A

Project Report

on

Gesture-Based Input System

Submitted for partial fulfillment of the requirements for the award of the degree

of **BACHELOR OF ENGINEERING**

in

COMPUTER SCIENCE AND ENGINEERING

By

REEMA MAHABOOBA –2451-18-733-107
QUHURA FATHIMA–2451-18-733-100
AAKANKSHA NEELAKANTLA –2451-18-733-118

Under the guidance of

Dr. K. Venu Gopal Rao
Professor , Dean of Academics
Department of CSE

MATURI VENKATA SUBBA RAO (MVSR) ENGINEERING COLLEGE **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** (Affiliated to Osmania University & Recognized by AICTE)
Nadergul(V), Balapur(M), Ranga Reddy(D), Hyderabad – 501 510, Telangana, INDIA

(Academic Year: 2021 - 2022)

CERTIFICATE

This is to certify that this Project Report entitled "Gesture-Based Input System" is a bonafide work carried out by ***Ms.Reema Mahabooba (2451-18-733-107), Ms.Quhura Fathima (2451-18-733-100), Ms.Neelakantla Aakanksha (2451-18-733-118)*** *in partial fulfillment of the requirements for the award of the degree of* ***Bachelor of Engineering*** *in* ***Computer Science And Engineering*** *from* ***Maturi Venkata Subba Rao (MVSR) Engineering College,*** *affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2021-22 under our guidance and supervision.*

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma to the best of our knowledge and belief.

Internal Guide

K. Venu Gopal Rao
Professor, Dean of Academics.

Head of the Department

J.Prasanna Kumar
Professor and head of CSE

DECLARATION

This is to certify that the work reported in the present project entitled "GESTURE-BASED INPUT SYSTEM" is a record of bonafide work done by us/me in the Department of Computer Science and Engineering, Maturi Venkata Subba Rao (MVSR) Engineering College, Osmania University. The reports are based on the project work done entirely by us and not copied from other sources. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our / my knowledge and belief.

Reema Mahabooba[2451-18-733-107]

Aakanksha[2451-18-733-118]

Quhura Fathima[2451-18-733-100]

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and indebtedness to our project coordinator/guide Dr. K. Venu Gopal Rao, Professor and Dean of academics, CSE Department, MVSREC for his valuable suggestions and interest throughout the course of this project.

We are also thankful to our principal Dr.G.KanakaDurga Madam and Prof.J.Prasanna Kumar Sir, Head, Department of Computer Science and Engineering, Maturi Venkata Subba Rao (MVSR) Engineering College, Hyderabad for providing supporting infrastructure, environment, and ambiance for completing this project successfully as a part of our B.E.(CSE) Degree.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completing this work.

Reema Mahabooba (2451-18-733-107)
Quhura Fathima (2451-18-733-100)
Neelakantla Aakanksha (2451-18-733-118)

VISION

- To impart technical education of the highest standards, producing competent and confident engineers with an ability to use computer science knowledge to solve societal problems.

MISSION

- To make the learning process exciting, stimulating, and exciting.
- To impart adequate fundamental knowledge and soft skills to students. ● To expose students to advanced computer technologies in order to excel in engineering practices by bringing out creativity in students.
- To develop economically feasible and socially acceptable software.

PEOs:

PEO-1: Achieve recognition through demonstration of technical competence for successful execution of software projects to meet customer business objectives..

PEO-2: Practice life-long learning by pursuing professional certifications, higher education, or research in the emerging areas of information processing and intelligent systems at a global level.

PEO-3: Contribute to society by understanding the impact of computing using a multidisciplinary and ethical approach.

PROGRAM OUTCOMES (POs)

At the end of the program the students (Engineering Graduates) will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.
2. **Problem analysis:** Identity, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge

and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions**.**
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Lifelong learning:** Recognise the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

13. (PSO-1) Demonstrate competence to build effective solutions for computational real-world problems using software and hardware across multi-disciplinary domains. 14. (PSO-2) Adapt to current computing trends for meeting the industrial and societal needs through holistic professional development leading to pioneering careers or entrepreneurship.

COURSE OBJECTIVES AND OUTCOMES

Course Objectives:

➢To enhance practical and professional skills.

➢ To familiarize tools and techniques of systematic literature survey and documentation

➢ To expose the students to industry practices and teamwork.

➢ To encourage students to work with innovative and entrepreneurial ideas

Course Outcomes:

CO1: Summarize the survey of the recent advancements to infer the problem statements with applications towards society

CO2: Design a software-based solution within the scope of the project.

CO3: Implement test and deploy using contemporary technologies and tools

CO4: Demonstrate qualities necessary for working in a team.

CO5: Generate a suitable technical document for the project..

ABSTRACT

Gesture-based input system is a new way to interact with a computer system unlike traditional input devices (Keyboard and mouse). Webcam is used to recognize users' hands based on gestures necessary system operations are performed like play/pause music, play games, virtual keyboard, etc, Gesture-Based Input System recognizes hand gestures in real-time using the power of Neural Networks. Through this System a user can interact with the computer easily without any traditional devices like Keyboard, Mouse and etc. future development can be made to recognize face gestures so even disabled people can easily interact with computers. In our system we propose some non-complex algorithms and hand gestures to decrease the hand gesture recognition complexity and would be more easy and simple to control real-time computer systems

Reema Mahabooba (2451-18-733-107)
Neelakantla Aakanksha (2451-18-733-118)
Quhura Fathima (2451-18-733-100)

Index

LIST OF TABLES

LIST OF FIGURES

Figure No Figure Name Page No

CHAPTER - 1 INTRODUCTION

COMPUTERS and computerized devices have become an eminent element of our society. They increasingly influence many aspects of our lives; for example, the way we communicate, the way we perform our actions, and the way we interact with our environment. Thus a new concept of interaction emerged Human-Computer Interaction (HCI). Although computers have made tremendous advancements, the common HCI still relies on input devices such as keyboard, mouse, and joysticks. By the underlying prototype, users express their significance to the computer, a user using their hands to perform button clicks, positioning the mouse, and key presses. This is rather an unnaturally restrictive way of interacting with end-user systems. With the increase in the interaction of computers in our daily life, it would be worthy enough to get a Perceptual User Interface (PUI) to interact with computers as humans interact with each other. Vision-based gesture recognition is an important technology for the friendly human-computer interface and has received more and more attention in recent years. The applications designed for gesture recognition generally require restricted background, a set of gesture commands, and a camera for capturing images. The gesture used in an application for performing an action must represent the action which is being performed by it and also it must be logically explainable, thus for controlling a media player like VLC dynamic hand gestures could be more intuitive and natural.

1.1 Problem Statement

Gesture-based input system is a new way to interact with computer systems unlike traditional input devices (Keyboard and mouse). As the various hand gestures are frequently used by humans so the aim of this project is to reduce external hardware interaction which is required for computer applications, and hence this causes the system more reliable for use with ease. The required model is trained and verified.

[1]

1.2 Objectives

- To minimize the use of the keyboard and mouse on the computer.
- To integrate gesture recognition features into any computer at a low cost.

[1]

- To help in the development of a non-tangible way to interact with the video player.

1.3 Visual and Thermal Images

- **HSV Scale Image** First the image is captured by the webcam, then various image processing is done on it. The original image is then converted into HSV ie Hue Saturation and Value. This is done to detect the portion of the hand and separate it from the background. HSV defines a type of color space. Value is defined as brightness. In HSV, the hue represents a color. In this system, we have considered Hue in a range from 0 to 20. Saturation indicates the range of grey in the color space. We have considered saturation range from 55 to 255. Value is the brightness of the color and varies with color saturation. In this system, we have considered a range from 0 to 255 for Value, when the value is 0 the color space will be totally black. With the increase in the value, the color space brightens and shows various colors.
- **Threshold Image** In this module we have obtained a threshold image using HSV ranges of color detection. Here we find the biggest contour for hand detection. FindBiggestContour() with the use of OpenCV function i.e. cvFindContours() to make a list of various contours. In order to binary threshold image, a contour is a region of white pixels. Each region is approximated by a bounding box, and the contour corresponding to the largest box is taken and transmitted.
- **Filtered Image** In this module we extract the noisy pixels detected other than the hand border. By calling extractContourInfo () we extract noisy pixels and analyze the contour.
- **Calculate Center Of Gravity** In section, the box surrounding the contour is used to receive a center. This is enough as the underlying shape is a rectangular card, due to which the contour and box are almost the same. Therefore, a rounding box around a hand can simply have other COG or angle from the hand itself; in such case, it is very important to analyze the hand contour other than a rounding box, by using moments. In this system, we used spatial moments to obtain the center of gravity of an input binary image. This same method can be used to a contour to obtain its center or centroid. In this, we can calculate second-ordered mixed moments, which will give info about the spread of pixels around the centroid. Second-order moments can be brought together to return the angle of the contour's major axis with respect to the x-axis. In the OpenCV moments notation, the m () moments function takes the following two arguments, a and b, which are been used as powers for x and y. The I() function is the intensity for the pixel defined by its (x, y) coordinate. s is the no of pixels that make up the shape. Then we take the contour, then is the angle of its major axis to

the horizontal, with the +yaxis pointing downwards. The m() function, it can be presented as that below below-Identifying the fingertips is been identified in the 1st row in the code, a convex hull is been wrapped around the contour by the OpenCVs cvConvexHull2() and this polygon is matched to the contour by the cvConvexityDefects() to find its flaws. The fingertips are been saved in a tipPts [] array, and the finger fold in foldPts[], and depths in the depths[]. The analysis mostly generates too many problems, thus reduceTips () is been called at the end of findFingerTips(). It then carries out two simple tests to filter out problems that are not fingertips. It removes points having shallow defect depths and coordinates having too big an angle between the neighboring fold points reduceTips() and saves the other remaining tip points in the global fingerTips list. T

1.4 Motivation

There are a lot of critical situations in the day-to-day lives of disabled people. We have come up with a small part of such a life to make it easier using computer vision technology. In this paper,we have discussed a low system that uses hand gesture recognition technology to control the VLC media player. Among many computer visions based interactive systems, designing hand gestures and facial expression-based HCI system retains to be a highly challenging task. Our main purpose is to find a nontangible way to interact with the computer. Our project aims at modifying the existing video player controlled by human hand gestures by making use of Convex Hull and OpenCV.

1.5 System Architecture

The Model-View-Controller (MVC) framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller. Hence the abbreviation MVC. Each architecture component is built to handle specific development aspects of an application. MVC separates the business logic

and presentation layer from each other. It was traditionally used for desktop graphical user interfaces (GUIs). Nowadays, MVC architecture has become popular for designing web applications as well as mobile apps.

The following figure [1.2] depicts the flow of the process that we do in building the model

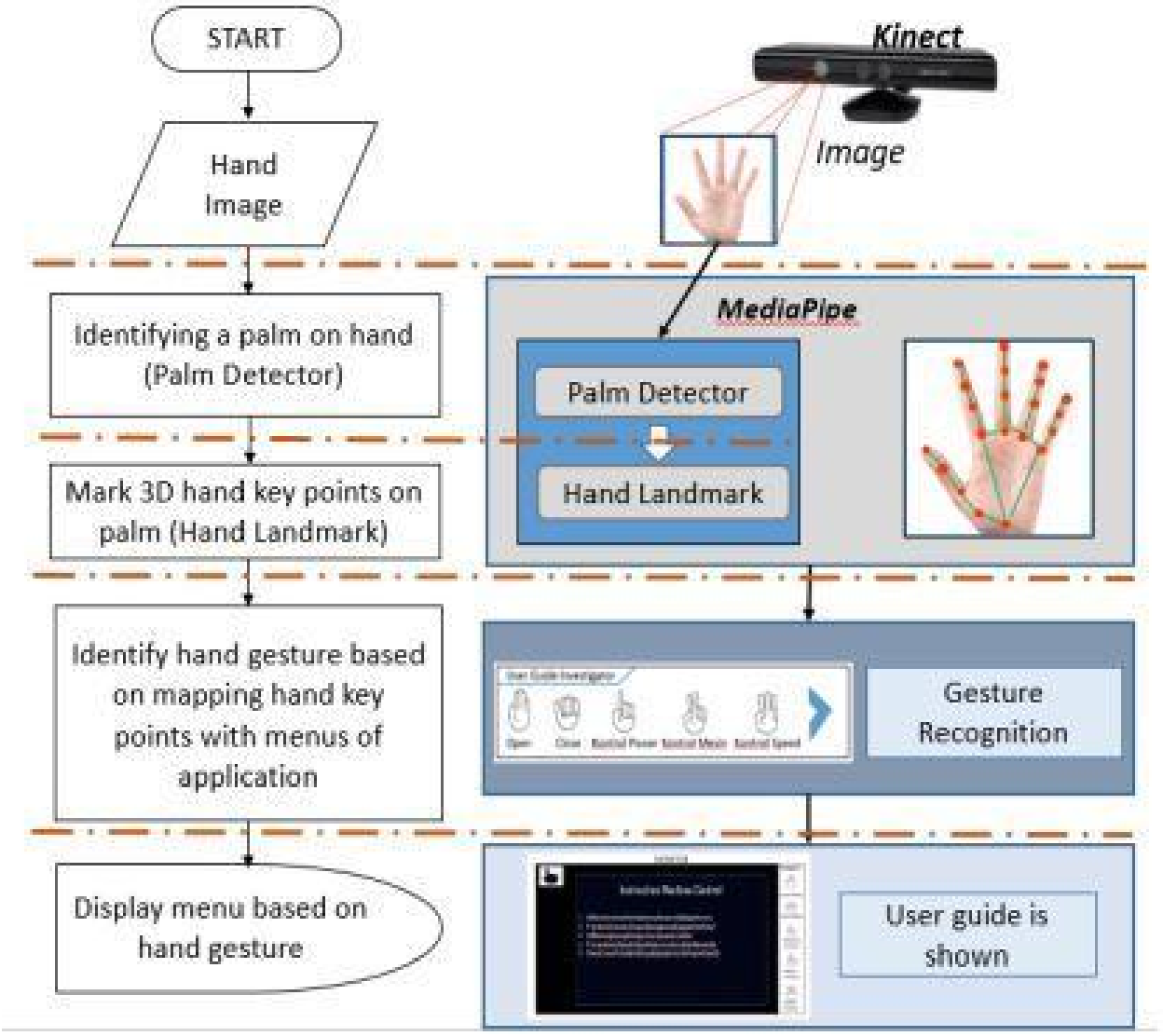


Figure 1.1- System Architecture

Steps involved in the system

➢ **Encoding the images**

Firstly, take a single image of both the types i.e visual and thermal. Then these images are encoded respectively using encoders.

➢ **Fusion of encoded images**

Then, the encoded images are fused together into a single encoded image. The feature maps carry out the process of fusion.

➢ **Decoding the fused image**

Once the features are fused, the fused feature vector is transferred to the decoder block which decodes it back to image content. From this module, a final

fused image is obtained which is then transferred to the next module called the ResNet block.

➢ **Mobilenet-SSD**

The fused image is then passed to the Single Shot Detector (SSD) object detection model which uses Mobilenet as the backbone. The output of this will be the image of

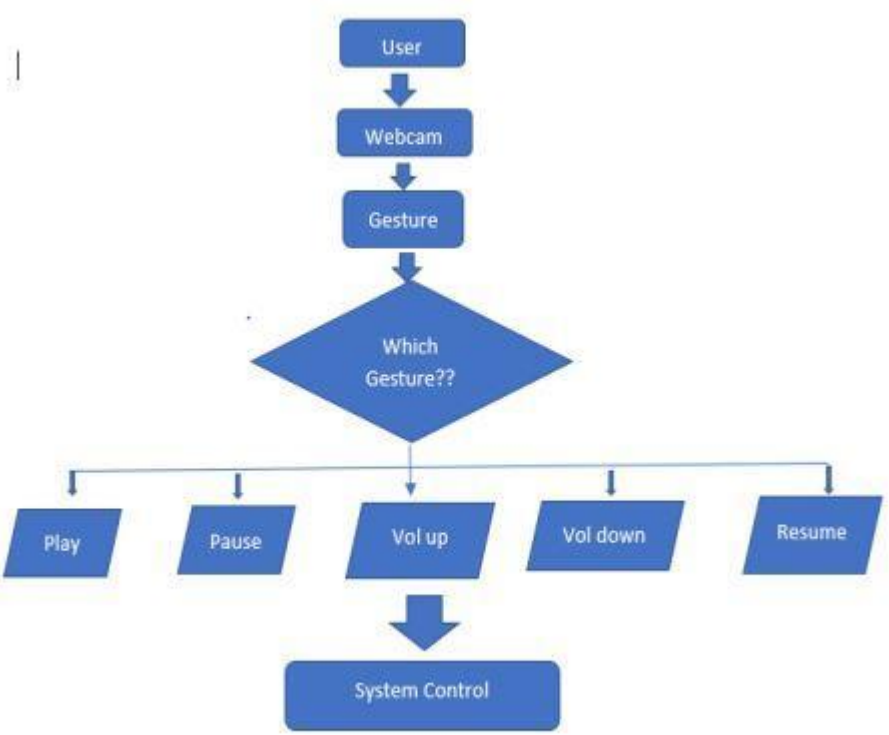


Figure 1.2- Data Flow

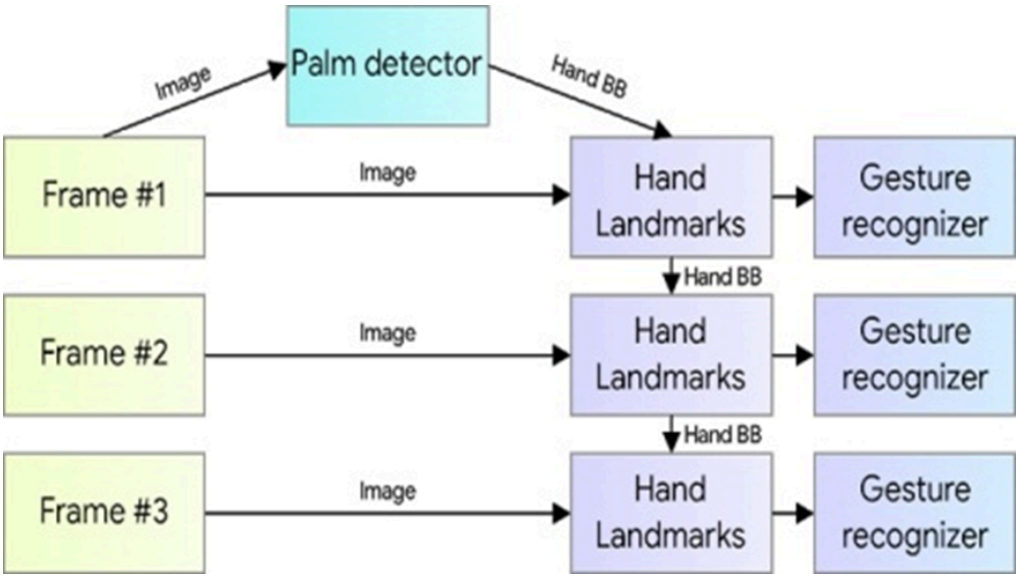


Figure 1.3- Technical Architecture

1.6 Scope

Through this System a user can interact with the computer easily without any traditional devices like Keyboard, Mouse, etc. future development can be made to recognize face gestures so even disabled people can easily interact with computers.

1.7 Collection of Datasets

The dataset has been collected from the Kaggle dataset.

Sample images:

RGB Thermal

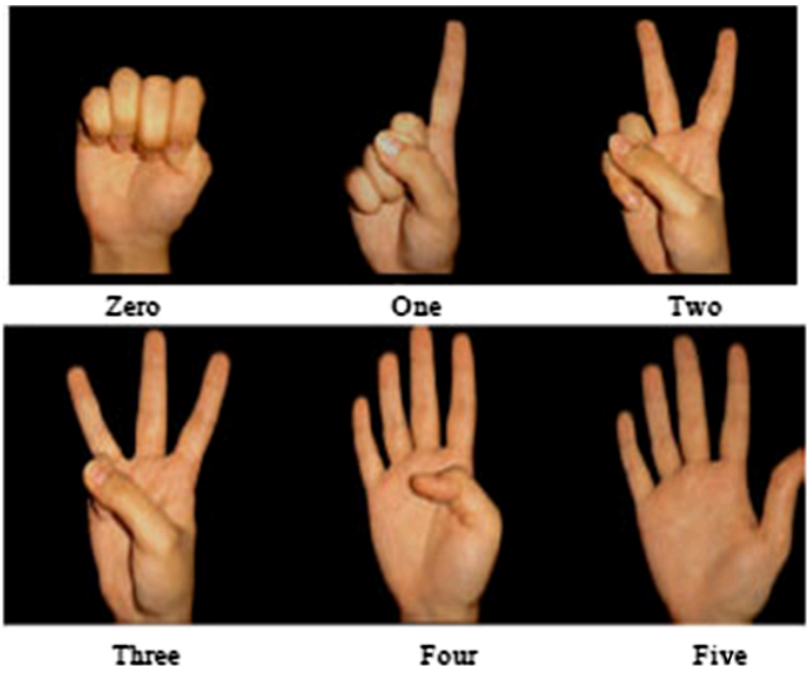


Figure 1.4- Sample data

1.8 Challenges

1.8.1 Computational cost is very high

The computational cost is high because in the model we are using InfraRed Cameras

which are more expensive than the usually visible cameras.

1.8.2 Performance is limited to datasets considered

For this type of application, the performance measure is restricted to the number of datasets and the number of images taken into consideration. Here, we are considering only one dataset i.e, LLVIP [45]. If we increase the number of datasets there will be a change in the performance.

1.8.3 Integration of RGB -Thermal image pairs create additional overhead

The integration of necessary features from both the RGB and thermal images add additional overhead to the architecture.

1.9 Application Areas

VLC Media Player: The VLC media player is a free and open-source, portable, cross-platform media player software and streaming media server developed by the VideoLAN project. VLC is available for desktop operating systems and mobile platforms, such as Android, iOS, and iPadOS.

Control a computer system without direct physical contact: Gesture control is the ability to recognize and interpret movements of the human body to interact with and control a computer system without direct physical contact. The term “natural user interface” is becoming commonly used to describe these interface systems, reflecting the general lack of any intermediate devices between the user and the system.

Sign language translation: Sign language recognition using image-based hand gesture recognition techniques. Abstract: Hand gesture is one of the methods used in sign language for non-verbal communication. It is most commonly used by deaf & dumb people who have hearing or speech problems to communicate among themselves or with normal people.

CHAPTER - 2 LITERATURE SURVEY

2.1 Background

Artificial intelligence (AI) is the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. Artificial Intelligence is the study of "intelligent agents", any system that perceives its environment and takes actions that maximize its chance of achieving its goals.

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. Machine learning is a branch of artificial intelligence (AI) and computer science which uses data and algorithms to imitate the way that humans learn and work by gradually improving its accuracy.

The steps involved in Supervised Machine learning are

1. Data Collection
2. Data Preparation

3. Choose a Model
4. Train the Model
5. Evaluate the Model
6. Parameter Tuning
7. Make predictions

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

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

In 2014, Viraj Shinde, Tushar Bacchav, Jitendra Pawar, and Mangesh Sanap developed a “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 is more suitable for controlling real-time computer systems. It is applicable only for the application Of PowerPoint presentations.

In 2014, N. Krishna Chaitanya and R. Janardhan Rao presents “Controlling of windows media player application using hand gesture recognition”, this system uses various hand gestures as input to operate the windows media player application. This system uses single-hand gestures and its directional motion which defines a particular gesture for the above-mentioned application. In this system, a decision tree has been used for classification. This system only supports the windows media player applications and not any others.

In 2012, Ram Rajesh J., Sudharshan R., Nagarjuna D., and Aarthi R., “Remotely controlled PowerPoint presentation navigation using hand gestures” developed the system in which slides of power point presentations are controlled without using any marker and gloves. In this system, the developer used the

segmentation algorithm for hand detection. After detecting hand calculation is for action figures. If the fingers are not stretched properly while making a gesture then the application did not work properly.

In 2012, Ruize Xu, Shengli Zhou, and Wen J. Li, "MEMS Accelerometer Based Non-specific-User Hand Gesture Recognition", had built a system that can recognize various hand gestures such as up, and down, right, and left, cross, and round. Three various modules were developed that recognize various hand gestures. The Signals by MEMS (MicroElectromechanical System) 3- axes accelerometers were provided as input. The motion of the hand in three perpendicular directions is been detected by three accelerometers and sent to the system by Bluetooth. A segmentation algorithm was been applied and finally, the various hand gestures were recognized by matching gestures that were already saved in the system. People mostly prefer the internet to have daily updates on weather, news, etc. So for this purpose, they perform keyboard and mouse operations. This system gives less accuracy in finding the terminal points of gestures due to the small size of the database of hand gestures.

In 2010, Anupam Agrawal and Siddharth Swarup Rautaray, "A Vision-based Hand Gestures Interface for Operating VLC Media Player Application "system, in the K nearest neighbor algorithm has been used to recognize the various gestures. VLC media player features that were operated by hand gestures include play, pause, Fullscreen, stop, increase volume, and decrease volume. Lucas Kanade Pyramidical's Optical Flow algorithm has been used to recognize hands from the input video. This above-mentioned algorithm recognizes moving points in the input image. Then K-means was been used to find the center of the hand. Using this center, the hand is recognized. This system uses a database that consists of various hand gestures and then the input was compared with this stored image and accordingly VLC media player was controlled. The present application is less robust in the recognition phase.

In 2006, Erol Ozgur and Asanterabi Malima, build an "A Fast Algorithm for Vision-Based Hand Gestures Recognition for Robot Control" which controlled robots using hand gestures but considered limited gestures. Firstly segmentation of the hand region was carried followed by locating the fingers and then finally classifying the gestures. The algorithm used is invariant to the translation, rotation, and scale of the hand. This system is applicable to robot control applications with reliable performance.

In 2003, Ahmed Elgammal, Vinay Shet, Yaser Yacoob, and Larry S. Davis presents "Learning Dynamics for Exemplar-based Gesture Recognition". This system addresses the problem of recognizing the dynamics of an exemplar-based recognition system. It processes the nonparametric HMM (Hidden Markov Model) approach that uses HMM with arbitrary states to identify the gestures in a vast exemplar

distribution. This reduces the want for long training in HMM observation model. The approach is based on the recognition of each gesture as a series of learned body gestures (exemplar).

In 2002, Lars Bretzner, Ivan Laptev, and Tony Lindeberg, “Hand Gesture Recognition using Multi-Scale Color Features, Hierarchical Models and Particle Filtering”, published algorithms for hand tracking, hand posture recognition. In this system, on each input image multi-scale color feature detection is carried out. By using particle filtering, with a layered sampling referred to as hierarchically layered sampling hands are been detected and also tracked. Overall body poses of a person are captured for different gestures.

CHAPTER 3- SYSTEM DESIGN

3.1 System Requirements

Software & Hardware requirements

HARDWARE REQUIREMENTS

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum

SOFTWARE REQUIREMENTS

- Programming Language / Platform : Python
- IDE : pycharm/jupyter

3.2 SOFTWARE REQUIREMENT SPECIFICATION

What is SRS?

Software Requirements Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consists of two basic activities:

Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, Specification languages and tools, and checking the specifications are addressed during this activity. The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic of this phase.

Role of SRS:

The purpose of the SRS is to reduce the communication gap between the clients and the developers. SRS is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

Purpose:

The purpose of this document is to describe all external requirements for the E-learning System. It also describes the interfaces for the system.

- **Scope:**

This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also by the basis for validating the final deliver system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alternations without the permission of the client.

- **Overview:**

The SRS begins the translation process that converts the software Requirements into the language the developers will use. The SRS draws on the Use Cases from the user Requirement

Document and analyses the situations from a number of perspectives to discover and eliminate inconsistencies, ambiguities and omissions before development progresses significantly under mistaken assumptions.

3.3 Functional Requirements:

- Data acquisition: Data acquisition testing of our system was done by in the laptop.
- Segmentation: Image segmentation techniques. The complete process w the next section. o Skin Detection Model for d region. o Approximate Median subtraction of background. It has been observed that by the u methods for segmentation the blob wa better for further process.
- Feature extraction: In our work we h direction of the hand region as a feat
- Recognition Phase: In this work D used as a classification tool for gestures.
- VLC interaction: Give the appropriate command to the VLC player according to the recognized gesture.

3.4 NON Functional Requirements

The major non-functional Requirements of the system are as follows:

- **Usability:** The system is designed entirely with automated operations hence there is no user interference.
- **Reliability:** The system is more reliable because of the qualities that are inherited from the python platform. The code is built by using python which is more reliable.
- **Performance:** This system is developing in the high level languages and uses advanced front-end and back-end technologies .The response time on the client system is very less.
- **Supportability:** The system is designed is such a way that it is supportable on any platform and also it is supported on a wide range of hardware &software platform, which is having PVM, built into the system.

3.5 Environmental Setup

To build this project, we used Python and Jupyter Notebook.

3.5.1 Python:

The Python programming language is an Open Source, cross-platform, high level, dynamic, interpreted language.

The Python emphasizes readability, clarity and simplicity, whilst maximizing the power and 'philosophy' expressiveness available to the programmer. The ultimate compliment to a Python programmer is not that his code is clever, but that it is elegant. For these reasons Python is an excellent 'first language', while still being a powerful tool in the hands of the seasoned and cynical programmer.

Python is a very flexible language. It is widely used for many different purposes. Typical uses include:

- Web application programming with frameworks like Zope, Django and Turbogears
- System administration tasks via simple scripts
- Desktop applications using GUI toolkits like Tkinter or wxPython (and recently Windows Forms and IronPython)
- Creating windows applications, using the Pywin32 extension for full windows integration and possibly Py2exe to create standalone programs
- Scientific research using packages like Scipy and Matplotlib
- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.
- Python Syntax compared to other programming languages
- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

3.5.2 Jupyter notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Jupyter Notebooks are a powerful way to write and iterate on your Python code for data analysis. Rather than writing and re-writing an entire program, you can write lines of code and run them one at a time. Then, if you need to make a change, you can go back and make your edit and rerun the program again, all in the same window.

Jupyter Notebook is built off of IPython, an interactive way of running Python code in the terminal using the REPL model (Read-Eval-Print-Loop). The IPython Kernel runs the computations and communicates with the Jupyter Notebook front-end interface. It also allows Jupyter Notebook to support multiple languages. Jupyter Notebooks extend IPython through additional features, like storing your code and output and allowing you to keep markdown notes.

If you'd rather watch a video instead of read an article, please watch the following instructions on how to use a Jupyter Notebook. They cover the same information.

LAUNCH A NOTEBOOK

To launch a Jupyter notebook, open your terminal and navigate to the directory where you would like to save your notebook. Then type the command jupyter notebook and the program will instantiate a local server at localhost:8888 (or another specified port).

```
[9] -> jupyter notebook
[I 18:31:51.264 NotebookApp] Serving notebooks from, local directory: /Users/janedoe
[I 18:31:51.264 NotebookApp] 0 active kernels
[I 18:31:51.264 NotebookApp] The Jupyter Notebook is running at:
[I 18:31:51.264 NotebookApp] http://localhost:8888/?token=f9b639294933fa68c64c78effb9b65
19f9d8c45c903baa43
```

A browser window should immediately pop up with the Jupyter Notebook interface; otherwise, you can use the address it gives you. The notebooks have a unique token since the software uses pre-built Docker containers to put notebooks on their own unique path. To stop the server and shutdown the kernel from the terminal, hit control-C twice.

JUPYTER INTERFACE

Now you're in the Jupyter Notebook interface, and you can see all of the files in your current directory. All Jupyter Notebooks are identifiable by the notebook icon next to their name. If you already have a Jupyter Notebook in your current directory that you want to view, find it in your files list and click it to open.

To create a new notebook, go to New and select Notebook - Python 2. If you have other Jupyter Notebooks on your system that you want to use, you can click Upload and navigate to that particular file.

Notebooks currently running will have a green icon, while non-running ones will be grey. To find all currently running notebooks, click on the Running tab to see a list.

INSIDE THE NOTEBOOK

When you open a new Jupyter notebook, you'll notice that it contains a cell.

Cells are how notebooks are structured and are the areas where you write your code. To run a piece of code, click on the cell to select it, then press SHIFT+ENTER or press the play button in the toolbar above. Additionally, the Cell dropdown menu has several options to run cells, including running one cell at a time or to run all cells at once.

CHAPTER -4 IMPLEMENTATION

The images and their masked labels are in .png format. The images are being captured from the live video stream and these images are further processed to achieve efficiency of the detection stage. In the detection stage, initially, contours are identified for the images and then these contours are used for the creation of the Convex Hull around the hand region. Then these image outputs are passed as control commands to the VLC media player to perform the associated action

it is clear that, using three dimensions matrix pixel values (RGB values) for all the images in it crucial to distinguish between these images. Semantic Segmentation technique where the objective is to find different regions in an image and tag its corresponding labels, for this first step includes segmenting only the hand region from the live video stream and identify different possible gestures. The convex hull cluster is of peaks that cover the region of the hand. Here, we must clear the principle of the convex set, which means all lines between any 2 points within the hull are entirely within it. After determining the gesture, the specific functioning is performed. The method of recognizing the movement is a dynamic process. After operating the specific command from the gesture, go back to the initial step to accept other images to be processed, and so on.

After this the gesture pictures were segregated into labelled folders as play, pause, volume up, volume down and resume appropriately. Exploratory Image details Analysis is an approach for analysing details related to images to summarize their main characteristics, often with visual methods. Much complex visualization can be achieved with matplotlib and usually, there is no need to import other libraries. This is used for gaining a better understanding of data aspects like: -main features of data -variables and relationships that hold between them -identifying which variables are important for our problem Next we plan to input this processed images as input into the VLC media player and perform the necessary actions.

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

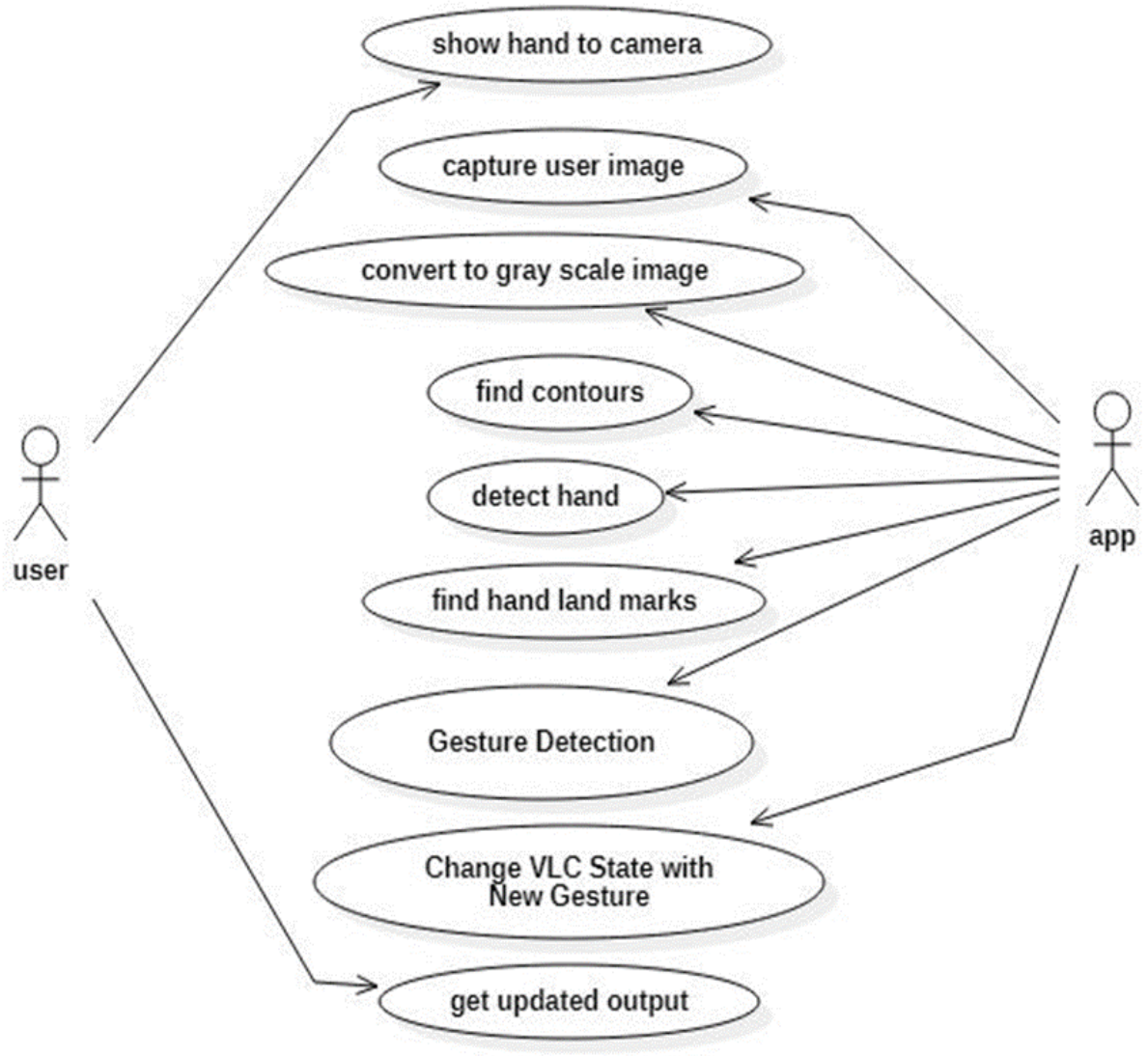
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

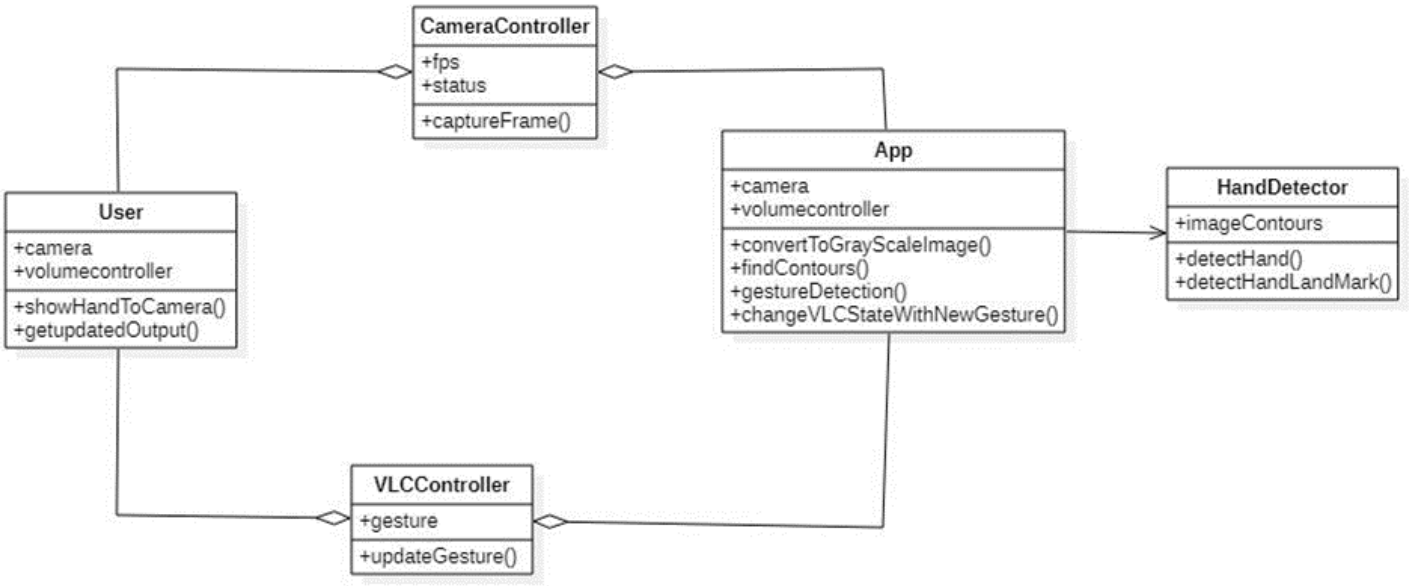
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system

functions are performed for which actor. Roles of the actors in the system can be depicted.



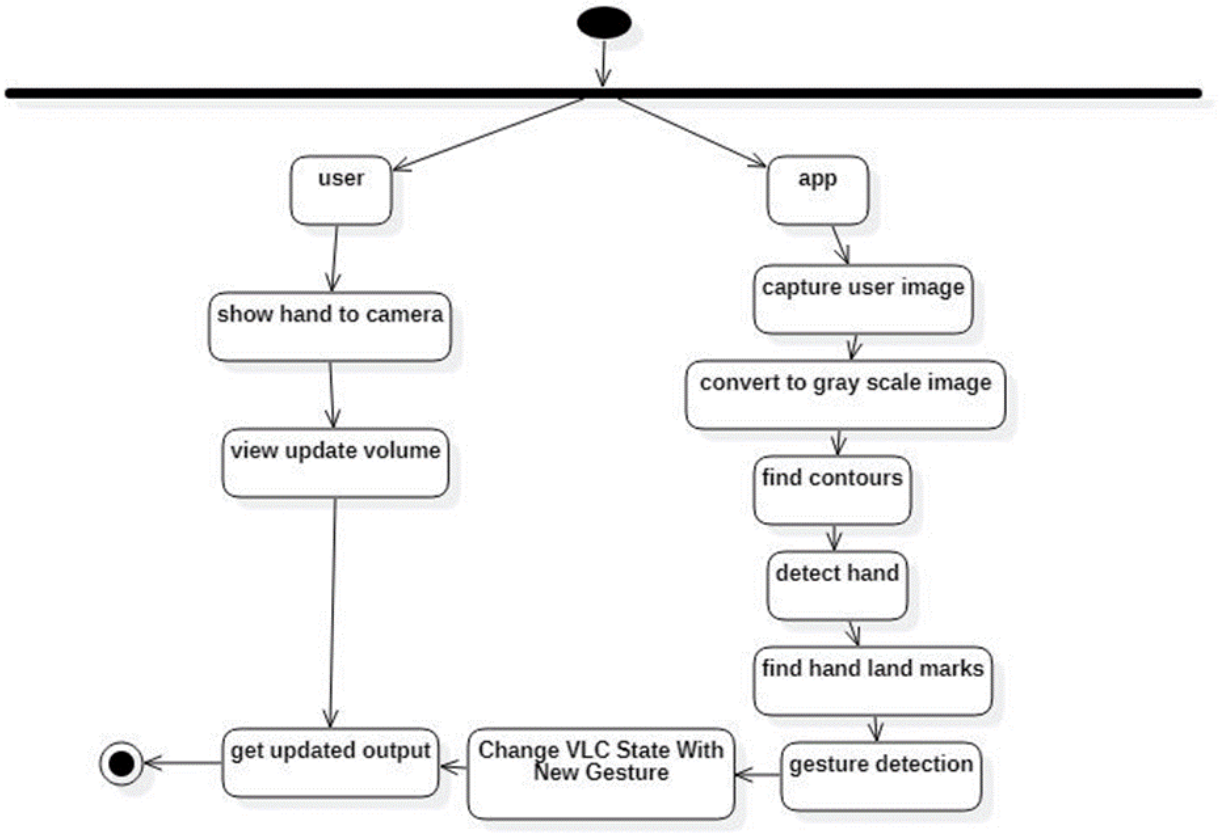
Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



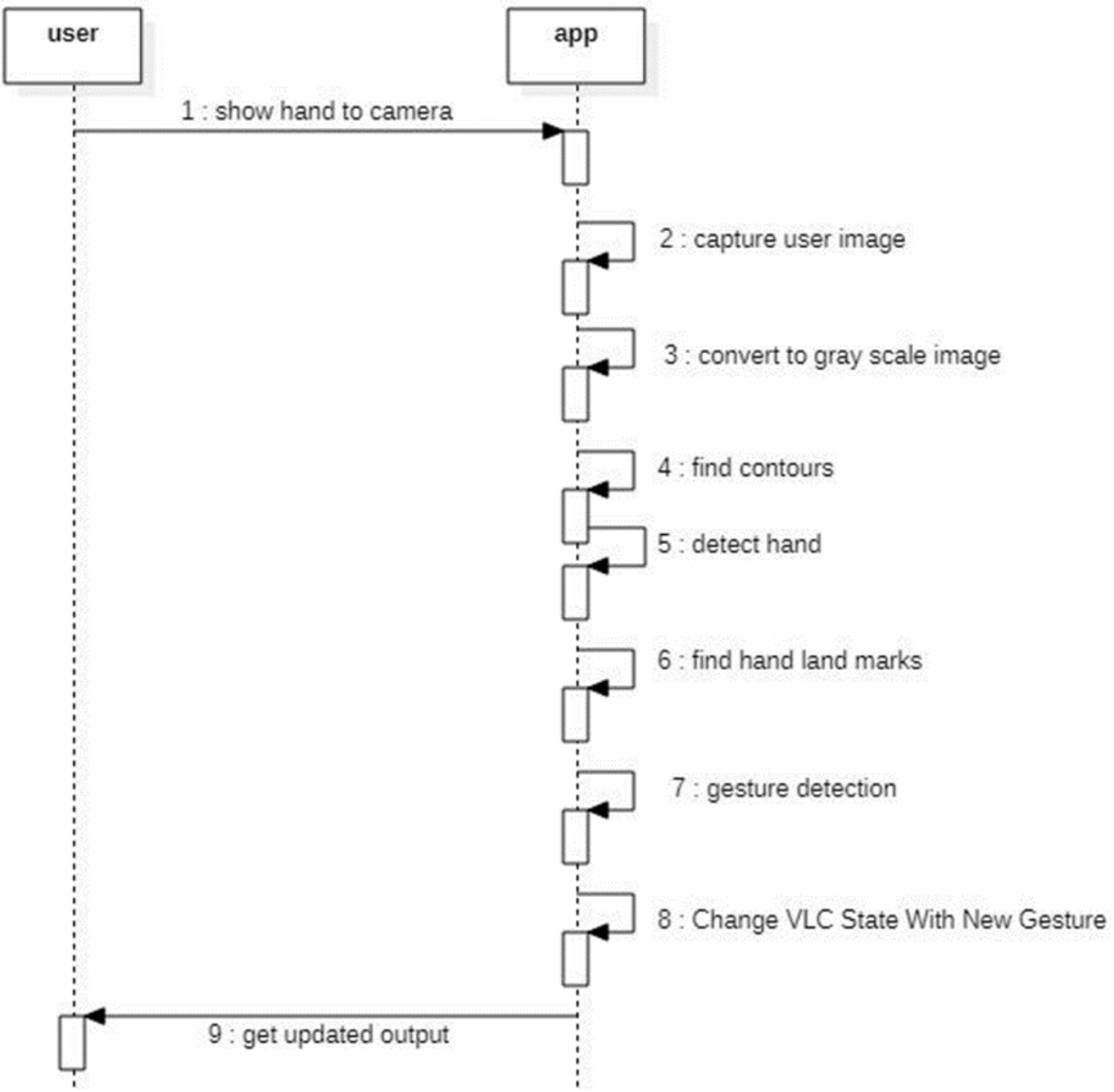
Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



CHAPTER - 5 TESTING & RESULTS

Software Testing is a process of executing the application with the intent to find any software bugs. It is used to check whether the application met its expectations and whether all the functionalities of the application are working. The final goal of testing is to check whether the application/model is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of the application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such a way that the scope of finding the bugs is maximum.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at a component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly

defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

The organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black-box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, like most other kinds of tests, must be written from the definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages, and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.1 System Test Cases:

A test case is a set of test data, preconditions, expected results, and post conditions, developed for a test scenario to verify compliance against a specific requirement. For our project, we have designed and executed a few test cases to check if the project meets the functional requirements.

Tested	**Test name**	**Inputs**	**Expected output**	**Actual Output**	**status**
1	Load Model	Hand Detection Model	Model Loaded Successfully	Model loaded	success
2	Detect Hand	Image Captured	Need to Detected	Hand Bounding	success

		from webcam	Hand Region	Box Detected	
3	Detect Hand Key points	Hand Image	Need to Detect all key points of hand	Key points detected and Visualized	success
4	Detect Hand Gesture	Hand Image	Need to Detect Hand Gesture Based on Key points	Hand Gesture Detected	success
5	Operate VLC	Hand Gesture	Need to Change the state of Vlc Media Player	VLC Player State Changed	success

TEST CONDITION	**INPUT SPECIFICATION**	**OUTPUT SPECIFICATION**	**PASS/FAIL**
The user is validating the object detection model for thermal image	User passes only the thermal image	Displays the image with the objects being detected	PASS

Table 5.1.2.2 Test Objective: Verification of object detection model for thermal image

TEST CONDITION	**INPUT SPECIFICATION**	**OUTPUT**	**PASS/FAIL**

	SPECIFICATION
The user is validating the object detection model for fused image	User passes only the fused image Displays the image with the objects being detected PASS

Table 5.1.2.3 Test Objective: Verification of object detection model for fused image

5.2 Experimental Results

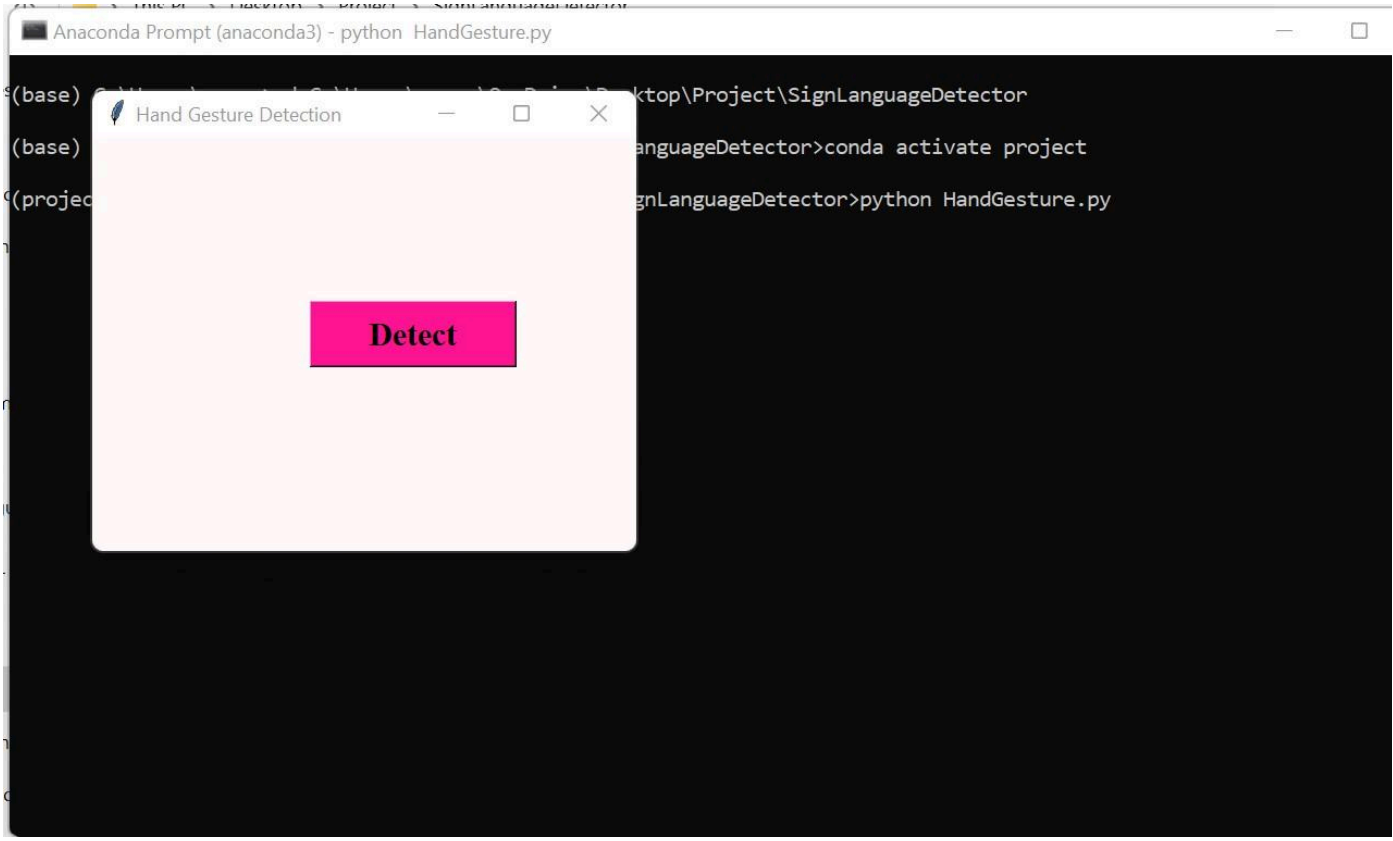


Figure 5.2.1: Detection Image

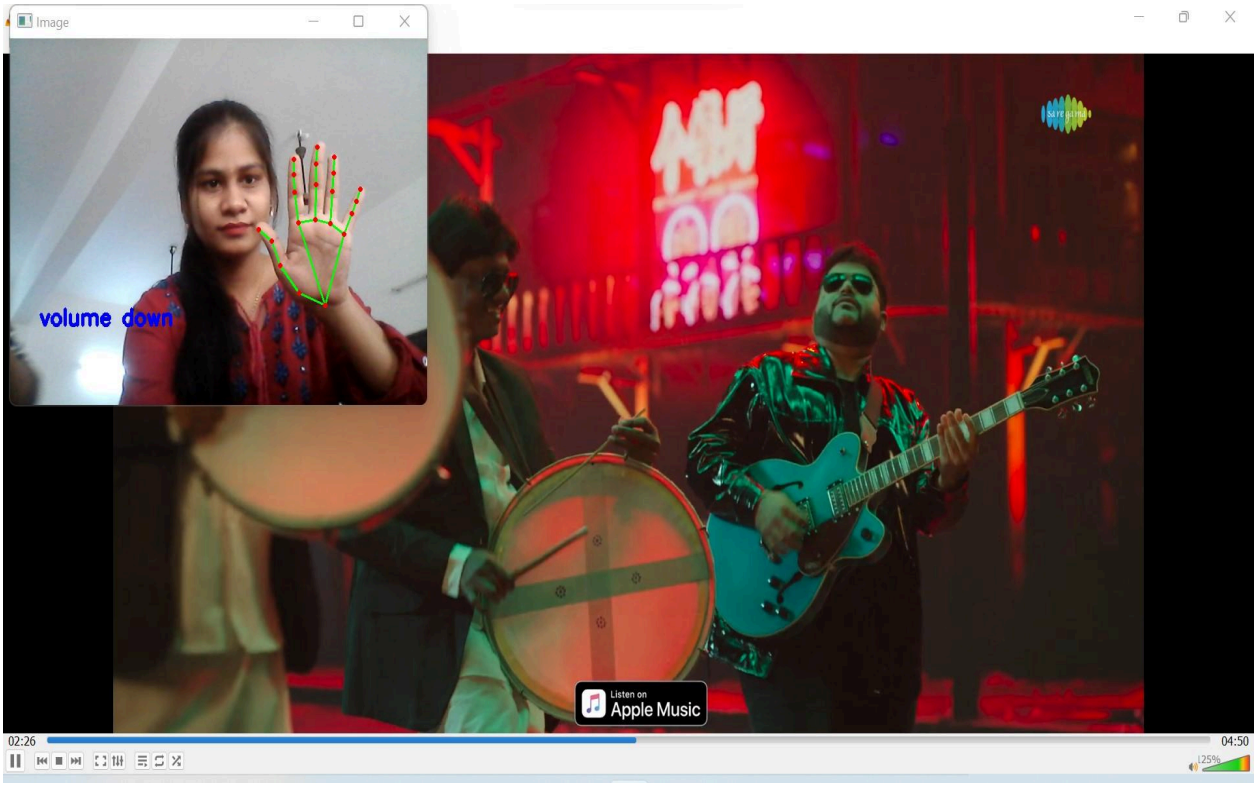


Figure 5.2.2: Volume Down Image

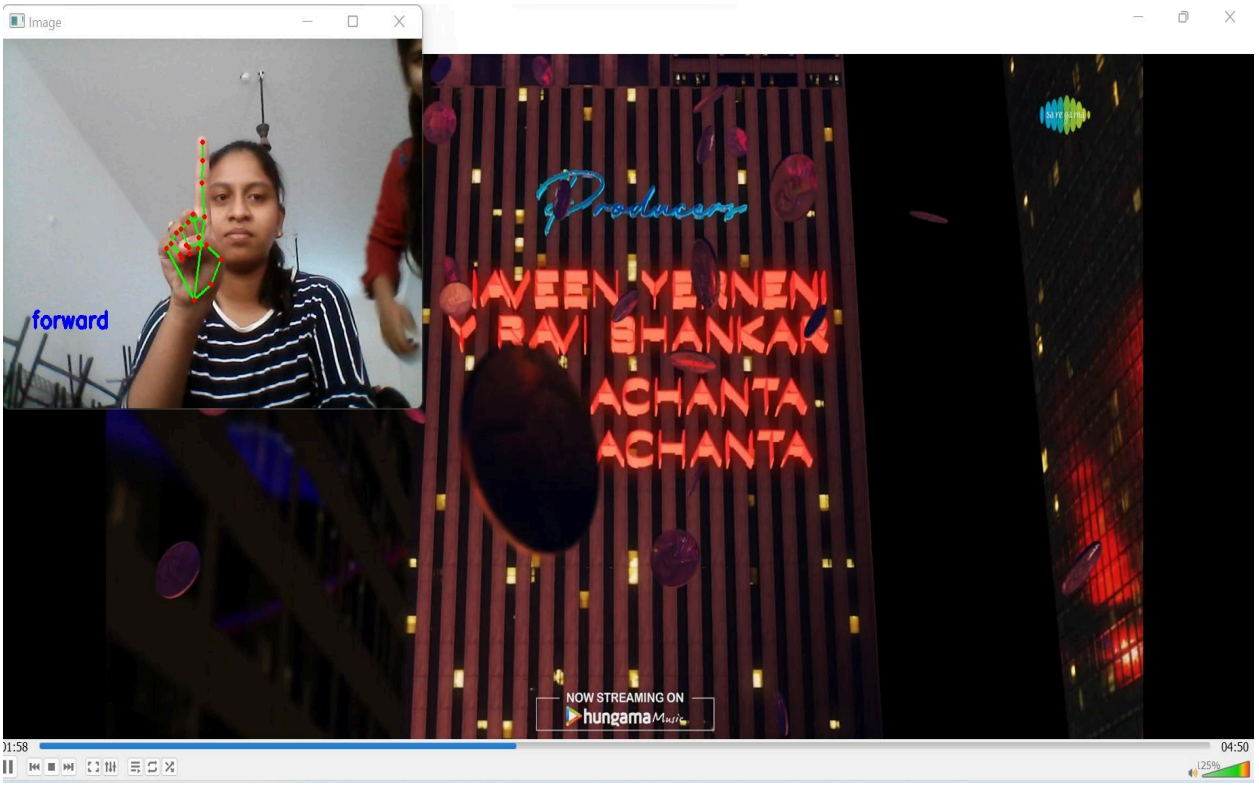


Figure 5.2.3: Forward Image

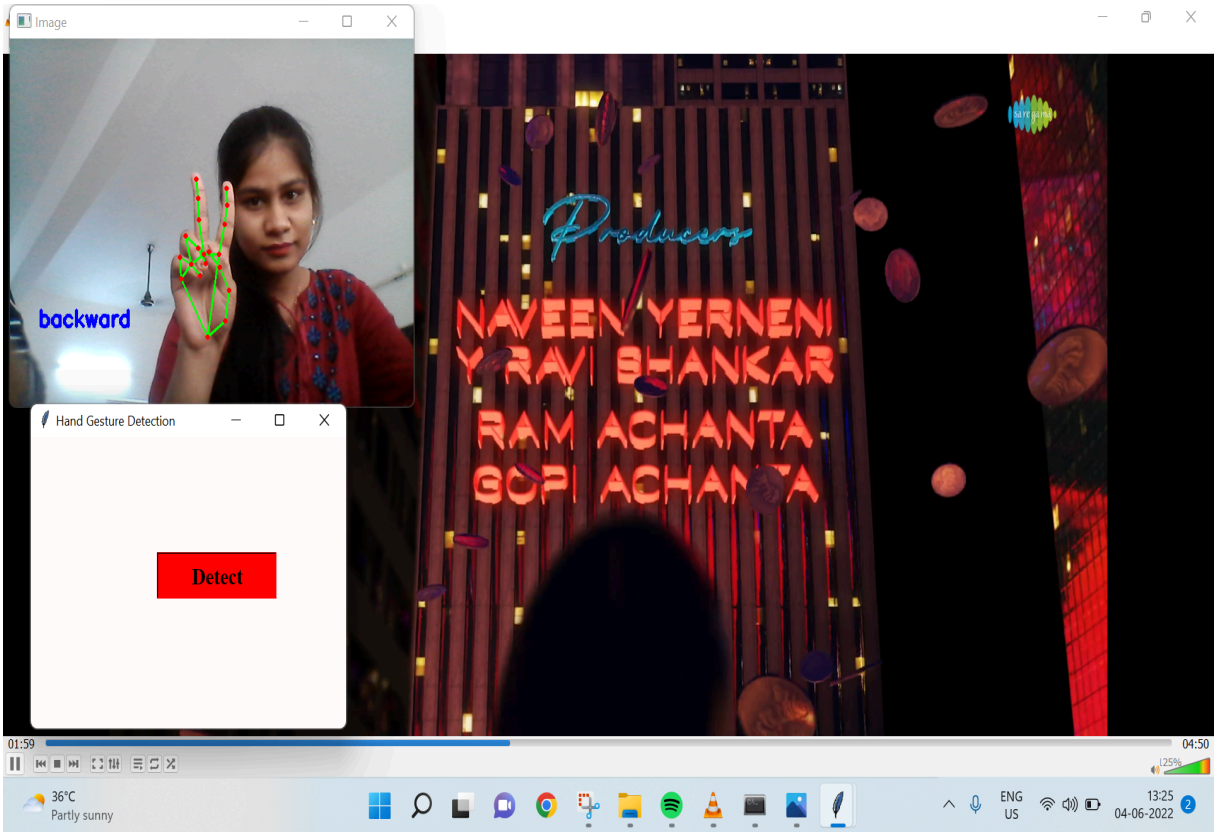


Figure 5.2.4 Backward Image

CHAPTER - 6 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 not many 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.

Depending on the contour defects and making use of OpenCV techniques the gestures are identified and classified into different gestures and there are visual threshold and color representations in the result. During the live video, streaming frames are extracted, processed, and converted into proper images along with region segmentation then these image outputs are supplied as input into the VLC video player to perform the associated action command related to VLC.

Future Scope:

As per our plans, we have completed our project implementation. Next, we may aim to come with more gestures and their associated actions to be performed for building a completely modified VLC media player. A considerable amount of research yet needs to be done in this field as we need to take latency into consideration. The final aim is to reduce the project response time in real life as even a second delay could cause different control to be functioned.

Here we have implemented and restricted our project only for 6 gestures further scope would be to increase the number of gesture recognition and commands for the VLC video player.

CHAPTER - 7 REFERENCES

[1] M. Turk and G. Robertson, “Perceptual user interfaces,” Communications of the ACM, vol. 43(3), March 2000.

[2] A. Nandy, S. Mondal, J. S. Prasad, P. Chakraborty and G. C. Nandi, “Recognizing & interpreting Indian Sign Language gesture for Human Robot Interaction,” 2010 International Conference on Computer and Communication Technology (ICCCT), pp. 712-717, IEEE, Sept. 2010.

[3] Y. Wu and T. S. Huang, "Vision-Based Gesture Recognition: A Review", Lecture Notes in Computer Science, Vol. 1739,pp. 103-115, 1999.

[4] V. Pavlovic, R. Sharma and T. S. Huang, “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review,” IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 19(7), pp. 677-695, 1997.

[5] S. S. Rautaray and A. Agrawal, “A Vision based Hand Gesture Interface for Controlling VLC Media Player,” International Journal of Computer Applications, vol. 10(7), pp. 11–16, November 2010.

[6] C. Ó. Conaire, N. E. O'Connor and A. F. Smeaton, “Detector adaptation by maximizing agreement between independent data sources,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1-6, June 2007.

[7] N. J. B. McFarlane and C. P. Schofield, “Segmentation and tracking of piglets in images,” Machine Vision and Applications, vol. 8, issue 3, pp. 187-193, 1995.

[8] M. J. Jones and J. M. Rehg, “Statistical color models with application to skin detection,” IEEE

Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 274-280, 1999.

[9] S. Battiato, D. Cantone, D. Catalano and G. Cincotti; “An Efficient Algorithm for the Approximate Median Selection Problem,” Algorithms and Complexity Lecture Notes in Computer Science, Springer, vol. 1767, pp. 226-238, 2000.

[10] A. Nandy, J. S. Prasad, S. Mondal, P. Chakraborty and G. C. Nandi, “Recognition of Isolated Indian Sign Language Gesture in Real Time,” Information Processing and Management, Communications in Computer and Information Science, Springer, vol. 70, pp. 102-107, 2010.

[11] A. Nandy, J. S. Prasad, P. Chakraborty, G. C. Nandi, S. Mondal, “Classification of Indian Sign Language In Real Time,” International Journal on Computer Engineering and Information Technology (IJCEIT), vol. 10, issue 15, 2010.

APPENDIX

Code:

```
import cv2 #opencv library(img processing)
import time
import os
import pyautogui as p #keyboard operations
import tkinter as tk   #UI
from tkinter import *

#Window is our Main frame of system
window = tk.Tk()
window.title("Hand Gesture Detection")

window.geometry('500x380')
window.configure(background='snow')


import cv2
import mediapipe as mp #hand detection //cpoints detect
import time

def run(): #1st exectue

     class handDetector():
          def __init__(self, mode=False, maxHands=2, detectionCon=0.5,
trackCon=0.5):
               self.mode = mode
               self.maxHands = maxHands
               self.detectionCon = detectionCon
               self.trackCon = trackCon

               self.mpHands = mp.solutions.hands
               self.hands = self.mpHands.Hands(self.mode, self.maxHands,

self.detectionCon, self.trackCon)
               self.mpDraw = mp.solutions.drawing_utils

          def findHands(self, img, draw=True): #draws the image of hand
               imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
               self.results = self.hands.process(imgRGB) #hand is detected
and put in result
               # 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) #we draw
circle at ech keypoint
                          # print(id, cx, cy)
                          lmList.append([id, cx, cy])
                          if draw:
                               cv2.circle(img, (cx, cy), 15, (255, 0, 255),
cv2.FILLED)

                return lmList


     wCam, hCam = 640, 480

     cap = cv2.VideoCapture(0)
     cap.set(3, wCam)
     cap.set(4, hCam)

     pTime = 0

     detector =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("Result:",totalFingers)

                result=""
                if totalFingers==1:
                     p.press("space")
                     result = "forward"
                elif totalFingers==2 :
                     p.press("left")
                     result="backward"
                elif totalFingers==3:
                     p.press("right")
                     result = "volume up"
                elif totalFingers==4:
                     p.press("up")
                     result = "volume down"
                elif totalFingers==5:
                     p.press("down")
                     result = "volume down"

                cv2.putText(img, str(result), (45, 375),
cv2.FONT_HERSHEY_SIMPLEX,1, (255, 0, 0),3)

           cTime = time.time()
           fps = 1 / (cTime - pTime)
           pTime = cTime

           cv2.imshow("Image", img)
           cv2.waitKey(1000)

clearButton1 = tk.Button(window, text="Detect",command=run,fg="black"
,bg="deep pink"  ,width=10 ,height=1, activebackground = "Red"
,font=('times', 15, ' bold '))
clearButton1.place(x=200, y=150)

window.mainloop()
```

PROJECT RUBRICS

Course code	*Statement* After completing this course the student must demonstrate the knowledge and ability to (Student will be able to)	*Cognitive Level*
CO1	Summarize the survey of the recent advancements to infer the problem statements with applications towards society	Analyze, Evaluate
CO2	Design a software based solution within the scope of the project.	Analyze
CO3	Implement test and deploy using contemporary technologies and tools	Apply,Evaluate,Analyze
CO4	Demonstrate qualities necessary for working in a team.	Apply.Evaluate.Analyze
CO5	Generate a suitable technical document for the project.	Apply,Evaluate,Analyze

Table 1: Course Outcomes - Cognitive levels

Course Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2
CO1	3	3	3	3	3	3	3	1					3	3
CO2	3	3	3	3	3	2	3	1					3	3
CO3	3	2	3	3	3	2	3	1		1			3	3
CO4								1	3	3	3	3		3
CO5								1	2	3	3	3		

Table 2: Course Articulation Matrix

II. Rubrics for Project:

F	C No	Criterion [c]	Exemplary 4	Satisfactory 3	Developing 2	Unsatisfactory 1
	I	**Identify/Define Problem** Ability to identify a suitable problem and define the project objectives.	Demonstrates a skillful ability to identify / articulate a problem and the objectives are well defined and prioritized.	Demonstrates ability to Identify / articulate a problem and All major objectives are identified.	Demonstrates some ability to identify / articulate a problem that is partially connected to the issues and most major objectives are identified but one or two minor ones are missing or priorities are not established.	Demonstrates minimal or no ability to identify / articulate a problem and many major objectives are not identified.
	II	**Collection of Background Information:** Ability to gather background Information (existing knowledge, research, and/or indications of the problem)	Collects sufficient relevant background information from appropriate sources, and is able to identify pertinant/critical information;	Collects sufficient relevant background information from appropriate sources;	Collects some relevant background information from appropriate Sources.	Minimal or no ability to collect relevant background information
	III	**Define scope of the problem** Ability to identify problem scope suitable to the degree considering the impact on society and environment	Demonstrates a skillful ability to define the scope of problem accurately mentioning the relevant fields of engineering precisely. Considers, explains and evaluates the impact of	Demonstrates ability to define problem scope mentioning the relevant fields of engineering broadly. Considers and explains the impact of engineering interventions on society and	Demonstrates some ability to define problem scope mentioning some of the relevant fields . Some consideration of the impact of engineering interventions on society and environment.	Demonstrates minimal or no ability to define problem scope and fails to mention relevant fields of engineering. Minimal or no consideration of the impact of engineering

			engineering interventions on society and environment.	environment		interventions on society and environment

IV	**Understanding the Design Process and Problem Solving:** Ability to explain the design process including the importance of needs, specifications, concept generation and to develop an approach to solve a problem.	Demonstrates a comprehensive ability to understand and explain a design process. Considers multiple approaches to solving a problem, and can articulate reason for choosing solution	Demonstrates an ability to understand and explain a design process. Considers multiple approaches to solving a problem, which is justified and considers consequences.	Demonstrates some ability to understand and explain a design process. Considers a few approaches to solving a problem; doesn't always consider consequences.	Demonstrates minimal or no ability to understand and explain a design process. Considers a single approach to solving a problem. Does not consider consequences.
V	**Implementing Design Strategy and Evaluating Final Design:** Ability to execute a solution taking into consideration design requirements using appropriate tool (software/hardware);	Demonstrates a skillful ability to execute a solution taking into consideration all design requirements using the most relevant tool.	Demonstrates an ability to execute a solution taking into consideration design requirements using relevant tool.	Demonstrates some ability to execute a solution but not using most relevant tool.	Demonstrates minimal or no ability to execute a solution. Solution does not directly attend to the problem.

VI	To evaluate/confirm the functioning of the final design. To deploy the project on the target environment	Demonstrates a skillful ability to evaluate/confirm the functioning of the final design skillfully, with deliberation for further Improvement after deployment.	Demonstrates an ability to evaluate/confirm the functioning of the final design. The evaluation is complete and has sufficient depth.	Ability to evaluate/confirm the functioning of the final design, but the evaluation lacks depth and/or is incomplete.	Demonstrates minimal or no ability to evaluate/confirm the functioning of the final design.

VII	**Proper Use of Others' Work:** Ability to recognize, understand and apply proper ethical use of intellectual property, copyrighted materials, and research.	Always recognizes and applies proper ethical use of intellectual property, copyrighted materials, and others' research.	Recognizes and applies proper ethical use of intellectual property, copyrighted materials, and others' research.	Some recognition and application of proper ethical use of intellectual property, copyrighted materials, and others' research.	Minimal or no recognition and/or application of proper ethical use of intellectual property, Copyrighted materials, or others' research.
VIII	**Individual Work Contributions and Time Management:** Ability to carry out individual Responsibilities and manage time (estimate, prioritize, establish deadlines/ milestones, follow timeline, plan for contingencies, adapt to change).	Designated jobs are accomplished by deadline; completed work is carefully and meticulously prepared and meets all requirements.	Designated jobs are accomplished by deadline; completed work meets requirements.	Designated jobs are accomplished by deadline; completed work meets most requirements.	Some Designated jobs are accomplished by deadline; completed work meets some requirements.

IX	**Leadership Skills**: Ability to lead a team. (i) Mentors and accepts mentoring from others. (ii) Demonstrates capacity for initiative while respecting others' roles. (iii) Facilitates others' involvment. (iv) Evaluates team Effectiveness and plans for improvements	Exemplifies leadership skills.	Demonstrates leadership skills.	Demonstrates some leadership skills at times.	Demonstrates minimal or no leadership skills.

X	**Working with Others:** Ability to listen to, collaborate with, and champion the efforts of others.	Skillfully listens to, collaborates with, and champions the efforts of others.	Listens to, collaborates with, and champions the efforts of others.	Sometimes listens to, collaborates with, and champions others' efforts.	Rarely listens to, collaborates with, or champions others' efforts.
XI	**Technical Writing Skills** Ability to communicate the main idea with clarity. Ability to use illustrations properly to support ideas (citations, position on page etc)	Main idea is clearly and precisely stated. Materials are seamlessly arranged in a logical sequence Illustrations are skillfully used to support ideas	Main idea is understandable. Material moves logically forward, Illustrations are properly used to support ideas	Main idea is somewhat understandable. Materical has some logical order and is somewhat coherent or easy to follow. Illustrations are for the most part properly used to support ideas	Main idea is difficult to understand. Material has little logical order, and is often unclear, inconherent. Illustrations are used, but minimally support ideas. (not properly cited etc)
XII	**Communication Skills for Oral Reports** Ability to present strong key ideas and supporting details with clarity and concision. Maintain contact with audience, and ability to complete in the allotted time	Presentation logically and skillfully structured. Key ideas are compelling, and articulated with exceptional clarity and concision. Introduction, supporting details and summary are clearly evident and memorable, and ascertain the credibility of the speaker Presentation fits perfectly within time constraint.	Presentation has clear structure and is easy to follow. Key ideas are clearly and concisely articulated, and are interesting. There is sufficient detail to ascertain speaker's authority, and presentation includes an introductionand summary.Presentation fits within time constraint, though presenter might have to subtly rush or slow down.	Presentation has some structure. Key ideas generally identifiable, although not very remarkable. Introduction, supporting details and/or summary may be too broad, too detailed or missing. Credibility of the speaker may be questionable at times. Presentation does not quite fit within time constraint; presenter has to rush or slow down at end	Presentation rambles. Not organized; key ideas are difficult to identify, and are unremarkable. No clear introduction, supporting details and summary. Speaker has no credibility. Presentation is unsuitably short or unreasonably long.

XIII	Use of software project management principles and tools (versioning, time schedules etc)	Employ all the appropriate tools or engineering techniques. Clearly demonstrates mastery of several areas of the curriculum.	Employ the appropriate tools or engineering techniques	Employ some management tools	Does not make use of any tool
XIV	**Extend Scope of Work:** Ability to extend the project through implementation in other study areas	Demonstrates a skillful ability to explore a subject/topic thoroughly, discusses the road map to extend the project in other areas.	Demonstrates an ability to explore a subject/topic, and shows possible areas in which project can be extended	Demonstrates some ability to explore a subject/topic, providing some knowledge of areas in which project can be extended	Demonstrates minimal or no ability to explore a subject/topic, and does not discuss future work clearly mentioning other areas

III. Rubrics Evaluation: