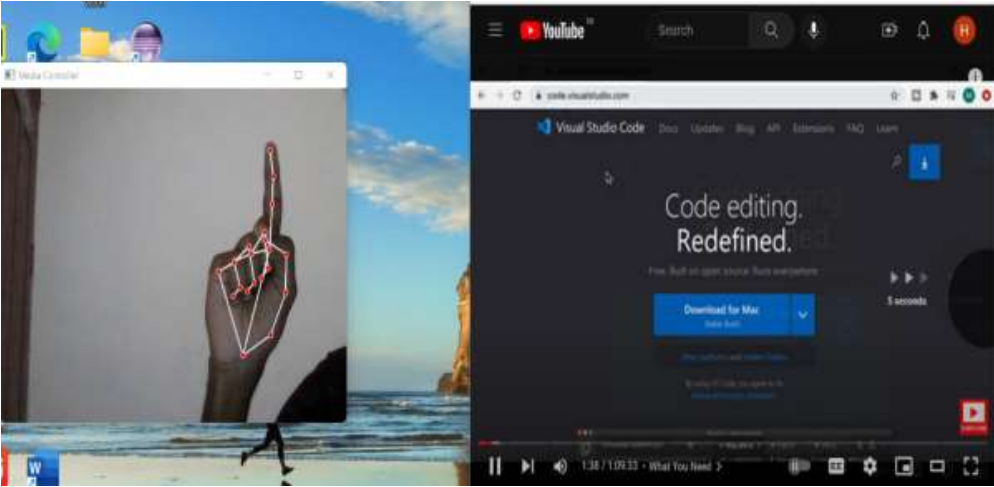
**11.1 SCREENSHOTS: VIDEO CONTROL USING HAND GESTURES**



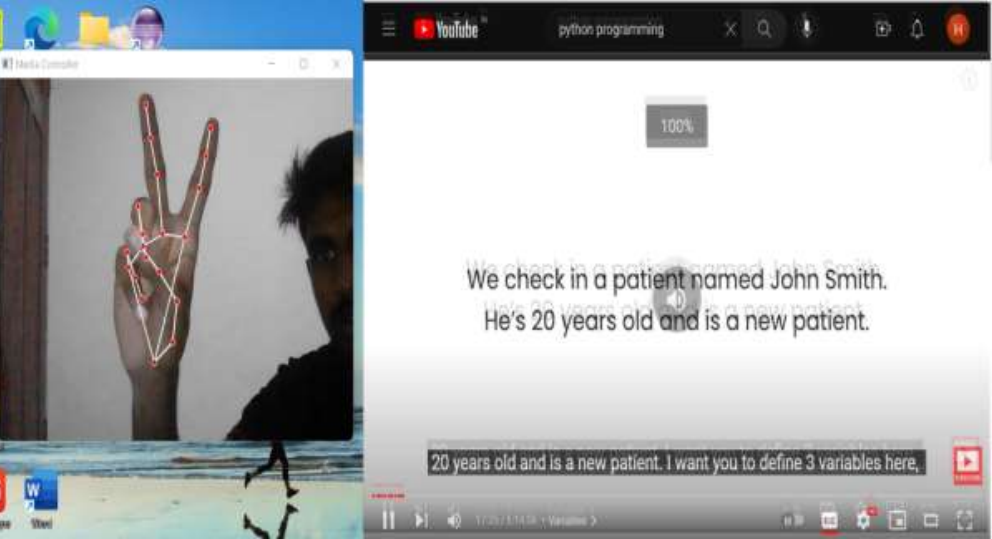
**PLAY/PAUSE**



**REWIND**

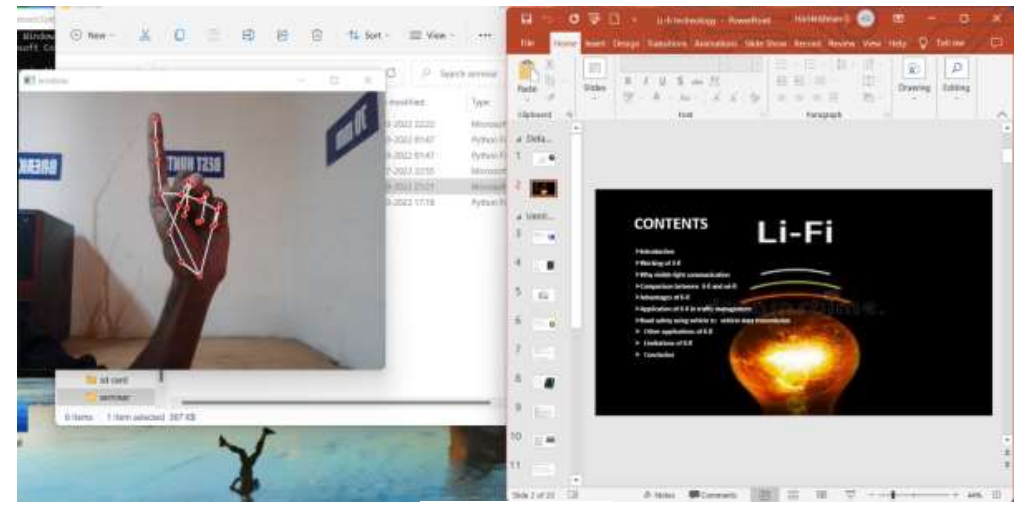


**FORWARD**

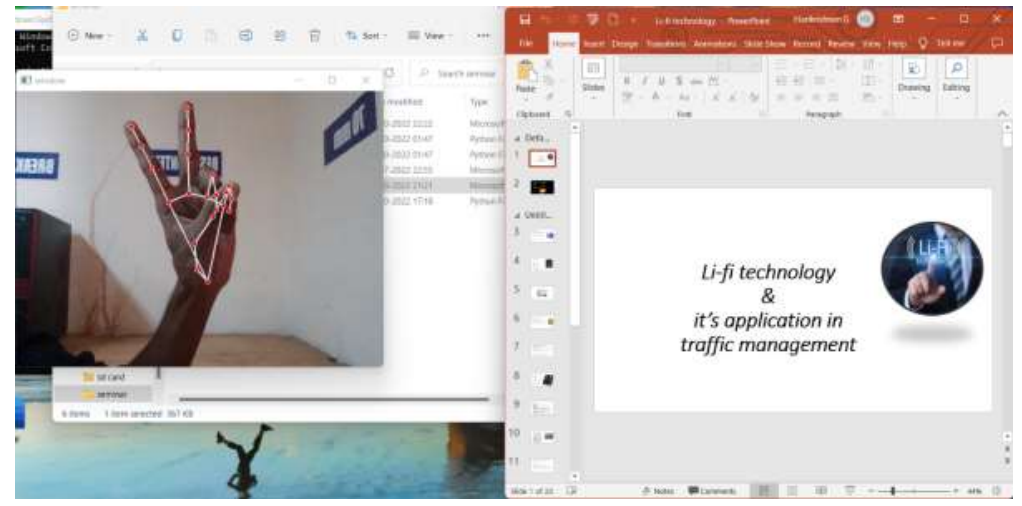


**VOLUME UP**

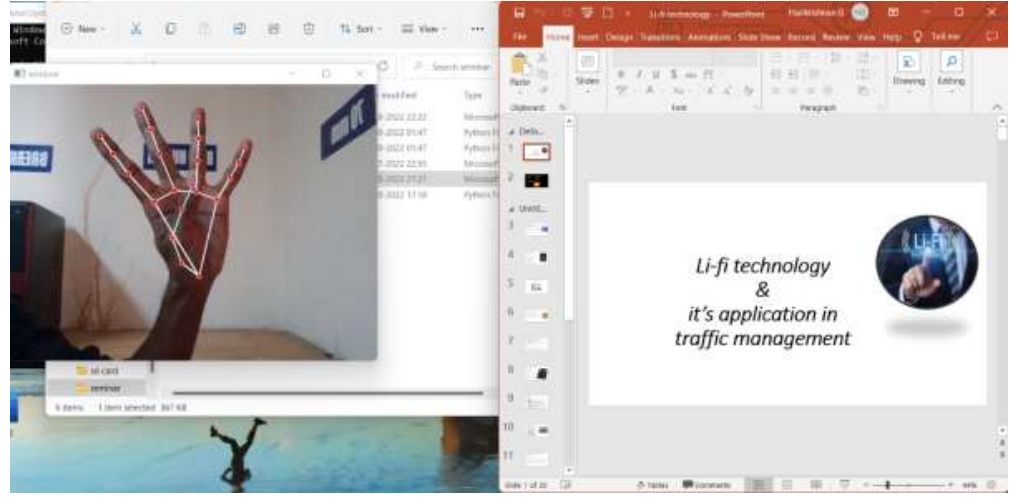
**SCREENSHOTS: POWERPOINT CONTROL USING HAND GESTURES**



**NEXT SLIDE**

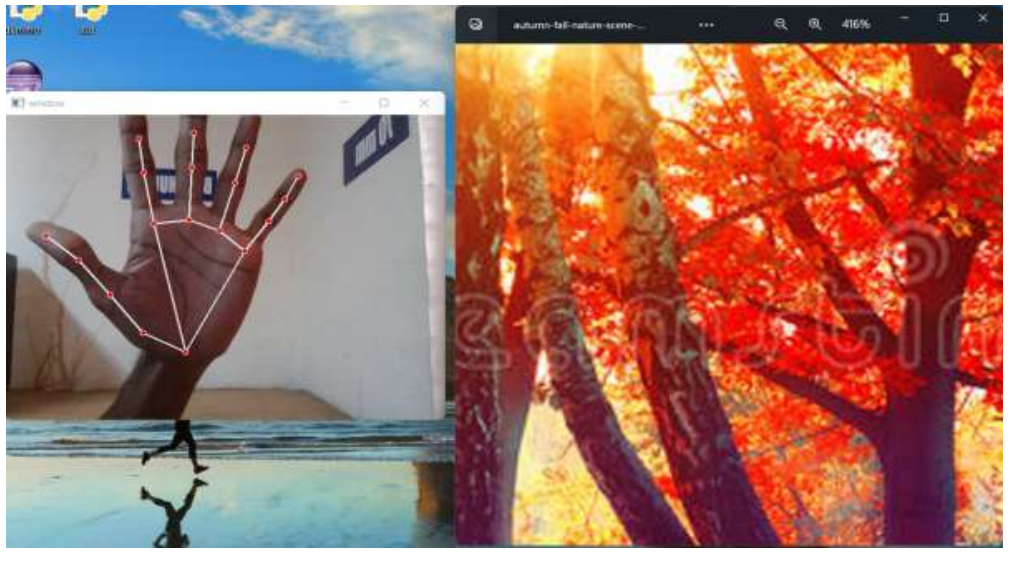


**PREVIOUS SLIDE**

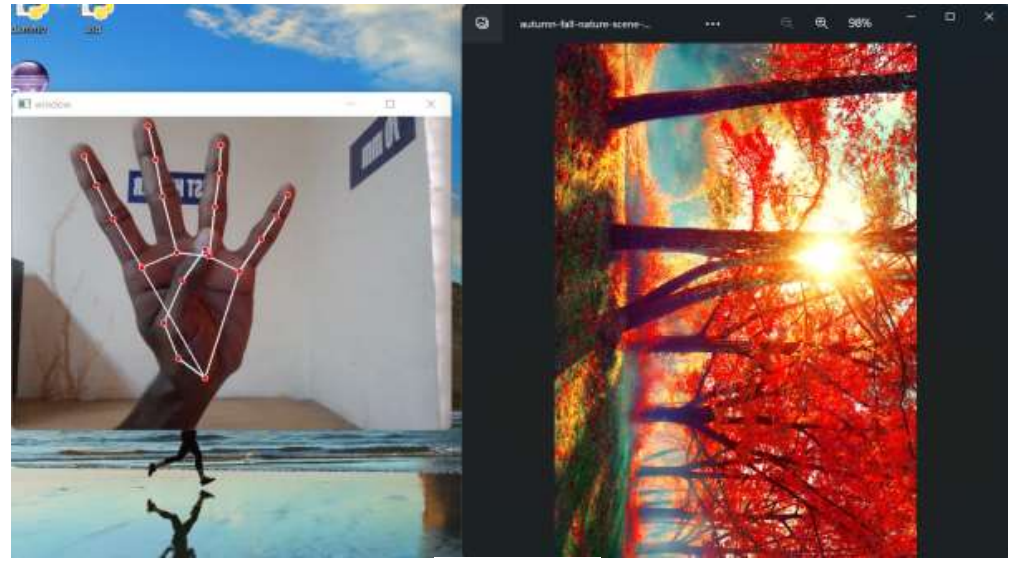


**EXIT FULLSCREEN**

**SCREENSHOTS: IMAGE CONTROL USING HAND GESTURES**



**ZOOM IN**



**ROTATE**



**ZOOM OUT**

**11.2 CODING**

HOME PAGE

import streamlit as st

from PIL import Image

import base64

st.set\_page\_config(

page\_title="Gesture Controlling") st.sidebar.success("select a page above")

def add\_bg\_from\_local(image\_file): with open(image\_file, "rb") as image\_file:

encoded\_string = base64.b64encode(image\_file.read()) st.markdown(

f"""

<style>

.stApp {{ background-image: url(data:image/{"png"};base64,{encoded\_string.decode()}); background-size: cover

}}

</style>

""",

unsafe\_allow\_html=True

)

add\_bg\_from\_local("D:/1.my project/file control/istockphoto-1301592082-170667a.jpg")

st.title("MULTI MEDIA CONTROL USING HAND GESTURES")

html\_temp2 = """

<body style="background-color:white;padding:10px;">

<h3 style="color:#0000FF ;text-align:left;">About Web App</h3>

The Main aim of this application is to use the most natural form i.e., Hand gestures to interact with the computer system. These gestures are implemented in such a way that they are easy to perform, fast, efficient and ensuring an immediate response.

The application uses your device's camera to give you touch-free and remote-free control over your media player application ,Images and Powerpoint presentation etc..

(without any special hardware).

</body>

<div style="background-color:black;padding:10px;margin-bottom:10px;">

<h4 style="color:white;">Prepared using:</h4>

<ul style="color:#FFFF00;">

<li>Opencv </li>

<li>mediapipe </li> <li>Streamlit </li>

<li>PyAutoGui </li>

</ul>

</div>

"""

st.markdown(html\_temp2, unsafe\_allow\_html=True)

st.markdown("1. Video control \n"

"2. PowerPoint control \n"

"3. Image control")

image = Image.open("D:/1.my project/model.png")

st.image(image, caption='21 Hand Landmarks ',width=700, use\_column\_width=700, clamp=False, channels='RGB', output\_format='auto')

st.sidebar.title("Made By:") html\_temp3 = """

<ul style="font-weight:bold;">

<li>Meera </li>

<li>Midhun </li>

<li>Sandra</li>

<li>Alen</li>

</ul>

"""

st.sidebar.markdown(html\_temp3, unsafe\_allow\_html=True)

html\_temp4 = """

<body style="background-color:white;padding:5px;">

<h3 style="color:#f63366 ;text-align:center;">Control video using Hand Gestures </h3>

<ul> Gestures and their Function

<li>one: forward/rewind </li>

<li>two : Volume up/Volume down</li>

<li>three :Full Screen</li>

<li>Zero:Play/Pause</li>

<li> No Hand : No gesture: No action </li>

<h4 style="font-weight:bold;"><li>Press Button "q" to Exit</li></h4>

</ul>

"""

st.markdown(html\_temp4, unsafe\_allow\_html=True) html\_temp5 = """

<body style="background-color:white;padding:5px;">

<h3 style="color:#f63366 ;text-align:center;">Control PowerPoint using Hand Gestures </h3>

<ul> Gestures and their Function

<li>one: next slide </li>

<li> two: previous slide</li>

<li>three: home</li>

<li>four: exit full screen</li>

<li>five :full screen</li>

<li> No Hand : No gesture: No action </li>

<h4 style="font-weight:bold;"><li>Press Button "q" to Exit</li></h4>

</ul>

"""

st.markdown(html\_temp5, unsafe\_allow\_html=True)

html\_temp6 = """

<body style="background-color:white;padding:5px;">

<h3 style="color:#f63366 ;text-align:center;">Control Image using Hand Gestures </h3>

<ul> Gestures and their Function

<li>one: next slide </li>

<li>two : previous slide</li>

<li>three :exit full screen</li>

<li>four :rotate </li>

<li>five :full screen </li>

<li> No Hand : No gesture: No action </li>

<h4 style="font-weight:bold;"><li>Press Button "q" to Exit</li></h4>

</ul>

"""

st.markdown(html\_temp6, unsafe\_allow\_html=True)

2.VIDEO

import streamlit as st import cv2 import mediapipe as mp import pyautogui import webbrowser

#st.title("Media player control using hand gesture") html\_temp7 = """

<body style="background-color:white;padding:5px;">

<h3 style="color: #FF0000 ;text-align:left;">Mediaplayer Control Using Hand Gestures </h3>

<ul> Gestures and their Function

<li>one: next slide </li>

<li>two : previous slide</li>

<li>three :exit full screen</li>

<li>four :full screen </li>

<li> No Hand : No gesture: No action </li>

<h4 style="font-weight:bold;"><li>Press Button q to Exit</li></h4>

</ul>

"""

st.markdown(html\_temp7, unsafe\_allow\_html=True) html\_temp5 = """

<body style="background-color:white;padding:5px;">

<h3 text-align:left;">Demo Video</h3>

"""

st.markdown(html\_temp5, unsafe\_allow\_html=True) st.video("D:/pictures/Videos/VID-20220830-WA0001.mp4")

p="D:"

r=st.button("select file")

if r:

webbrowser.open\_new(p)

run=st.button("click") if run:

mp\_drawing = mp.solutions.drawing\_utils mp\_hands = mp.solutions.hands ##################################

tipIds = [4, 8, 12, 16, 20] state = None

Gesture = None

############################ def fingerPosition(image, handNo=0):

lmList = [] if results.multi\_hand\_landmarks:

myHand = results.multi\_hand\_landmarks[handNo] for id, lm in enumerate(myHand.landmark):

# print(id,lm) h, w, c = image.shape cx, cy = int(lm.x \* w), int(lm.y \* h) lmList.append([id, cx, cy]) return lmList

# For webcam input: cap = cv2.VideoCapture(0) with mp\_hands.Hands( min\_detection\_confidence=0.8, min\_tracking\_confidence=0.5) as hands:

while cap.isOpened():

success, image = cap.read()

if not success:

print("Ignoring empty camera frame.")

# If loading a video, use 'break' instead of 'continue'.

continue

# Flip the image horizontally for a later selfie-view display, and convert

# the BGR image to RGB. image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR\_BGR2RGB) image.flags.writeable = False results = hands.process(image)

# Draw the hand annotations on the image.

image.flags.writeable = True image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR) if results.multi\_hand\_landmarks: for hand\_landmarks in results.multi\_hand\_landmarks:

mp\_drawing.draw\_landmarks( image, hand\_landmarks, mp\_hands.HAND\_CONNECTIONS) lmList = fingerPosition(image)

#print(lmList) if len(lmList) != 0:

fingers = [] for id in range(1, 5): if lmList[tipIds[id]][2] < lmList[tipIds[id] - 2][2]:

fingers.append(1) if (lmList[tipIds[id]][2] > lmList[tipIds[id] - 2][2] ):

fingers.append(0) totalFingers = fingers.count(1) print(totalFingers)

#print(lmList[9][2])

if totalFingers == 4: state = "Play"

if totalFingers == 0 and state == "Play":

state = "Pause" pyautogui.press('space') print("Space") if totalFingers == 1: if lmList[8][1]<300: print("left") pyautogui.press('left') if lmList[8][1]>400: print("Right") pyautogui.press('Right') if totalFingers == 2: if lmList[9][2] < 210: print("Up") pyautogui.press('Up') if lmList[9][2] > 230: print("Down") pyautogui.press('Down') if totalFingers == 3 and state == "Play":

state = "fullscreen" pyautogui.press('f') print("fullscreen")

cv2.imshow("Media Controller", image) if cv2.waitKey(1) == 113: cv2.destroyAllWindows()

cap.release() break

3.POWERPOINT

import streamlit as st import cv2 import mediapipe as mp import pyautogui import webbrowser import time

#st.title("Presentation control ")

html\_temp6 = """

<body style="background-color:white;padding:5px;">

<h3 style="color:#800080 ;text-align:left;">Presentation Control Using Hand Gestures </h3>

<ul> Gestures and their Function

<li>one: next slide </li>

<li>two: previous slide</li>

<li>three: home</li>

<li>four: exit full screen</li>

<li>five: full screen</li>

<li> No Hand: No gesture: No action </li>

<h4 style="font-weight: bold;"><li>Press Button "q" to Exit</li></h4>

</ul>

"""

st. markdown(html\_temp6, unsafe\_allow\_html=True)

p="D:/seminar/Li-fi technology.pptx" r=st.button("select file")

if r:

webbrowser.open\_new(p)

run=st.button("click")

if run: def count\_fingers(lst): cnt = 0

thresh = (lst.landmark[0].y\*100 - lst.landmark[9].y\*100)/2

if (lst.landmark[5].y\*100 - lst.landmark[8].y\*100) > thresh:

cnt += 1

if (lst.landmark[9].y\*100 - lst.landmark[12].y\*100) > thresh:

cnt += 1

if (lst.landmark[13].y\*100 - lst.landmark[16].y\*100) > thresh:

cnt += 1

if (lst.landmark[17].y\*100 - lst.landmark[20].y\*100) > thresh:

cnt += 1

if (lst. landmark[5].x\*100 - lst. landmark[4].x\*100) > 6:

cnt += 1

return cnt

cap = cv2.VideoCapture(0)

drawing = mp.solutions.drawing\_utils hands = mp.solutions.hands hand\_obj = hands.Hands(max\_num\_hands=1)

start\_init = False

prev = -1

while True:

end\_time = time.time() \_, frm = cap.read() frm = cv2.flip(frm, 1)

res = hand\_obj.process(cv2.cvtColor(frm, cv2.COLOR\_BGR2RGB))

if res.multi\_hand\_landmarks:

hand\_keyPoints = res.multi\_hand\_landmarks[0]

cnt = count\_fingers(hand\_keyPoints)

if not(prev==cnt): if not(start\_init): start\_time = time.time() start\_init = True

elif (end\_time-start\_time) > 0.2:

if (cnt == 1):

pyautogui.press("right")

elif (cnt == 2):

pyautogui.press("left")

elif (cnt == 5):

pyautogui.press(["fn","f5"])

elif (cnt == 4):

pyautogui.press("esc")

elif (cnt == 3):

pyautogui.press("home")

prev = cnt start\_init = False

drawing.draw\_landmarks(frm, hand\_keyPoints, hands.HAND\_CONNECTIONS)

cv2.imshow("window", frm) if cv2.waitKey(1) == 113: cv2.destroyAllWindows() cap.release() break

4.IMAGE

import streamlit as st import cv2 import mediapipe as mp import pyautogui import webbrowser import time from PIL import Image

html\_temp6 = """

<body style="background-color:white;padding:5px;">

<h3 style="color:#800080 ;text-align:left;">gallery image Control Using Hand Gestures </h3>

<ul> Gestures and their Function

<li>one: right </li>

<li>two : left </li>

<li>three :zoom out</li>

<li>Four :full screen </li>

<li>Five :zoom in </li>

<li> No Hand : No gesture: No action </li>

<h4 style="font-weight:bold;"><li>Press Button "q" to Exit</li></h4>

</ul>

"""

st.markdown(html\_temp6, unsafe\_allow\_html=True)

image = Image.open("D:/1.my project/model.png")

st.image(image, caption='21 Hand Landmarks',width=700, use\_column\_width=700, clamp=False, channels='RGB', output\_format='auto')

p="D:/photos/ptoto 1/images (2).jfif" r=st.button("select file")

if r:

webbrowser.open\_new(p)

run=st.button("click") if run: def count\_fingers(lst):

cnt = 0

thresh = (lst.landmark[0].y\*100 - lst.landmark[9].y\*100)/2

if (lst.landmark[5].y\*100 - lst.landmark[8].y\*100) > thresh:

cnt += 1

if (lst.landmark[9].y\*100 - lst.landmark[12].y\*100) > thresh:

cnt += 1

if (lst.landmark[13].y\*100 - lst.landmark[16].y\*100) > thresh:

cnt += 1

if (lst.landmark[17].y\*100 - lst.landmark[20].y\*100) > thresh:

cnt += 1

if (lst.landmark[5].x\*100 - lst.landmark[4].x\*100) > 6:

cnt += 1

return cnt

cap = cv2.VideoCapture(0)

drawing = mp.solutions.drawing\_utils

hands = mp.solutions.hands hand\_obj = hands.Hands(max\_num\_hands=1)

start\_init = False

prev = -1

while True:

end\_time = time.time() \_, frm = cap.read() frm = cv2.flip(frm, 1)

res = hand\_obj.process(cv2.cvtColor(frm, cv2.COLOR\_BGR2RGB))

if res.multi\_hand\_landmarks:

hand\_keyPoints = res.multi\_hand\_landmarks[0]

cnt = count\_fingers(hand\_keyPoints)

if not(prev==cnt): if not(start\_init):

start\_time = time.time()

start\_init = True

elif (end\_time-start\_time) > 0.2:

if (cnt == 0):

print(0)

if (cnt == 1):

pyautogui.press("right")

print("right")

elif (cnt == 2): pyautogui.press("left")

print("left")

elif (cnt == 5): with pyautogui.hold("ctrl"):

pyautogui.press("=",presses=5) print("zoom in") elif (cnt == 3): with pyautogui.hold("ctrl"):

pyautogui.press("-",presses=5) print("zoom out") if (cnt == 4):

pyautogui.hotkey("ctrl","r") print("rotate")

prev = cnt start\_init = False

drawing.draw\_landmarks(frm, hand\_keyPoints, hands.HAND\_CONNECTIONS)

cv2.imshow("window", frm)

if cv2.waitKey(1) == 113: cv2.destroyAllWindows() cap.release()

break