

AN IMS [ISO 21001, ISO 9001, ISO 14001, ISO 45001] CERTIFIED TRAINING INSTITUTION

NETTUR TECHINICAL TRAINING FOUNDATION

PROJECT REPORT

On

GESTURE CONTROLLED MEDIA PLAYER (GCMP)

SUBMITTED BY

ALAPATI PRAKASH GUPTHA - NEC0818003

KASULA MAHENDRA VINITH KUMAR - NEC0818028

SANKA GOPI KRISHNA - NEC0818052

RAHUL L N - NEC0818074

*In Partial Fulfillment of The Requirements*

*For The Award of The Diploma*

*In*

COMPUTER ENGINEERING AND IT INFRASTRUCTURE

NTTF ELECTRONICS CITY, BANGALORE - 560100

2018 - 2021



AN IMS [ISO 21001, ISO 9001, ISO 14001, ISO 45001] CERTIFIED TRAINING INSTITUTION

NETTUR TECHINICAL TRAINING FOUNDATION

PROJECT REPORT

On

GESTURE CONTROLLED MEDIA PLAYER (GCMP)

PROJECT DONE BY

ALAPATI PRAKASH GUPTHA - NEC0818003

KASULA MAHENDRA VINITH KUMAR - NEC0818028

SANKA GOPI KRISHNA - NEC0818052

RAHUL L N - NEC0818074

DIPLOMA IN COMPUTER ENGINEERING AND IT INFRASTRUCTURE

NTTF ELECTRONICS CITY, BANGALORE - 560100

2018 – 2021



AN IMS [ISO 21001, ISO 9001, ISO 14001, ISO 45001] CERTIFIED TRAINING INSTITUTION

NETTUR TECHINICAL TRAINING FOUNDATION

CERTIFICATE

This is to Certify that the Project Titled

GESTURE CONTROLLED MEDIA PLAYER (GCMP)

Is a Bonafide report of Project done By

ALAPATI PRAKASH GUPTHA - NEC0818003

KASULA MAHENDRA VINITH KUMAR - NEC0818028

SANKA GOPI KRISHNA - NEC0818052

RAHUL L N - NEC0818074

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DIPLOMA IN COMPUTER ENGINEERING & IT INFRASTRUCTURE UNDER THE INSTITUTION NETTUR TECHNICAL TRAINING FOUNDATION, ELECTRONICS CENTER, BANGALORE DURING THE ACADEMIC YEAR 2018-2021.

CLASS CO-ORDINATOR PROJECT GUIDE

HEAD EXTERNAL EXAMINER

(NTTF SCHOOL OF COMPUTER SCIENCES)

**ACKNOWLEDGEMENT**

First, we would like to express our heartfelt gratitude to NTTF (Nettur Technical Training Foundation) Electronics Training Centre, Bengaluru for giving us the opportunity to undertake the final year project.

We would like to take this opportunity to mention a few names, without whom, our project would not be successful. To begin with, we would like to thank our Head NTTF school of computing sciences, **Mr. Ramanathan S** and **Mr. V Rajasekaran,** principal NEC, for their constant encouragement and words which pushed us to do better.

We are thanked to our program manager **Mr. Jagdish G K** and our Course head of CP08 **Mrs. Dharmambal** for being a source of inspiration throughout our project. We are great to **Mr. Liju Gangadharan** project coordinator for his greater support to be making our project work.

Once more we are indebted and grateful to **Mrs. Menaka A** our project guide for his valuable guidance for each stage of our project as reality.

With the deep sense and gratitude, we express our heartfelt thanks to all other staff members who gave valuable suggestions and constant encouragement for successful fulfillment of the project.

**ABSTRACT**

Human Computer Interaction can acquire several advantages with the introduction of different natural forms of device free communication. Gestures are a natural form of actions which we often use in our daily life for interaction, therefore, to use it as a communication medium with computers generates a new paradigm of interaction with computers. This paper implements computer vision and gesture recognition techniques and develops a vision based low-cost input device for controlling the VLC player through gestures. This hand gesture recognition and control technique will not only replace the use of mouse to control the VLC player but also provide different gesture vocabulary which will be useful in controlling other applications.

**INDEX (SOFTWARE)**

**SLNO.**   **CONTENT PAGE NO**

**CHAPTER - I**  **INTRODUCTION**

1.1 Introduction\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.2 Objective of the Project\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.3 System Study\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.3.1 Study on Existing System\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.3.2 Demerits of Existing System\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.3.3 Study on Proposed System\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.3.4 Merits of Proposed System\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.3.5 Literature Survey\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHAPTER – II**  **SYSTEM ANALYSIS**

2.1 User Requirement Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.2 Software Requirement Specification (SRS)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.3 System Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.3.1 Hardware Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.3.2 Software Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.4 Final Outline of Proposed System\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHAPTER – III DESIGN AND DEVELOPMENT PROCESS**

3.1 Fundamental Design Concepts\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.1.1 Modules and Functionalities \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.1.2 Description of Modules \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2 Design Notations \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2.1 DFD Symbols \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2.2 Data Flow Diagram \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3 Design Process \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3.1 Input Design \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3.2 Table Design \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3.3 Output Design \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.4 Development Approach \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.4.1 Development Model / Approach \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHAPTER – IV PLANNING**

4.1 Gantt Chart \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHAPTER – V TESTING AND IMPLEMENTATION**

5.1 Testing \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.1.1 Testing and Methodologies \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.1.2 Test Cases \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.2 System Implementation \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.2.1 Implementation Procedures \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHAPTER – VI** **CONCLUSION**

6.1 Conclusion \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6.2 Scope for Future Enhancements \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ANNEXURE**- A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sample Code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**BIBLIOGRAPHY** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHAPTER 1**

**INDRODUCTION**

**1.1 INTRODUCTION**

As we know, the vision-based technology of hand gesture recognition is an important part of human-computer interaction (HCI). In the last decades, keyboard and mouse play a significant role in human-computer interaction. However, owing to the rapid development of hardware and software, new types of HCI methods have been required. Technologies such as speech recognition and gesture recognition receive great attention in the field of HCI.

Gesture is a symbol of physical behavior or emotional expression. It includes body gesture and hand gesture. It falls into two categories: static gesture and dynamic gesture. For the former, the posture of the body or the gesture of the hand denotes a sign. For the latter, the movement of the body or the hand conveys some messages. Gesture can be used as a tool of communication between computer and human. It is different from the traditional hardware-based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user intent through the recognition of the gesture or movement of the body or body parts. In the past decades, many researchers have strived to improve the hand gesture recognition technology. Hand gesture recognition has immense value in many applications such as sign language recognition, augmented reality (virtual reality), sign language interpreters for the disabled, robot and app control.

The workflow of Hand Gesture Control is described as follows. First, the hand region is detected from the original images from the input devices using OpenCV. Then, using the input video from the webcam the region of hand is identified and segmented into hand landmarks using MediaPipe framework. Then using Python Programming the recognition of hand gestures is accomplished by measuring the x & y values provided by certain landmarks of the hand by MediaPipe [Figure 1]. The input devices providing the original image information includes normal camera, webcam, and ToF (time of flight) camera. The ToF camera additionally provide the depth information, so it is easy to segment the hand region from the background in terms of the depth map. At last, the integration and control of VLC media player using hand gestures will be done by using an API called PyAutoGUI.

Text

Description automatically generatedIn this paper, we present an efficient and effective method for hand gesture recognition and Control. The hand region is detected. Then, the palm and fingers are segmented into landmarks to recognize hand gesture which can be classified by the x & y values of specific landmarks. Using these gestures, specific commands for specific gestures are assigned to VLC media player using PyAutoGUI API.

[Figure 1]

**1.2 OBJECTIVES OF OUR PROJECT**

* The scope of the project is to create a simple hand gesture control model using MediaPipe framework, Python, CV2, PyAutoGUI API etc.
* High priority of the project is it can work in any system regardless of using any special hardware.
* Using this procedure new forms HCI can be developed and it can be applied across all devices and control them.
* This project can be used to develop a model to detect the sign language and control a robot.

**1.3 SYSTEM STUDY**

**1.3.1 STUDY ON EXISTING SYSTEM**

Gesture Control technology is implemented in diverse ways like using some specialized hardware such as motion tracking sensors to track the motion. This very feature was implemented in a smartphone device “Pixel 4” made by Google and termed the feature as “Motion Sense”, they made a microscopic electronic chip called “SOLI”. It uses miniature radar for real-time motion tracking of the human hand and it can track sub-millimeter motion at high speeds with great accuracy. So, using this Pixel 4 will detect the motion of hands and can control certain app in real-time.

**1.3.2 DEMERITS OF EXISTING SYSTEM**

1. Needs specialized hardware to work.
2. Cannot be implemented using the same specialized hardware in other devices.

**1.3.3 STUDY ON PROPOSED SYSTEM**

* The proposed system will work using technologies like Machine Learning and Python.
* OpenCV will be used Capture the image through Webcam.
* Mediapipe Hand Tracking ML (Machine Learning) framework is used to detect Hands and its motion Then the detected Hands will be Segmented into landmarks [Figure 1].
* Using landmarks, the position of fingers is detected and can be used to recognize.
* Finally, using PyAutoGUI API will be used to control the VLC media player by integrating with different Gestures recognized in real-time.

**1.3.4 MERITS OF PROPOSED SYSTEM**

1. Accurate identification and recognition of hand.
2. No Specialized hardware required.
3. Works in most of all laptops.
4. Can be implemented in other ways easily.

**1.3.5 LITERATURE SURVEY**

In 2015, Chong Wang, “Super Pixel-Based Hand Gesture Recognition with Kinect Depth Camera” proposed the system which uses Kinect depth camera. It is based on a compact representation in the form of super pixels, which efficiently capture the shape, texture, and depth features of the gestures. Since this system uses Kinect depth camera, the cost of system is more.

In 2014, Swapnil D. Badgujar, "Hand Gesture Recognition System” proposed the system which recognize the unknown input gestures by using hand tracking and extraction method. This system is applied to recognize the single gesture. There is assumption of stationary background so that system will have smaller search region for tracking. This system only control mouse with the finger using it on web cam.

In 2014, Viraj Shinde, Tushar Bacchav, Jitendra Pawar and Mangesh Sanap developed “Hand Gesture Recognition System Using Camera”. They focus on using pointing behaviors for a natural interface to classify the dynamic hand gesture, they developed a simple and fast motion history image-based method. This paper presents low complexity algorithm and gestures recognition complexity and more suitable for controlling real time computer system. It is applicable only for the application of power point presentation.

**CHAPTER-II**

**SYSTEM ANALYSIS**

**2.1 USER REQUIREMENT SPECIFICATION**

In this project the user requirements are - Visual Studio Code (VSC), PYTHON 3, and packages like Mediapipe, CV2 must be installed in VSC.

**2.2 SOFTWARE REQUIREMENTS SPECIFICATION**

The software requirements specification can be divided into two, functional and non- functional requirements. Functional requirements describe the functions or operations that the system should perform.

**FUNCTIONAL REQUIREMENTS**

* User’s Hand must be identified regardless of Background.
* User can recognize the Hand Gesture.
* User can control VLC player.
* All the processes must happen in real time.

**NON-FUNCTIONAL REQUIREMENTS**

* Users Face will be blocked for increased performance.
* Works on any webcam video quality.
* Interval of 0.5 seconds is set to execute each command.
* Uses CPU not GPU.

**2.3 SYSTEM SPECIFICATIONS**

**HARDWARE SPECIFICATIONS -**

1. PROCESSOR - 1Ghz or more
2. RAM - 2GB or greater
3. ROM - 100GB (HDD Or SSD) or greater
4. WEBCAM - 720p or more

**SOFTWARE SPECIFICATION -**

1. Visual Studio Code
2. Python 3
3. Windows 10 OS
4. Multimedia Application (VLC Player)
5. Open CV2 Framework
6. Mediapipe Framework

**2. 4 FINAL OUTLINES OF THE PROPOSED SYSTEM**

* Gestures will be recognized in real time.
* A new form of HCI will be developed.
* Applications can be controlled seamlessly using our hand gestures.
* Can be used in all windows 10 laptops if the system requirements are satisfied.

**CHAPTER – III**

**DESIGN AND DEVELOPMENT PROCESS**

**3.1 FUNDAMENTAL DESIGN CONCEPTS**

**3.1.1 MODULES AND FUNCTIONALITIES**

**MODULES:**

* Image Capturing Module.
* Hand Identification Module.
* Gesture Recognition Module.
* Gesture Controlling Module.
* Integration Of VLC Player.

**FUNCTIONALITIES:**

* Swipe Right with One Finger — > Forward
* Swipe Left with One Finger — > Backward
* Swipe Up with two Fingers — > Volume Up
* Swipe Down with Two Fingers — > Volume Down
* Open Fist to Closed Fist — > Pause
* Closed Fist to Open Fist — > Play

**3.1.2 DESCRIPTION OF MODULES**

* **Image Capturing Module:**

Captures Image from the webcam feed using CV2 API.

* **Hand Identification Module:**

Using the video from the webcam feed the Mediapipe Framework will identify the hand and it will display the output in the cv2 window.

* **Gesture Recognition Module**:

Using the data provided by the hand Identification module the hand gestures will be recognized by using Mediapipe framework and some images.

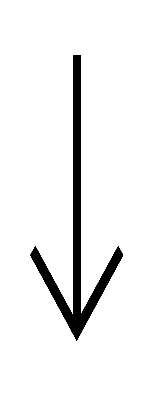
* **Gesture Controlling Module:**

Using gesture Recognition module various commands will be assigned to various gestures to provide a certain output like controlling an application function using PyautoGUI API.

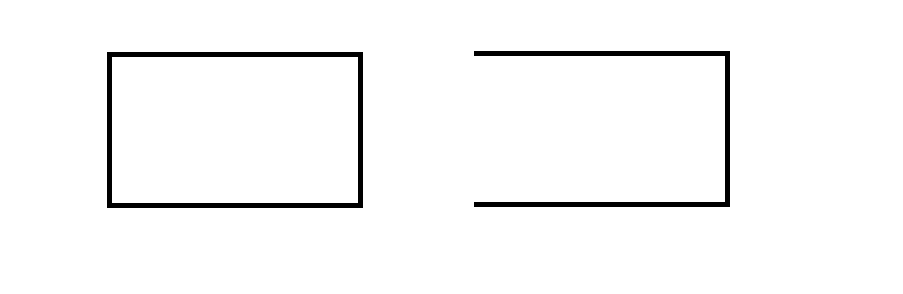
* **Integration of VLC media Player:** The Media player will be controlled by using the gesture controlling module with the help of PyautoGUI API.

**3.2 DESIGN NOTATION**

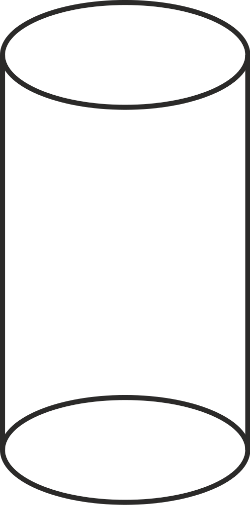
**3.2.1 DFD SYMBOLS**



An arrow identifiers data flow. It is the pipeline through which information flows.

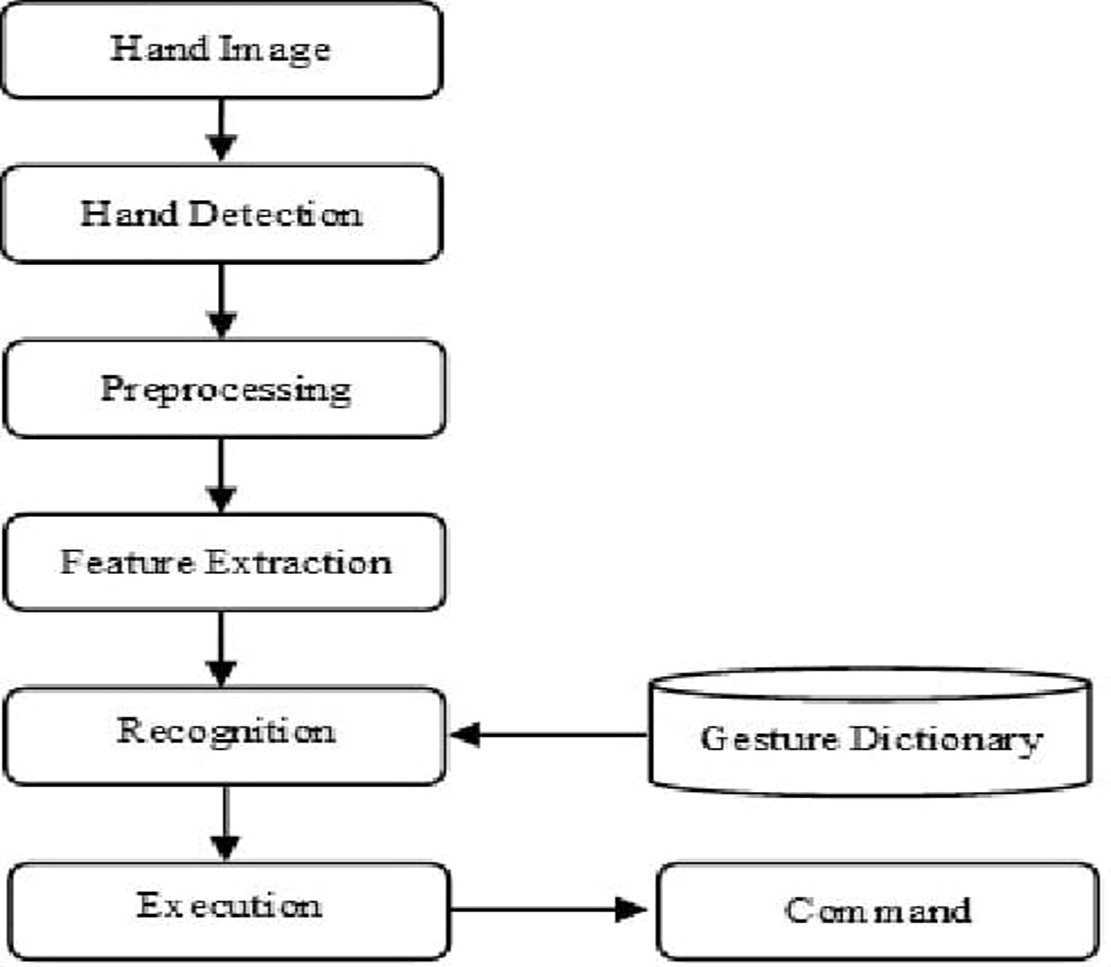


Open Rectangle represents the FUNCTION.



A CYLINDER DEFINES AS A DATABASE.

**3.2.2 DATA FLOW DIAGRAM**

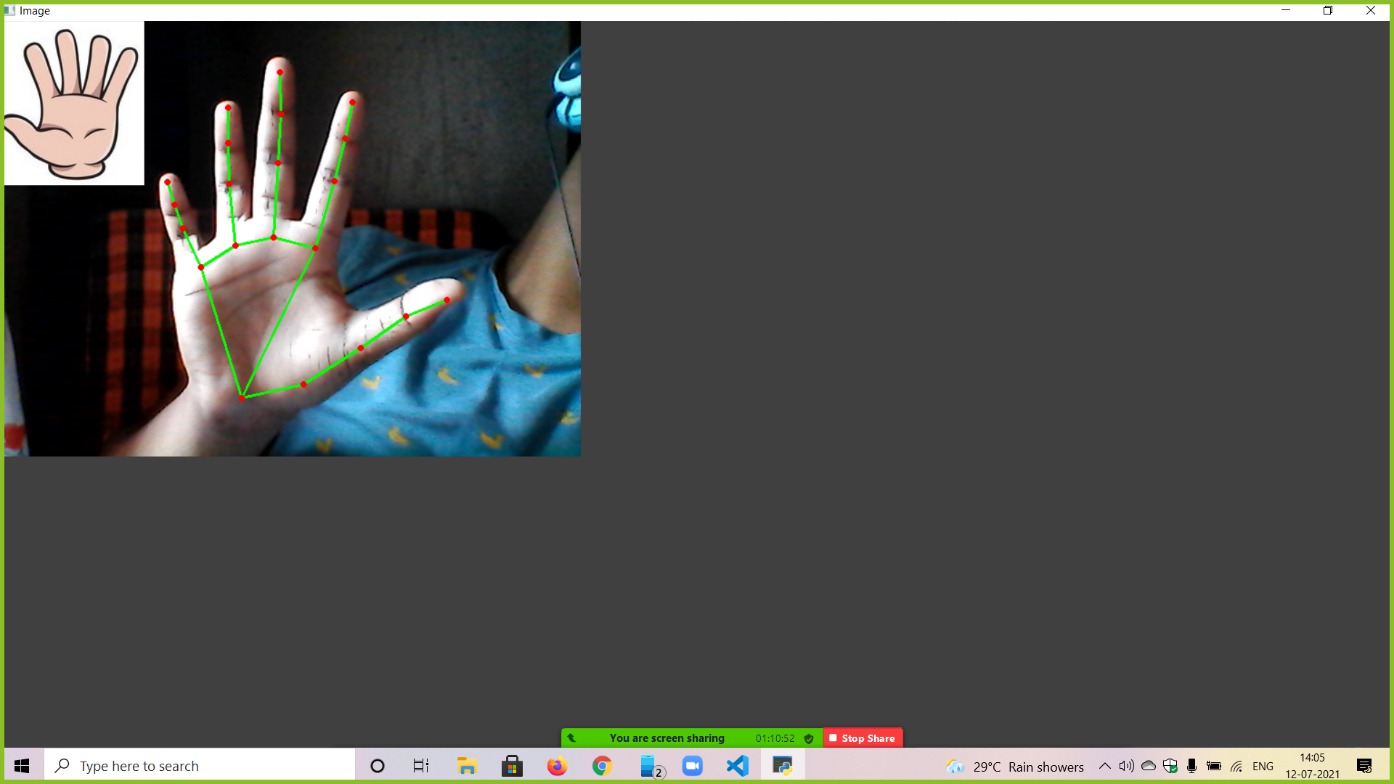


**3.3 DESIGN PROCESS**

**3.3.1 INPUT DESIGN**

**3.3.2TABLE DESIGN**

**3.3.3 OUTPUT DESIGN**



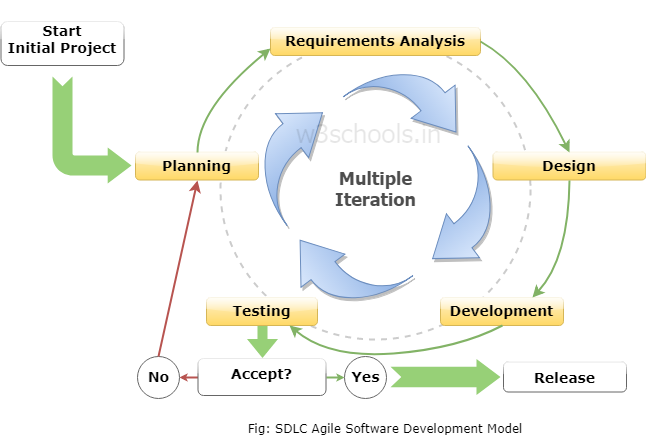
**3.4 DEVELOPMENT APPROACH**

**3.4.1 DEVELOPMENT MODEL APPROACH**

This project is made by the basis of agile model. Agile is a time-bound, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver all at once.

Iterations in Agile,

* Planning
* Requirements Analysis
* Design
* Development
* Unit Testing
* Deployment

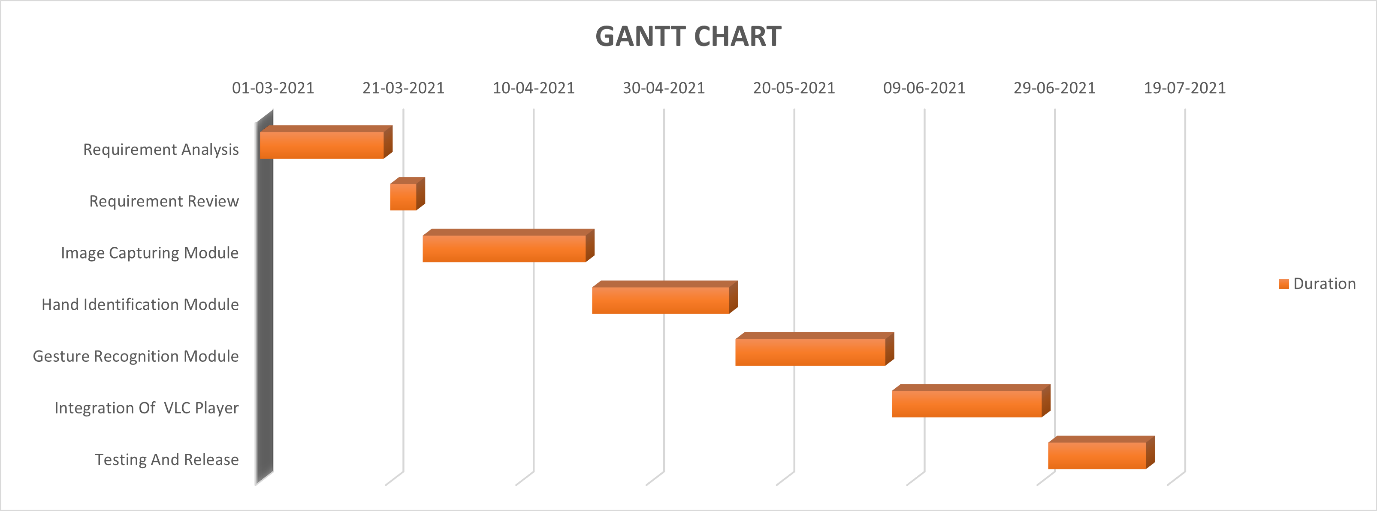


At the end of the iteration, the working product is released for the public.

**CHAPTER – IV**

**PLANNING**

**4.1 GANTT CHART**



A Gantt chart is a useful graphical tool which shows activities or tasks performed against time. It is also known as visual presentation of a project where the activities are broken down and displayed on a chart which makes it is easy to understand and interpret. A Gantt chart is a popular tool in project management. It basically drills down activities which need to be done by a fixed time period.

It is commonly used for tracking project schedules. On the chart, tasks are shown on the vertical axis while the scheduled time-spend is laid out on the horizontal axis. Each task is represented by a bar that shows the time required for the project. The bar then represents or shows percentage of tasks that have been completed. It also shows dependencies, which simply means the inter linkages between various activities in the project.

Gantt chart is a useful tool in planning and scheduling the projects. It keeps the management updated as to when the project will get completed. It also keeps the management informed about any additional resources that are required and manage dependencies between tasks. They are commonly used in scheduling production processes, employee roster or scheduling, events scheduling, production processes, etc. Microsoft Excel can also be used to create Gantt charts apart from other independent software available in the market.

**CHAPTER – V**

**TESTING AND IMPLEMENTATION**

**5.1 TESTING**

Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects) and verifying that the software product is fit for use.

After the implementation phase the testing part begins. Testing of the application determine the correctness, completeness, and quality of the application.

As the number of possible tests for even simple software components is infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects). The job of testing is an iterative process as when one bug is fixed; it can illuminate other, deeper bugs, or can even create new ones. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time,
* it is sufficiently usable,
* can be installed and run in its intended environments,
* Achieves the general result its stakeholder’s desire.

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions, but only that it does not function properly under specific conditions.The scope of software testing often includes the examination of code as well as the execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do.

In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

Being a developer, the main objective of us is to detect the bug as well as the other errors in an application. Error prevents to produce the correct output/desired output.

There are six kinds of testing is done in our project.

* Unit testing
* Performance testing
* Beta Testing
* Load Testing
* Acceptance Testing

**UNIT TESTING**

Unit testing is performed to test individual units of an application. The application compromises the various unit and modules, detecting errors in the unit is simple and consumes less time, as they are small inside. The unit testing is not just performed once during the application development but is repeated whenever the application is modified or used in a new environment.

* It ensures that all statements in the unit have been executed at least once.
* It tests data structure that represents relationship among individual elements.
* It ensures that the data entered in variable is of the same data type as defined in the unit.

**LOAD TESTING**

Load testing is the process of putting demand on a system or device and measuring its response.

The term load testing is used in diverse ways in the professional application testing community. Load testing refers to the practice application expected usage of application program by simulating multiple users accessing the program concurrently.

**BETA TESTING**

Application testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Application testing can also provide an object, independent view of the application. Application testing can be started as the process of validating and verifying that a computer program.

* Meets the requirement that guided its design and development.
* Works expected.
* Can be implemented with same characteristics.
* Satisfies the need of stakeholders.

**ACCEPTANCE TESTING**

In engineering and its various sub disciplines, acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests. In the case of application, acceptance testing performed by the customer is known as user acceptance testing, end-user testing, site testing, o field testing.

**5.1.1 TESTING METHODOLOGIES**

**5.1.1.1 BLACK BOX TESTING**

Black-box testing (also known as functional testing) treats the software as a "black box," examining functionality without any knowledge of internal implementation, without seeing the source code. Boundary value analysis and error guessing technique would be used to generate the test case for the functionality of this application.

**5.1.1.2 GUI TESTING**

It is used for testing all the visible components that can be seen on the device like Buttons, Navigation Drawer, Menus, and Toast etc.

**5.1.1.3 FUNCTIONALITY TESTING**

The entire functionality of the system is done in this testing. The goal of these test are to verify the acceptance, processing, retrieving and appropriate implementation of the project.

**5.1.1.4 INTEGRATION TESTING**

This testing is done after testing the functionalities of each modules. The module has been integrated and tested as a system. In this the interface issues can easily fixed and located.

**5.1.2 TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SERIAL NO** | **TEST CASE ID** | **DESCRIPTION** | **EXPECTED RESULT** | **ACTUAL RESULT** | **RESULT** |
| 1 | P1 | Run the Program | Camera window should be opened | Camera window opened | Pass |
| 2 | R2 | Run the program and check whether the hands are identified in the CV2 window | Hands must be identified | Hands are identified and drawn | Pass |
| 3 | R1 | Perform some hand gestures Infront of the camera | Gestures must be recognized | Gestures are recognized | Pass |
| 4 | C1 | Play a Video on VLC Player and perform the gesture ‘fist closed’ | Video must pause | Video got paused | Pass |
| 5 | C2 | Pause a Video on VLC Player and perform the gesture ‘Show your Palm’ | Video must play | Video got played | Pass |
| 6 | C3 | Play a Video on VLC Player and perform the gesture ‘One finger’ | Volume should be increased | Volume got increased | Pass |
| 7 | C4 | Play a Video on VLC Player and perform the gesture ‘two fingers’. | Volume should be decreased | Volume got decreased | Pass |
| 8 | C5 | Play a Video on VLC Player and show three opened fingers to camera. | Video should seek forward | Video got forward seek | Pass |
| 9 | C6 | Play a Video on VLC Player and show three opened fingers to camera. | Video should jump backward | Video got backward seek | Pass |
| 10 | C7 | Play a Video on VLC Player and do not perform a hand gesture Infront of camera | Nothing should happen | Nothing happened | Pass |
| 11 | P2 | Stop the program | The camera must turn off | The camera got turned off | Pass |

**5.2 SYSTEM IMPLEMENTATION**

**5.2.1 IMPLEMENTATION PROCEDURES**

During the implementation process we are launching this project in GitHub. So, the public users can download the code and run it in their system. After the implementation the public users must install the minimum requirements as specified so the program works error-free and the users can be able to control the VLC media player seamlessly with Hand Gestures.

**CHAPTER –VI**

**CONCLUSION**

**6.1 CONCLUSION**

In current world many facilities are available for providing input to any application some needs physical touch and some without using physical touch (speech, hand gesture etc.). But few applications are available which are controlled using current and smart facility of providing input which is by hand gesture. By this method user can handle application from distance without using keyboard and mouse. This application provides a novel human computer interface by which a user can control media player (VLC) using hand gesture. The application defines some gesture for controlling the functions of VLC player. The user will provide gesture as an input according to interested function. The application provides a flexibility of defining user interest gestures for specific command which make the application more useful for physically challenged people, as they can define the gesture according to their feasibility.

By this method user can handle application from distance without using keyboard and mouse. This application provides a novel human computer interface by which a user can control media player (VLC) using hand gesture. The application defines some gesture for controlling the functions of VLC player.

This Technology can be used as Communication Medium between Humans and Computers. It can be used in many areas such as app controlling and sign language detection. Using Python, Mediapipe & Open CV a Hand Gesture Recognition model can be made to control a given app.

**6.2 SCOPE FOR FUTURE ENHANCEMENT**

**Future scope**

This enhanced media player can help in minimizing human efforts. In future, this technique can be used to control systems using HCI like pdf reader, power point etc.

As a future prospect of this research, we are also going to investigate with the enormous number of gestures with different persons and motion type hand gestures are developed. We are also going to generalize our system so that it can be useful for other different media players available in market.

For controlling VLC, presently the application uses global keyboard shortcut in VLC and making keyboard event of that global shortcut with keyboard event () function. It is not the smart way of controlling any application. Inter-process communication technique can be applied for this. By applying inter-process communication then VLC can be replaced with other application very easily.

**ANNEXURE**

**SAMPLE CODE**

**GestureRecogition**

import cv2

import time

import os

import HandTrackingModule as htm

wCam, hCam = 640, 480

cap = cv2.VideoCapture(0)

cap.set(3, wCam)

cap.set(4, hCam)

folderPath = "FingerImages"

myList = os.listdir(folderPath)

print(myList)

overlayList = []

for imPath in myList:

image = cv2.imread(f'{folderPath}/{imPath}')

# print(f'{folderPath}/{imPath}')

overlayList.append(image)

print(len(overlayList))

pTime = 0

detector = htm.handDetector(detectionCon=0.75)

tipIds = [4, 8, 12, 16, 20]

while True:

success, img = cap.read()

img = detector.findHands(img)

lmList = detector.findPosition(img, draw=False)

# print(lmList)

if len(lmList) != 0:

fingers = []

# Thumb

if lmList[tipIds[0]][1] > lmList[tipIds[0] - 1][1]:

fingers.append(1)

else:

fingers.append(0)

# 4 Fingers

for id in range(1, 5):

if lmList[tipIds[id]][2] < lmList[tipIds[id] - 2][2]:

fingers.append(1)

else:

fingers.append(0)

# print(fingers)

totalFingers = fingers.count(1)

print(totalFingers)

h, w, c = overlayList[totalFingers - 1].shape

img[0:h, 0:w] = overlayList[totalFingers - 1]

cv2.imshow("Image", img)

cv2.waitKey(1)

**Hand Tracking**

import mediapipe as mp

import time

import cv2

cap=cv2.VideoCapture(0) #Zero Means Webcam

mpHands = mp.solutions.hands

hands= mpHands.Hands()

mpDraw= mp.solutions.drawing\_utils

while True:

success, img = cap.read()

imgRGB= cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

results = hands.process(imgRGB)

#print(results.multi\_hand\_landmarks)

if results.multi\_hand\_landmarks:

for handLms in results.multi\_hand\_landmarks:

for id, lm in enumerate(handLms.landmark):

#print(id, lm)

h, w, c = img.shape

cx, cy = int(lm.x\*w), int(lm.y\*h)

print(id, cx, cy)

mpDraw.draw\_landmarks(img, handLms,mpHands.HAND\_CONNECTIONS)

cv2.imshow("Image" ,img)

cv2.waitKey(1)

**Hand Tracking module**

import mediapipe as mp

import time

import cv2

class handDetector():

def \_\_init\_\_(self,mode=False,maxHands= 2, detectionCon=0.5, trackCon=0.5):

self.mode = mode

self.maxHands = maxHands

self.trakConf = trackCon

self.detectionCon = detectionCon

self.mpHands = mp.solutions.hands

self.hands= self.mpHands.Hands(self.mode,self.maxHands,self.trakConf,self.detectionCon)

self.mpDraw= mp.solutions.drawing\_utils

def findHands(self,img, draw=True):

imgRGB= cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

self.results = self.hands.process(imgRGB)

#print(results.multi\_hand\_landmarks)

if self.results.multi\_hand\_landmarks:

for handLms in self.results.multi\_hand\_landmarks:

if draw:

self.mpDraw.draw\_landmarks(img, handLms, self.mpHands.HAND\_CONNECTIONS)

return img

def findPosition(self, img, handNo=0, draw = True ):

lmList = []

if self.results.multi\_hand\_landmarks:

myHand = self.results.multi\_hand\_landmarks[handNo]

for id, lm in enumerate(myHand.landmark):

#print(id, lm)

h, w, c = img.shape

cx, cy = int(lm.x\*w), int(lm.y\*h)

#print(id, cx, cy)

lmList.append([id, cx, cy])

if draw:

cv2.circle(img,(cx,cy), 15, (255, 0, 255),cv2.FILLED)

return lmList

def main():

pTime = 0

cTime = 0

cap = cv2.VideoCapture(0) #Zero Means Webcam

detector = handDetector()

while True:

success, img = cap.read()

img = detector.findHands(img)

lmList = detector.findPosition(img)

if len(lmList) !=0:

print(lmList[4])

cTime = time.time()

fps = 1 / (cTime - pTime)

pTime = cTime

cv2.putText(img, str(int(fps)), (10,70), cv2.FONT\_HERSHEY\_PLAIN, 3, (255,0,255),3)

cv2.imshow("Image" ,img)

cv2.waitKey(1)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**BIBLIOGRAPHY**

**BOOK REFERENCE**

* Introduction to Machine Learning with Python (Andreas C. Müller and Sarah Guido)

**REFERENCE**

* Chong Wang, Zhong Liu and Shing-Chow Chan “Super Pixel-Based Hand Gesture Recognition with Kinect Depth Camera,” IEEE Trans. Multimedia, vol. 17, no. 1, Jan. 2015
* Swapnil D. Badgujar, Gourab Talukdar, Omkar Gondhalekar and Mrs. S.Y. Kulkarni “Hang Gesture Recognition System,” International Journal of Scientific and Research Publications, vol. 4, Issue 2, Feb. 2014, pp. 2250- 3153.
* Viraj Shinde, Tushar Bacchav, Jitendra Pawar and Mangesh Sanap “Hand Gesture Recognition System Using Camera,” International Journal of Engineering Research & Technology (IJERT), Vol. 3, Issue 1, January – 2014

**WEB REFERENCE**

* [MediaPipe](https://google.github.io/mediapipe/solutions/hands)
* [OpenCV-Python Tutorials](https://www.docs.opencv.org/master/d6/d00/tutorial_py_root.html)
* [PyAutoGUI Documentation](https://pyautogui.readthedocs.io/en/latest/)