VOLUME CONTROL USING GESTURES

A project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

G.NAGALAKSHMI (N180201)

T.LEELAVATHI(N180197)

E.SIVANI(N180202)

P.LIKITHA SRI(N181156)

M.RAMYA (N181140)

Under the Esteem Guidance of

Dr.D.V.NAGARJUNA DEVI

Assistant Professor, CSE Dept.

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

<u>CERTIFICATE OF COMPLETION</u>

This is to certify that the project entitled, "**VOLUME CONTROL USING GESTURES**" is work of "**G.NAGALAKSHMI(N180201) and T.LEELAVATHI (N180197) and E.SIVANI (N180202) and P.LIKITHA SRI(N181156) and M.RAMYA(N181140)** " is a recent record of bonafide work carried out by under our guidance and supervision for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session, February2023-July2023 at RGUKT – Nuzvid. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

-----------------------	------------------------
Dr.D.V.Nagarjuna Devi,	Examiner,
Project Supervisor,	Department of CSE,
Department of CSE,	RGUKT – Nuzvid.
RGUKT – Nuzvid.	

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

<u>CERTIFICATE OF EXAMINATION</u>

This is to certify that the work entitled,” **VOLUME CONTROL USING GESTURES**” is work of “**G.NAGALAKSHMI(N180201) and T.LEELAVATHI (N180197) and E.SIVANI (N180202) and P.LIKITHA SRI(N181156) and M.RAMYA(N181140)**” and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in third year of Bachelor of Technology for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn, as recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

Dr.D.V.Nagarjuna Devi,
Project Supervisor,
Department of CSE,
RGUKT – Nuzvid.

Examiner,
Department of CSE,
RGUKT – Nuzvid.

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

DECLARATION

We **" G.NAGALAKSHMI(N180201) and T.LEELAVATHI (N180197) and E.SIVANI (N180202) and P.LIKITHA SRI(N181156) and M.RAMYA(N181140)"** hereby declare that the project report entitled done by us under the guidance of **Dr.Nagarjuna Devi** is submitted for the partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering during the academic session **February 2023- July 2023** at RGUKT-Nuzvid.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. 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.

Date:13-07-2023
Place:Nuzvid

G.NAGALAKSHMI(N180201)
T.LEELAVATHI(N180197)
E.SIVANI(N180202)
P.LIKITHA SRI(N181156)
M.RAMYA(N181140)

ACKNOWLEDGEMENT

We would like to express our profound gratitude and regards to our guide " **Dr.D.V. Nagrjuna Devi** " for her exemplary guidance, monitoring and constant encouragement to us throughout the B.Tech course. We shall always cherish the time spent with her during the course of this work due to the invaluable knowledge gained in the field of Data Science.

We are extremely grateful for the confidence bestowed in us and entrusting our project entitled **"VOLUME CONTROL USING GESTURES"**. We express gratitude to Dr. Sadu Chiranjeevi (HOD of CSE) and other faculty members for being a source of inspiration and constant encouragement which helped us in completing the project successfully.

Finally, yet importantly, we would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

CONTENTS

ABSTRACT

The purpose of Volume Control using Gestures project is to control volume of the system using hand gestures.The system consists of a camera to recognise the gesture taken as input by the user. The main goal of hand gesture recognition is to create a system which can identify the human hand gestures and use same input as the information for controlling the device volume.Hand gestures are used to control the basic operation of a computer like increasing and decreasing volume.Therefore,people will not have to learn machine-like skills which are a burden most of the time.Hand gesture systems provides a natural and innovative modern way of non-verbal communication.These systems has a wide area of application in human computer interaction.In this project the system can be controlled by hand gesture without making use of the keyboard and mouse which are the traditional approaches to control the device.

1.Introduction

This introduction emphasizes the significance of hand gestures as a powerful communication medium in Human-Computer Interaction (HCI) and discusses the limitations of traditional input devices. The proposed system aims to utilize hand gestures for interaction with computers, using a desktop or laptop interface utilizing a web camera to record hand gestures. The report highlights the importance of implementing hand tracking systems and discusses various sensor devices used for digitizing hand and finger motions. The challenges in hand gesture recognition systems are addressed, including background images/videos and that can affect gesture recognition.The process of segmenting images to identify connected regions is explained. The report mentions the use of important packages such as OpenCV Python, MediaPipe, imutils, SciPy, and NumPy.

1.1 Purpose

The purpose of the project is to discuss a volume control using hand gesture recognition system based on detection of hand gestures. In this, the system consists of a high resolution camera to recognise the gesture taken as input by the user. The main goal of hand gesture recognition is to create a system which can identify the human hand gestures and use the same input as the information for controlling the device and by using real time gesture recognition specific users can control a computer by using hand gestures in front of a system video camera linked to a computer.

1.2 Overview

This is a Python based application which helps the user allow for convenient and flexible control of audio of the system without the need for physical connections.By eliminating the need for physical connections, users can control audio of the system using hand gestures to increase and decrease the volume.

1.3 Scope

The scope of this application is wide as it can be used without any physical connection.This application helps the user to change his/her system’s audio using seamless integration.

2 Analysis of Existing and Proposed Systems

2.1 Problem Statement

Gestures play a very important role in communication Information among humans. Nowadays new technologies of Human Computer Interaction (HCI) are being developed to deliver user's instructions to the Machines. Users can communicate with machines using hand, facial expressions, voice and touch. Gesture recognition has attracted a lot of research people interested in computer vision, pattern recognition, and human– computer interaction.

The main aim of the project is to use one of the important modes of interaction that is to control the system using hand gestures.

2.2 Existed System

The existing system will increase or decrease the volume by using physical interaction. He/she manually changes the volume.It uses physical interaction with the system to change the volume.

2.2.1 Disadvantages of Existed System

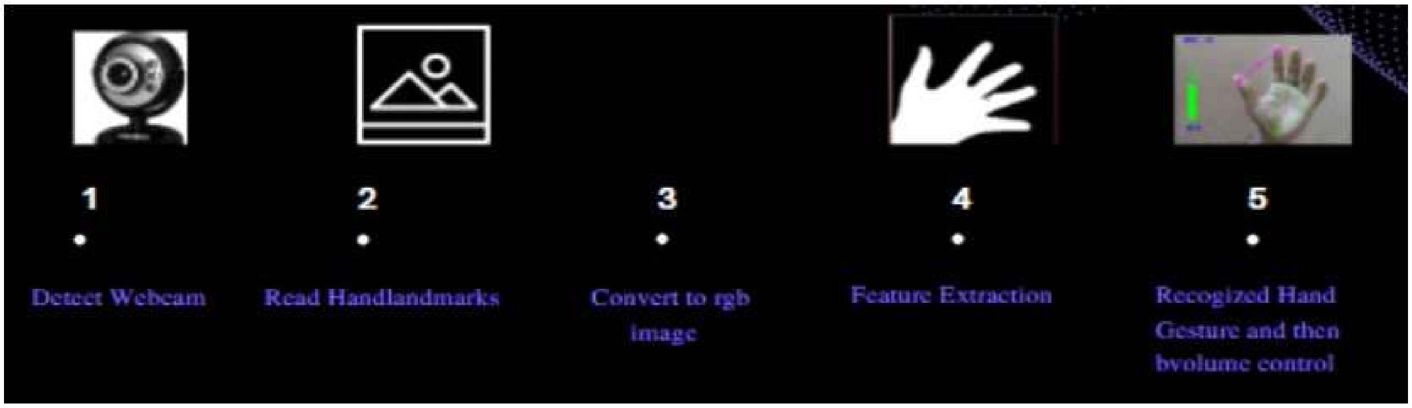
- Physical interaction

  Manually users have to increase or decrease the volume.

3. Proposed System

Considering the issues in the existing system our proposed method for the volume control using gestures project involves acquiring hand gestures, preprocessing and segmenting the gestures, recognizing the gestures using machine learning algorithms, mapping the recognized gestures to specific audio control commands, wirelessly transmitting the commands to the audio devices, executing the control actions, providing feedback to the user, and continuously refining the system based on user feedback. This method aims to enable accurate and responsive gesture recognition, seamless wireless communication, and intuitive control of audio devices to enhance the overall user experience.

- Detect hand landmarks
- Calculate the distance between thumb tip and index finger tip.
- Map the distance of thumb tip and index finger tip with volume range. For my case, distance between thumb tip and index finger tip was within the range of 50 – 300 and the volume range was from 0-100
- In order to exit press “ctrl+c”.



3.1 Advantages of Proposed System

● Natural way for users to interact with system through hand gestures
●Eliminating the need for physical buttons
● Accuracy around 98%
● Accessible with seamless integration

4. Requirements

4.1 Hardware Requirements

RAM : 4 GB(approx)
Disk : 320GB(approx)
Camera : Web Camera 10 Megapixels
Processor : Intel core I5

4.2 Software Requirements

Operating System : 64 bit windows
Code Editor : Python IDLE,Visual studio code
Programming Language : Python

4.3 Technologies

4.3.1 Python

It is Commonly used for developing websites and software,task automation,data analysis and data visualization. In python we used different kinds of libraries.

4.3.1.1 Numpy

It is very useful for fundamental scientific computations in Machine Learning.NumPy is a very popular python library with the help of a large collection of high-level mathematical functions.

4.3.1.2 Mediapipe

MediaPipe library is a versatile tool used for computer vision and machine learning tasks. It offers components for gesture recognition, facial recognition and tracking, object detection and tracking, augmented reality applications, body tracking, and audio processing. Developers can utilize MediaPipe's functionalities to build real-time multimedia processing pipelines for various applications.

4.3.1.3 PYCAW

Pycaw library is a python interface that allows developers to interact with the core audio API in windows. It provides tools for controlling and managing audio devices including volume control, playback and device information retrieval. PYCAW simplifies audio related operations in windows applications by offering convenient access to the core audio functionality.

4.3.1.4 opencv

OpenCV is a Python library that allows you to perform image processing and computer vision tasks. It provides a wide range of features, including object detection, face recognition, and tracking. In this OpenCV Tutorial in Python, we'll be learning more about the library.

4.3.1.5 Ctypes

The Ctypes library in Python enables developers to interface with c or c++ libraries,facilitating the calling of functions and usage of data types from within Python code.It serves as a bridge between Python and low-level languages,allowing access to system APIs,hardware interfaces,and integration with existing c/c++ libraries in python applications.ctypes is valuable for tasks requiring interaction with native code and external libraries.

4.3.1.6 Comtypes

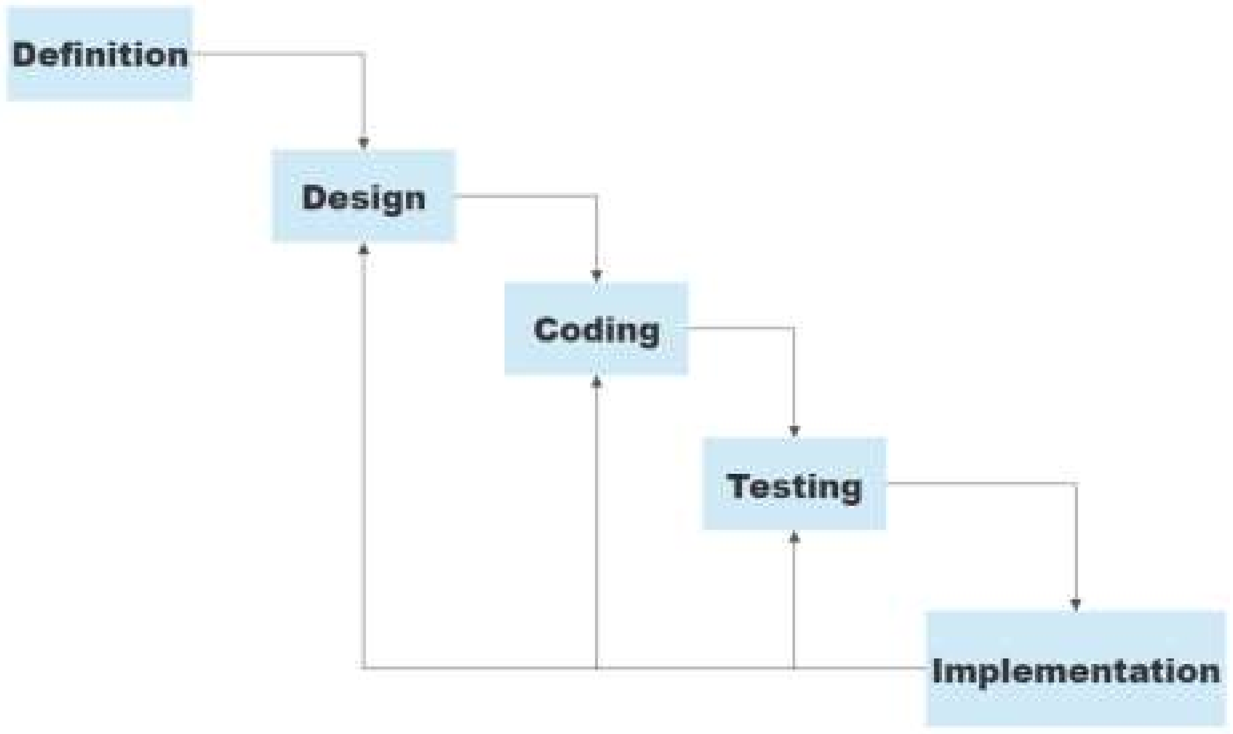
comtypes is a Python package that provides a convenient way to access and interact with COM (Component Object Model) components in Windows. COM is a technology developed by Microsoft that allows different software components to communicate with each other.The comtypes package allows you to create, manipulate, and call methods on COM objects using Python. It provides an interface to the underlying COM infrastructure, allowing you to work with COM objects and interfaces in a Pythonic way.

5.Methodology

Methodology is about the study of methods that have been used in order to complete a task or project. Thus, in this section, the project methodology will be discussed in two aspects of methodology, which is the system overview or the flow of the sub tasks and the development tools that have been used for developing the emotion aware music system. In the system overview, it would be more into the lifecycle model that had been identified as appropriate for the project while in the development tools section; it will give a brief explanation on what development.

5.1 Procedure Identification

As for this project, Iterative waterfall methodology has been considered to be applied as the methodology. An Iterative waterfall methodology structures a project into distinct phases with defined deliverables from each phase. The phases are definition, design, coding, testing and implementation stages. Below is the figure of the Iterative waterfall model.



5.1.1 Definition Stage

The first phase of this project is to try to capture what the system will do (its requirements) based on milestones. This is to ensure the system will be on track. Here, comprehensive research has been done to study the basic concept of the system itself. During this stage, the problem definition and problem statement of the system have been started.

5.1.2 Design Stage

The second stage determines how it will be designed. All the functionalities required are designed in this phase.

5.1.3 Coding Stage

In the middle of the stage the actual programming started. Here, the appropriate software and hardware tools as mentioned below are being applied in order to facilitate the project development process.

5.1.4 Testing Phase

The fourth phase is the full system testing where the stages of testing like unit testing, integration testing, system testing and user acceptance testing are involved in order to ensure quality of the system.

5.1.5 Implementation Stage

The final phase is focused on implementation tasks such as go-live, training, and documentation. Here, the documentation is being prepared to conclude on the overall research and experiment, which is basically to know which method can be applied.

6. Main Functionality

Wireless sound control is a python-based application, which helps the user to control the system audio using hand gestures. Wireless sound control acts as a medium to control the system using seamless integration .

6.1 Image Capturing

In order to utilize this application, the user have to give access to the webcam.It involves capturing images or sequences of images(videos) to preserve moments.

6.2 Image Classification

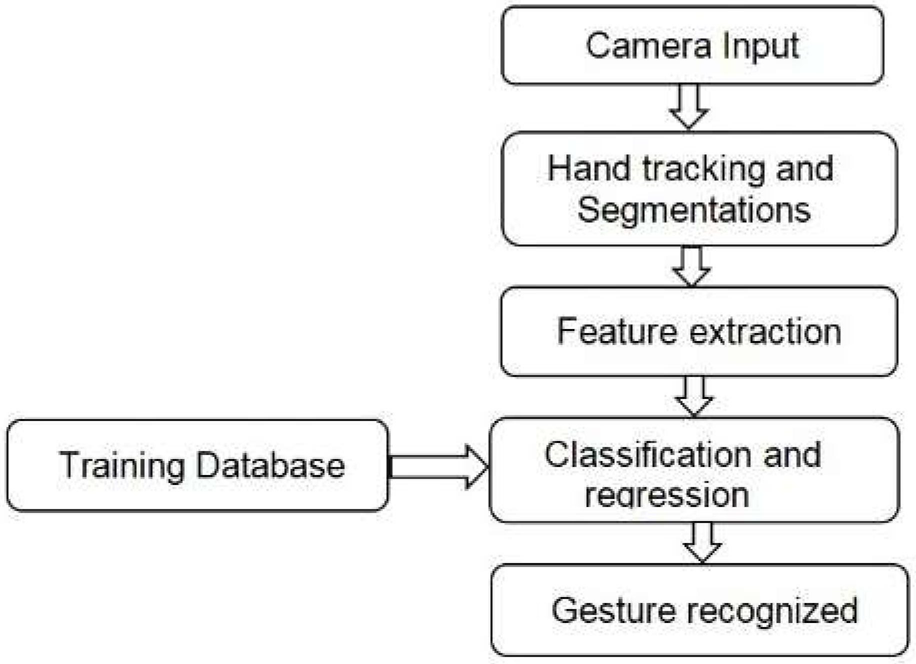
It enables users to control audio devices or systems using gestures and movements of their hands.It involves recognizing and interpreting hand gestures.After successfully capturing the image, it detects the hand and fingers.

6.3 Volume control based on hand gestures

After recognizing and interpreting hand gestures,mapping them to audio commands.Based on the distance between index finger and thumb it adjusts the system's volume from maximum to minimum.

7. Architecture

7.1 Architecture diagram



7.1.1 Camera Input

The camera in our device is used for this project. It detects our hand with points in it so as it can see the distance between our thumb finger tip and index finger tip. The distance between the thumb_tip and Index_finger_tip is directly proportional to the volume of the device. It offers a wide range of capabilities, including image and video input/output, image filtering, object detection, feature extraction, and more.

```
cap = cv2.VideoCapture(0)
```

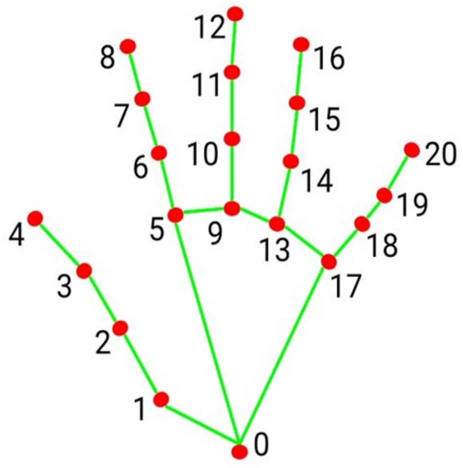
The cv2.VideoCapture(0) statement initializes a video capture object to capture video from the default camera (usually the webcam) by passing 0 as the argument. The image is captured and then converted to RGB and completes the processing of the image.

```
while True:
        success , img = cap.read()
        imgRGB = cv2.cvtColor(img , cv2.COLOR_BGR2RGB)
        results = hands.process(imgRGB)
```

7.1.2 Hand tracking and segmentations:

MediaPipe is an open-source library created by Google that uses machine learning for tasks like recognizing faces and gestures. MediaPipe Hands is a part of this library and focuses on accurately tracking hands and fingers. It relies on machine learning to estimate the positions of 21 important 3D points on a hand from a single image. This allows us to extract the coordinates of these key points and analyze hand movements.

```
mpHands = mp.solutions.hands
hands = mpHands.Hands()
mpDraws = mp.solutions.drawing_utils
```



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

7.1.3 Feature Extraction:

The feature extraction process in the provided code involves extracting hand landmarks and calculating the length of a specific hand gesture. The code uses the Mediapipe library to detect and track hand landmarks in real-time. For each detected hand, the coordinates of specific landmarks, such as the thumb tip and index finger tip, are extracted.

```
x1 , y1 = lmList[4][1] , lmList[4][2]
x2 , y2 = lmList[8][1] , lmList[8][2]
```

Draw circles at the thumb and index finger tips and a line connecting them. The distance between these landmarks is then calculated to determine the length of the gesture. These extracted features, the landmark coordinates and gesture length, are utilized for volume control based on hand gestures.

7.1.4 Classification and regression:

The classification tasks involve hand landmark detection and regression tasks in distance calculation.
Classification: The code uses the Mediapipe library for hand landmark detection. It identifies and assigns unique IDs to specific landmarks on the hand, such as the thumb tip and index finger tip. These landmarks are crucial for recognizing hand gestures accurately.
Regression: The code utilizes regression for calculating the distance between the thumb tip and index finger tip. By applying the math.hypot() function, the code measures the Euclidean distance between these landmarks, providing a numerical value that represents the length of the hand gesture.

length = math.hypot(x2- x1 , y2- y1)

7.1.5 Gesture recognized:

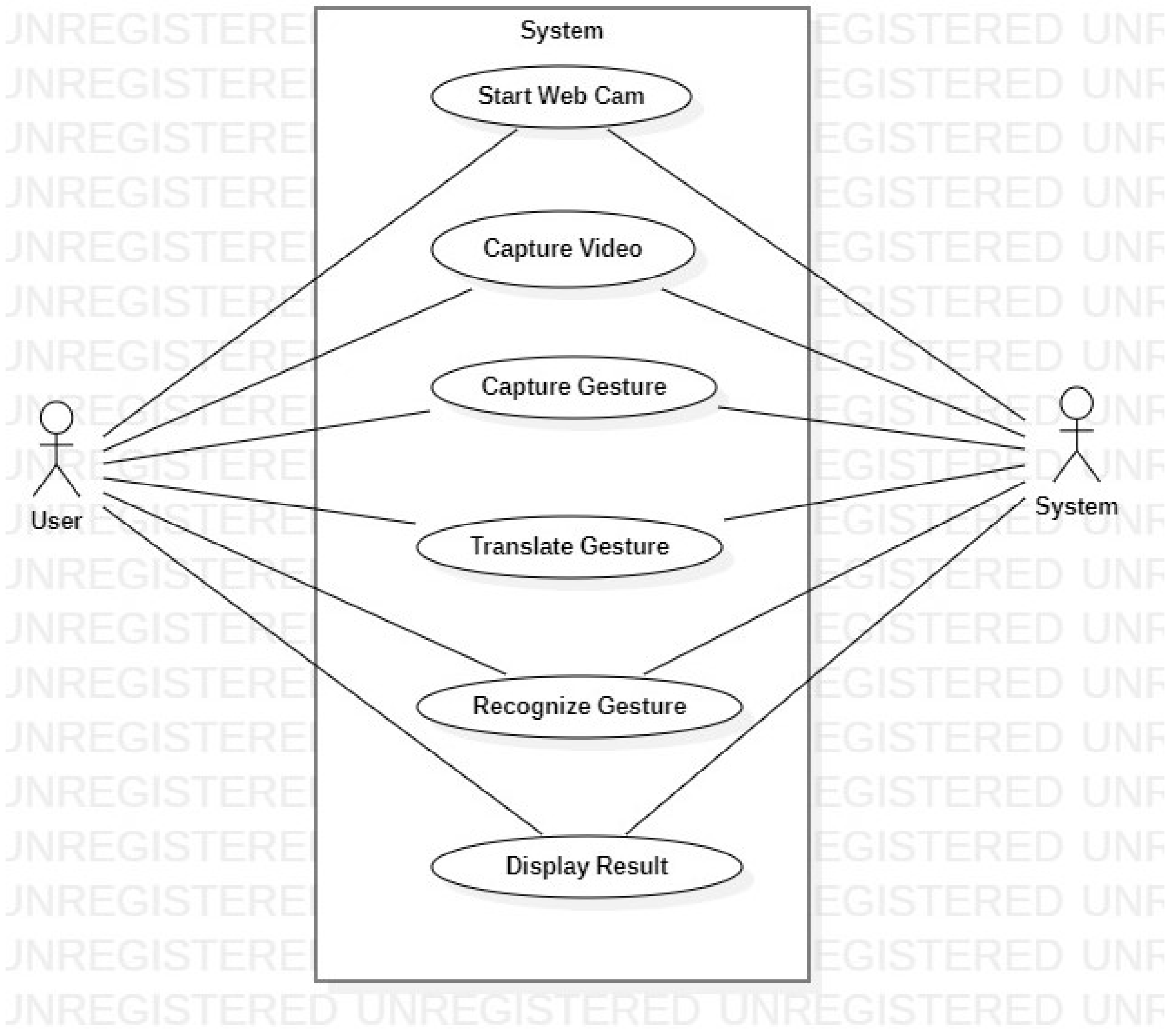
The hand gesture recognized is a pinch gesture, where the thumb tip and index finger tip come close together within a certain distance. The code detects this pinch gesture by calculating the distance between the thumb tip and index finger tip using the **math.hypot()** function. If the distance between the thumb tip and index finger tip is less than 50 , it is considered a pinch gesture. In response to this gesture, the code draws a white circle at the midpoint between the thumb tip and index finger tip.This pinch gesture is used to control the volume of the system. The calculated distance is then mapped to a specific volume range using the **np.interp()** function. Based on the mapped volume value, the code adjusts the master volume level of the system using the pycaw library.

7.1.6 Training Database:

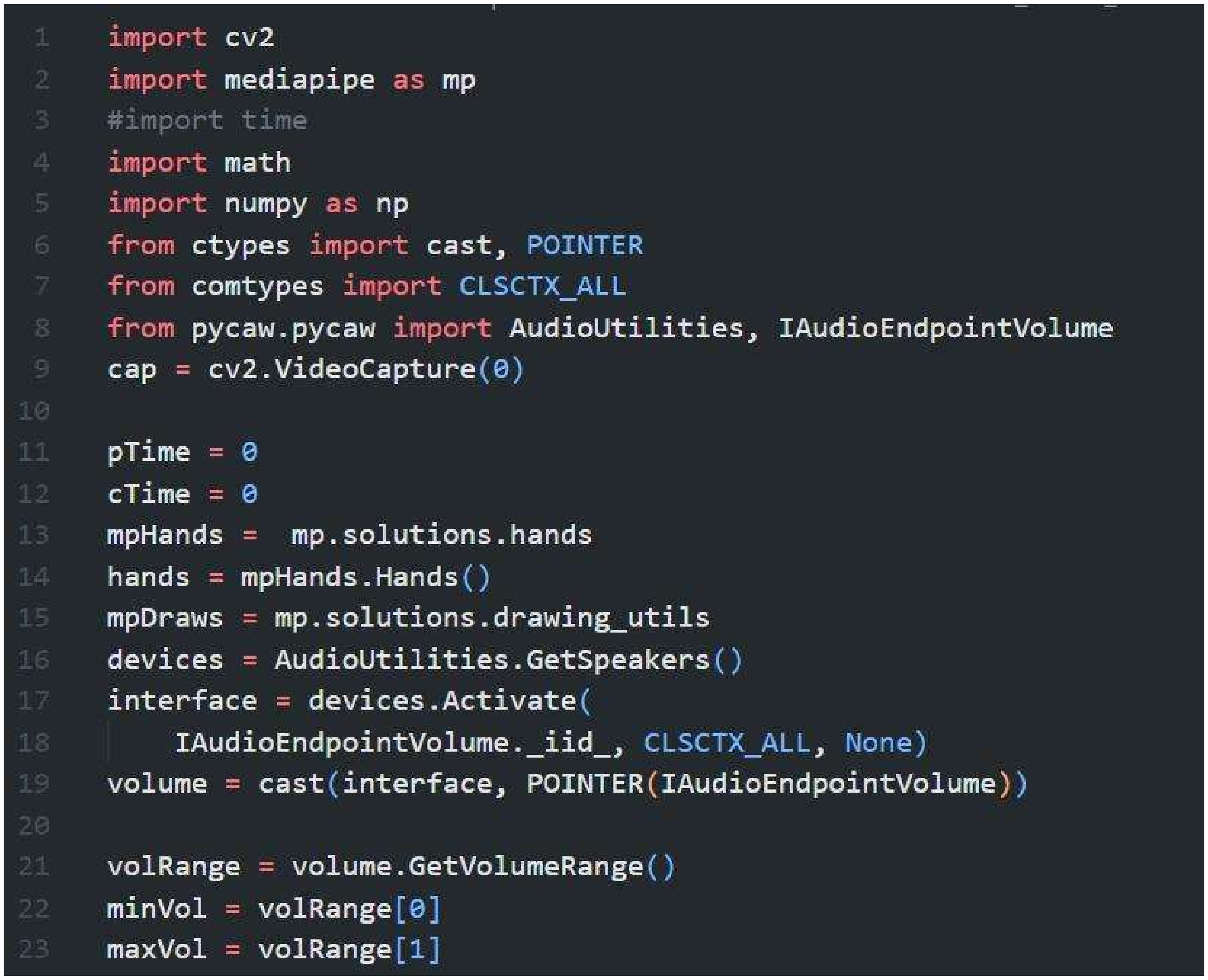
The project code focuses on real-time hand gesture recognition and volume control using hand landmarks. The code utilizes pre-trained models from the Mediapipe library for hand detection and tracking. These models are already trained on large datasets to accurately detect hand landmarks. To converting hand range to the volume range it uses:

```
vol = np.interp(length, [50, 300], [volMin, volMax])
volPer = np.interp(length , [50 , 300] , [0,100])
```

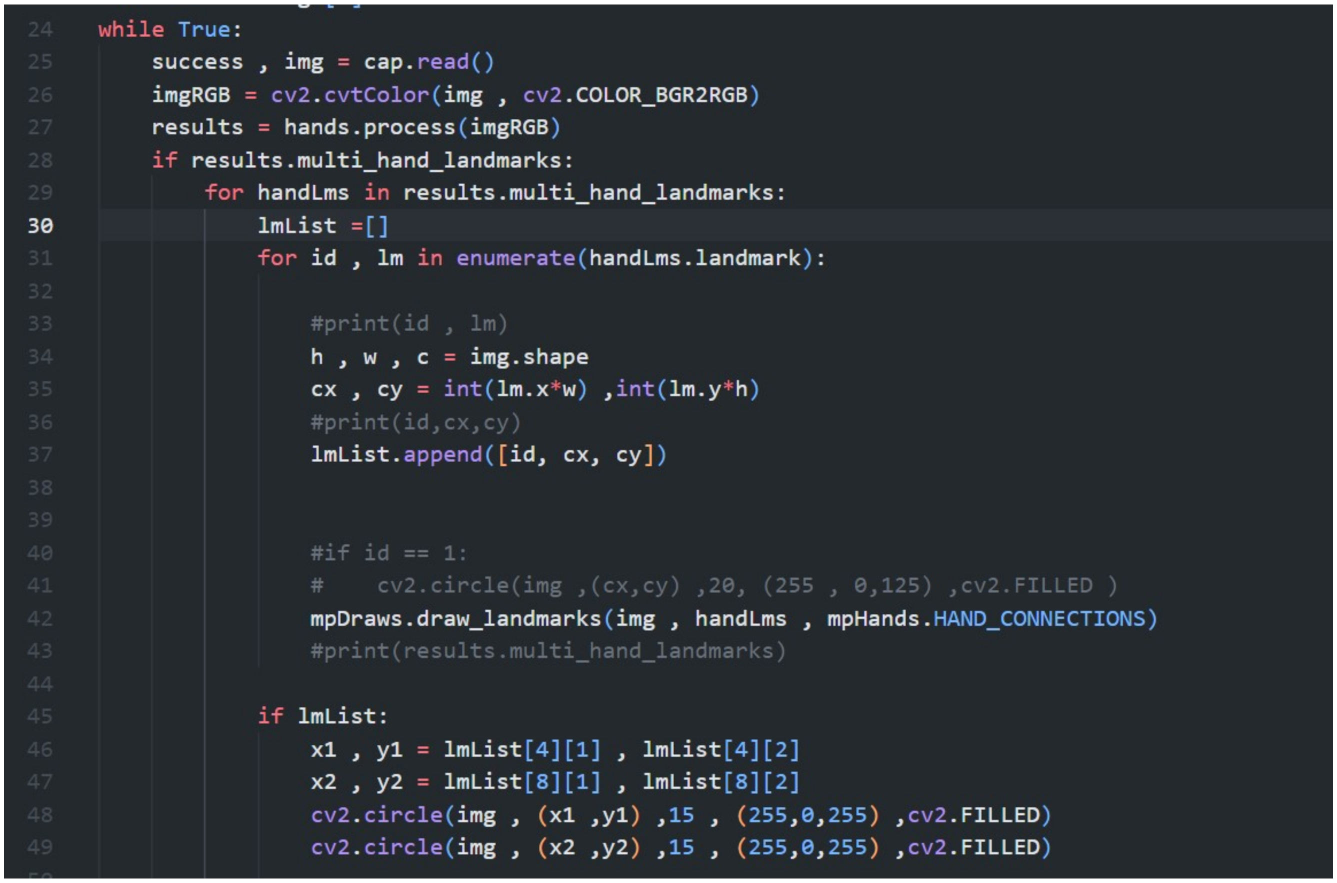
7.2 Use case diagram



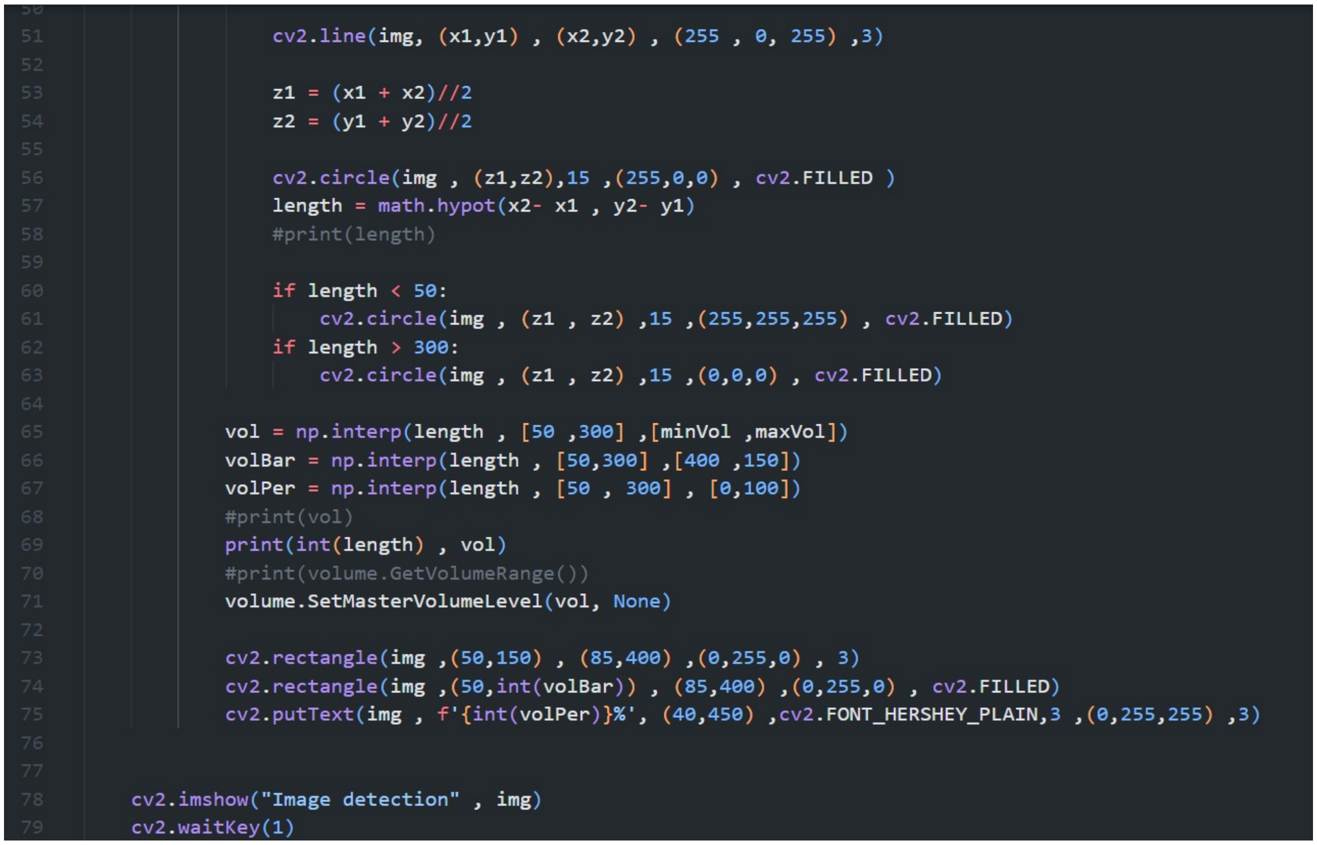
8.Implementation code:



```python
import cv2
import mediapipe as mp
#import time
import math
import numpy as np
from ctypes import cast, POINTER
from comtypes import CLSCTX_ALL
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume
cap = cv2.VideoCapture(0)

pTime = 0
cTime = 0
mpHands =  mp.solutions.hands
hands = mpHands.Hands()
mpDraws = mp.solutions.drawing_utils
devices = AudioUtilities.GetSpeakers()
interface = devices.Activate(
    IAudioEndpointVolume._iid_, CLSCTX_ALL, None)
volume = cast(interface, POINTER(IAudioEndpointVolume))

volRange = volume.GetVolumeRange()
minVol = volRange[0]
maxVol = volRange[1]
```

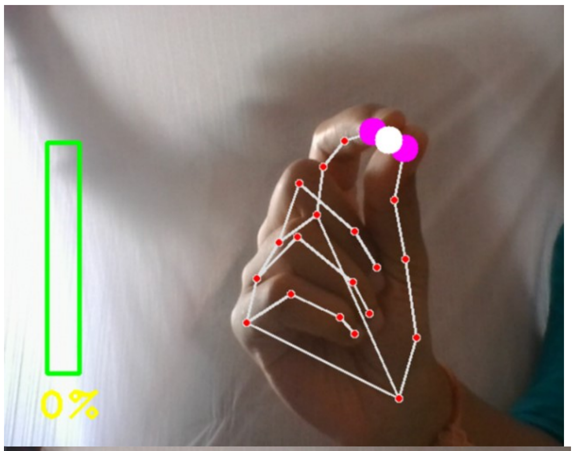


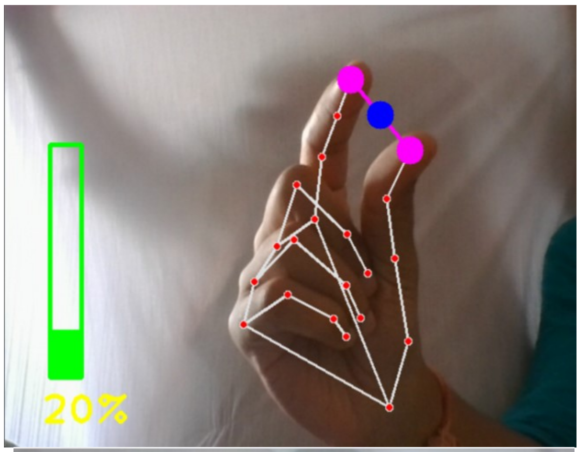
```
while True:
    success , img = cap.read()
    imgRGB = cv2.cvtColor(img , cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)
    if results.multi_hand_landmarks:
        for handLms in results.multi_hand_landmarks:
            lmList =[]
            for id , lm in enumerate(handLms.landmark):

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


                #if id == 1:
                #    cv2.circle(img ,(cx,cy) ,20, (255 , 0,125) ,cv2.FILLED )
                mpDraws.draw_landmarks(img , handLms , mpHands.HAND_CONNECTIONS)
                #print(results.multi_hand_landmarks)

            if lmList:
                x1 , y1 = lmList[4][1] , lmList[4][2]
                x2 , y2 = lmList[8][1] , lmList[8][2]
                cv2.circle(img , (x1 ,y1) ,15 , (255,0,255) ,cv2.FILLED)
                cv2.circle(img , (x2 ,y2) ,15 , (255,0,255) ,cv2.FILLED)
```

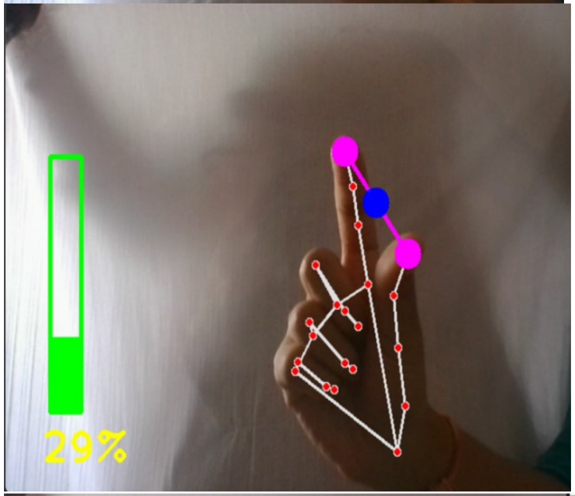


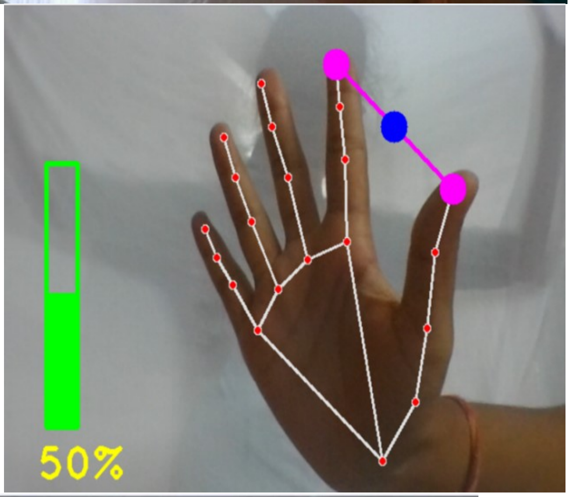
```
                cv2.line(img, (x1,y1) , (x2,y2) , (255 , 0, 255) ,3)

                z1 = (x1 + x2)//2
                z2 = (y1 + y2)//2

                cv2.circle(img , (z1,z2),15 ,(255,0,0) , cv2.FILLED )
                length = math.hypot(x2- x1 , y2- y1)
                #print(length)

                if length < 50:
                    cv2.circle(img , (z1 , z2) ,15 ,(255,255,255) , cv2.FILLED)
                if length > 300:
                    cv2.circle(img , (z1 , z2) ,15 ,(0,0,0) , cv2.FILLED)

            vol = np.interp(length , [50 ,300] ,[minVol ,maxVol])
            volBar = np.interp(length , [50,300] ,[400 ,150])
            volPer = np.interp(length , [50 , 300] , [0,100])
            #print(vol)
            print(int(length) , vol)
            #print(volume.GetVolumeRange())
            volume.SetMasterVolumeLevel(vol, None)

            cv2.rectangle(img ,(50,150) , (85,400) ,(0,255,0) , 3)
            cv2.rectangle(img ,(50,int(volBar)) , (85,400) ,(0,255,0) , cv2.FILLED)
            cv2.putText(img , f'{int(volPer)}%', (40,450) ,cv2.FONT_HERSHEY_PLAIN,3 ,(0,255,255) ,3)


    cv2.imshow("Image detection" , img)
    cv2.waitKey(1)
```

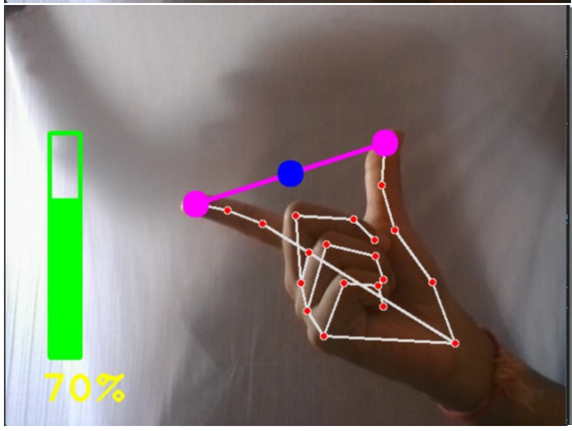
9.Output

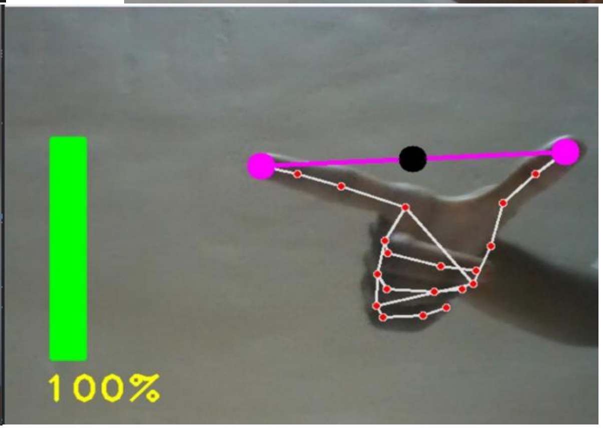












Results comparison with other approaches:

Deep Learning Based Approach:

In this approach convolution neural networks are used to control volume with hand gestures.This approach has the potential to achieve high accuracy and robustness by automatically learning complex patterns from data.

Drawbacks:

Data requirements: Deep learning models often require large labeled datasets for optimal performance. Acquiring such datasets can be challenging and time-consuming.

Computational resources: Training deep learning models can demand significant computational resources, including high-performance GPUs or specialized hardware.

Model complexity: Deep learning models can be complex and have a large number of parameters, making them harder to interpret and debug.

Machine Learning-Based Approach:

Machine learning models, such as decision trees, random forests, or support vector machines, can capture complex relationships between hand gestures and volume control.This approach involves training a model on a labeled dataset of hand gesture samples.

Drawbacks :

Data requirements: Machine learning models require a significant amount of labeled training data to learn accurately. Collecting and annotating such datasets can be time-consuming and expensive.

Training complexity: Training machine learning models may require substantial computational resources and expertise in selecting appropriate algorithms and hyperparameters.

Preprocessing and feature engineering: Extracting relevant features from raw input data and preprocessing it for training can be challenging and require domain knowledge.

10. Conclusion

This project is presenting a program that allows the user to perform hand gesture for convenient and easier way to control the software .A gesture based volume controller doesn’t require some specific type of physical devices and these can be operated in our real life on simple Personal Computers with a very low cost cameras as this not requires very high definition cameras to detect or record the hand gestures.Specifically, system tracks the tip positions of the counters and index finger of each hand.The main motive of this type of system is basically to automate the things in our system in order to make the things become easier to control. So in order to make it reliable we have used this system to make the system easier to control with the help of these applications.

11.References

1.Volume Control Using Gestures (https://ijisrt.com/assets/upload/files/IJISRT22MAY250_(1).pdf

2.GESTURE VOLUME CONTROL USING OPENCV

(https://www.irjmets.com/uploadedfiles/paper//issue_6_june_2022/27042/final/fin_irjmets1656354635.pdf)