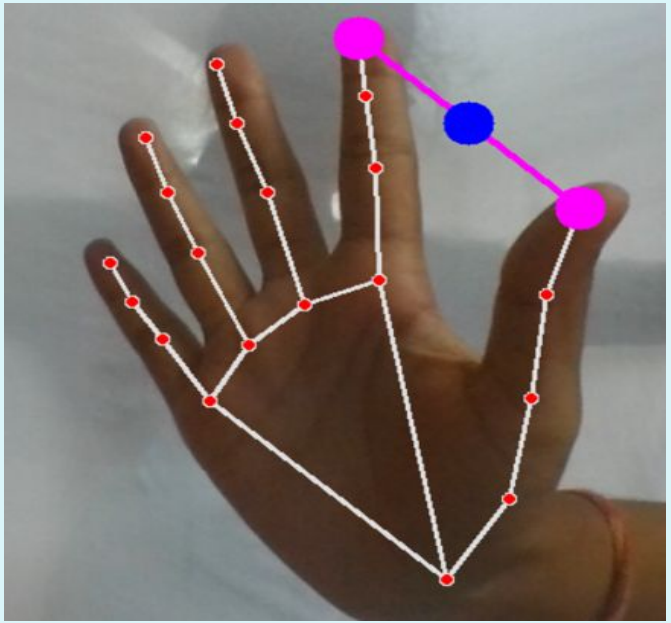
Volume Control using Gestures

Abstract

The purpose of Volume Control using Gestures project is to control volume of the system using hand gestures. The system consist 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.

Introduction

Volume Control using Gestures 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 objective of this project is to develop an interface which will capture human hand gesture dynamically and will control the volume level.This allows users to control the volume of their devices naturally, without the need for explicit instruction and without the need for additional hardware.Volume control using gestures uses OPENCV,Python,Pycaw,Mediapipe are used.



Motivation for the Work

- **Enhanced User Experience:** To provide users a natural and effortless way to adjust the volume of their devices, enhancing the overall user experience. By replacing traditional input methods like keyboard and mouse with hand gestures, the system aims to improve user satisfaction.

- **Seamless Integration:**Users can effortlessly control the volume without interruptions or distractions, enabling them to stay focused on their tasks, such as watching videos, listening to music, or participating in video conferences.

Motivation for the Work

- **Quick and Convenient Control:** Hand gesture recognition enables users to control the volume in a swift and convenient manner. Instead of searching for the volume buttons or sliders on a screen, users can make simple hand gestures in front of the camera to adjust the volume instantly. This can save time and effort, especially in situations where quick adjustments are required.

- **Personalization and Customization:** Gesture-based volume control allows users to personalize their interaction with devices. They can define their own gestures or gestures that are most comfortable and natural for them, enhancing the user experience and creating a sense of ownership over the control method.

Real-world Applications

Home Entertainment Systems: The system can be integrated with home entertainment systems, such as televisions, audio systems, or smart speakers. Users can control the volume of their devices using hand gestures, eliminating the need for remote controls or physical buttons.

Automotive Interfaces: Hand gesture recognition can be integrated into automotive interfaces to control various functions without the need for physical buttons or touchscreens. Drivers can adjust volume, answer calls, or navigate through infotainment systems using hand gestures, improving safety and minimizing distractions.

Gesture-based Presentations: Hand gestures can be utilized to control presentations or slideshows. Users can navigate through slides, zoom in or out, highlight content, or trigger specific actions by performing predefined gestures.

Research Gaps Identified

- **Robustness to varying environmental conditions:** One of the challenges mentioned is the impact of background images or videos, as well as lighting conditions, on the quality of hand gesture recognition. Further research could focus on developing algorithms or techniques that are more robust and less affected by variations in the environment, allowing accurate recognition in different settings.

- **Touch-Based Interaction:** Requires physical contact with a touch-sensitive surface, which may not be ideal in certain environments or situations.Smaller touch interfaces can be challenging to operate accurately, potentially leading to unintended volume adjustments.

Research Gaps Identified

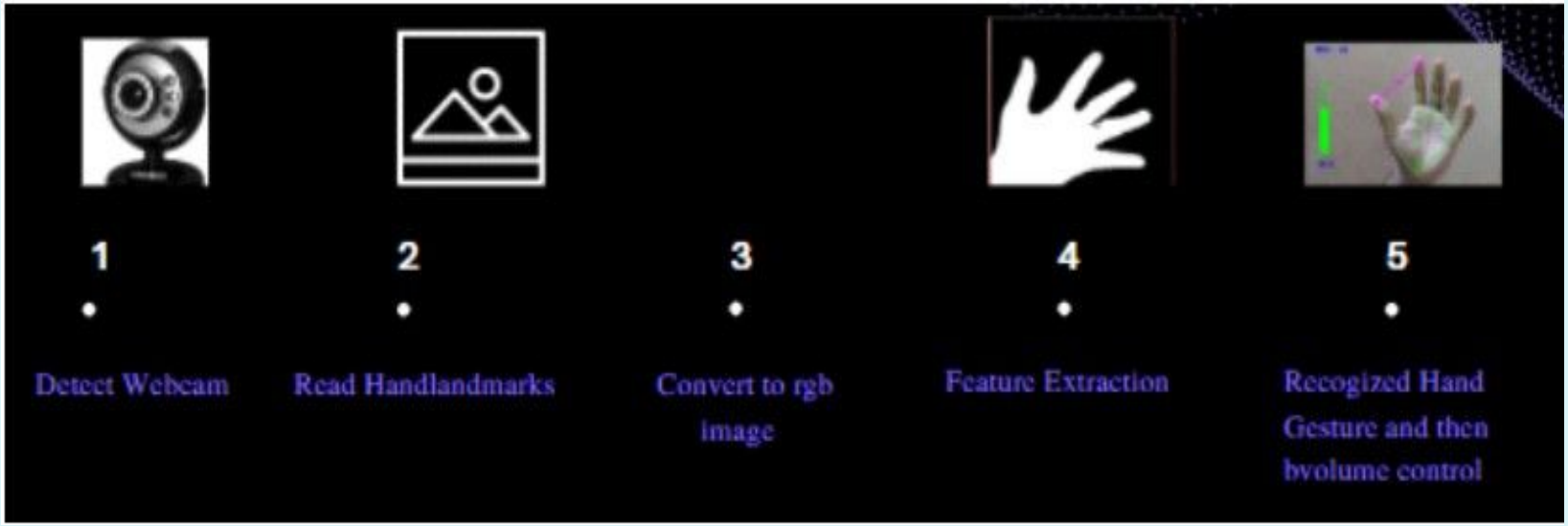
- **User Experience and Ergonomics:** Researching the user experience and ergonomic aspects of using hand gestures for volume control is another potential research gap. This could involve studying user preferences, comfort, and efficiency when performing hand gestures for audio control and identifying ways to optimize the interaction design for better user satisfaction

- **Social and collaborative gesture recognition:**Hand gestures are often used in social and collaborative contexts,such as group interactions or teamwork scenarios. Research gaps exist in developing gesture recognition systems that can effectively recognize and interpret gestures and can control volume in such social and collaborative settings, enabling seamless communication and collaboration between users.

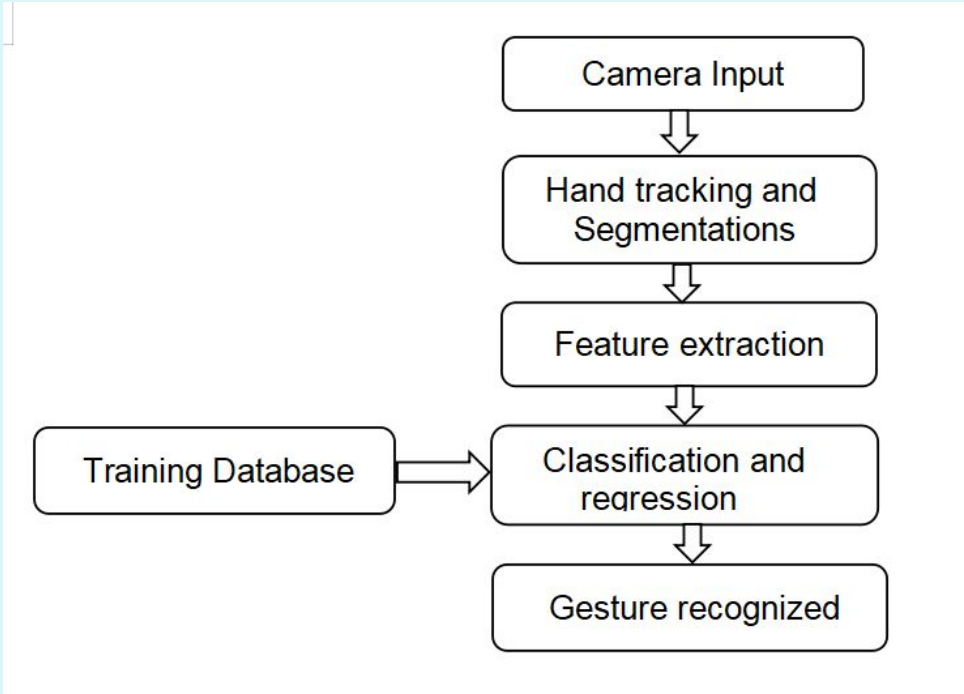
Proposed Method

- 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.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’.

Proposed method



Flowchart


Camera Input
Hand tracking and Segmentations
Feature extraction
Training Database
Classification and regression
Gesture recognized

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 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 complete the processing of the image.

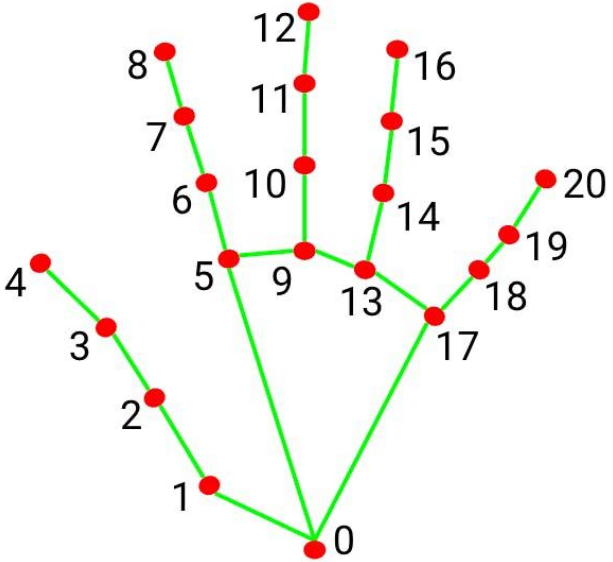
```
while True:
    success , img = cap.read()
    imgRGB = cv2.cvtColor(img , cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)
```

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
```

Hand tracking and segmentations



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

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.

Classification and regression

The classification tasks involves in 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)

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.

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])

Experimental setup

- Technologies/libraries used
- Datasets
- System Hardware
- Results table/screenshots/any
- Results comparison with other approaches

Technologies/libraries used

- Programming Language:
  - Python version-3.11.0
- Libraries:
  - OpenCV
  - numpy
  - Pycaw
  - Mediapipe
  - ctypes
  - comtypes
- Platform used:
  - Visual studio

Libraries

- **OpenCV:** Open source computer vision
  - To read a video capture/ Image capture from webcam and convert into RGB

- **Numpy:** Numerical python
  - Numpy is used to work with arrays and used for mathematical functions

- **Pycaw:** Python core audio window
  - It is used to access the device speakers
  - It allows to manipulate the audio settings such as volume control, mute and unmute

  Audio utilities: Interact with audio devices and their properties like retrieve information, activating audio end point volume interface

  IAudioEndpointVolume: Controlling the audio device volumes and represents the audio end point volume control for specific audio device to control volume level

Libraries

- **Media pipe:** This library is used to recognize the gestures.
  - To identify the hands and which is udes to draw landmarks and connections on the image

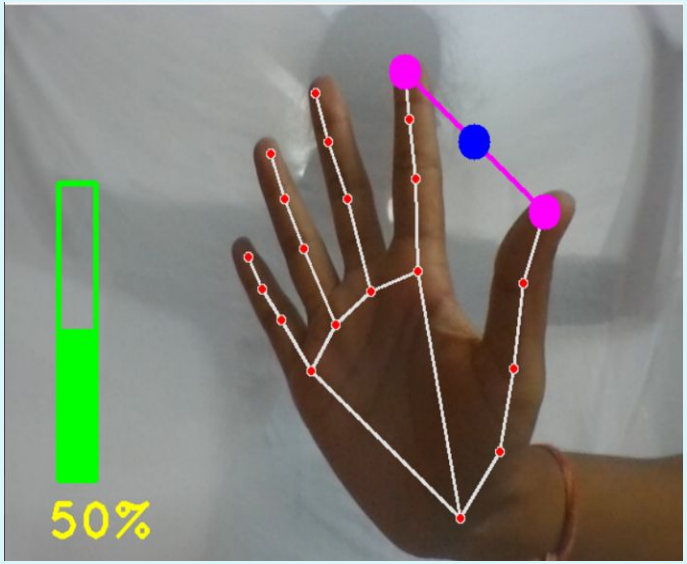
- **Ctypes:** Python library provides facilities for calling ‘c’ functions and working with ‘c’ data types.
  - **Cast:** casting is an construct to view a data object temporarily as another data type
  - **Pointer:** class is used to declare pointers to ‘c’ data types in python

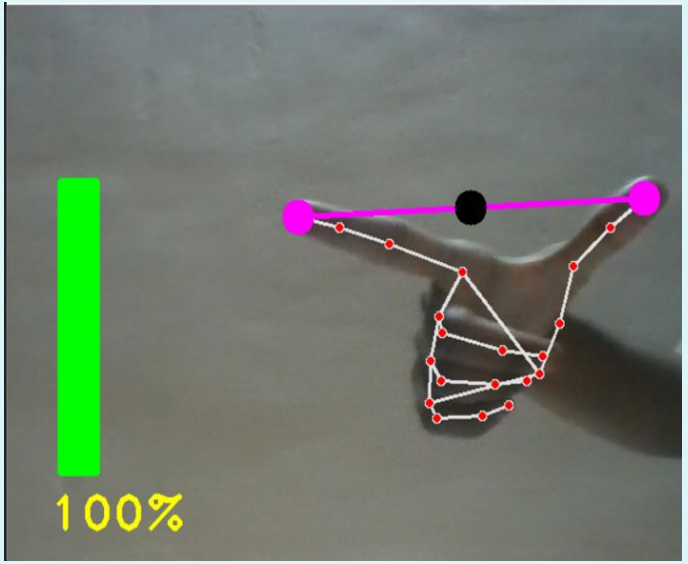
- **Comtypes:** This module is a python package that provides COM(component object model) . It allows to work with com objects and interfaces in python.
  - **CLSCTX_ALL:** It is used to activate the system audio

System Hardware

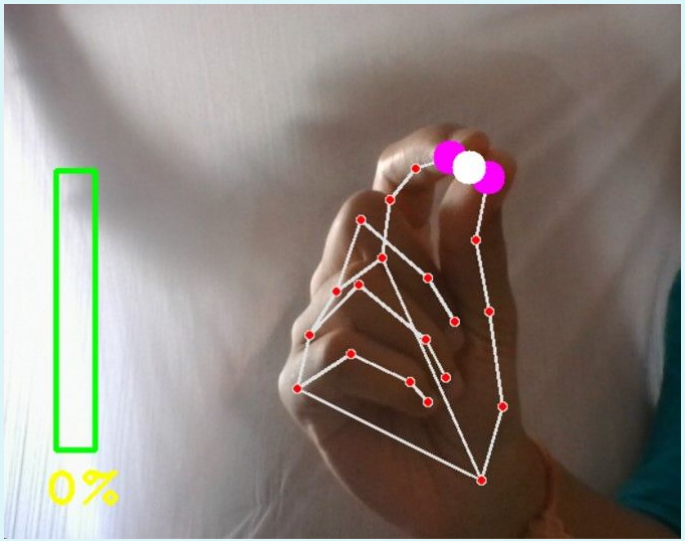
- **Processor:** Intel core i5
- **Memory(RAM):** 4GB
- **Disk:** 320(approx.)

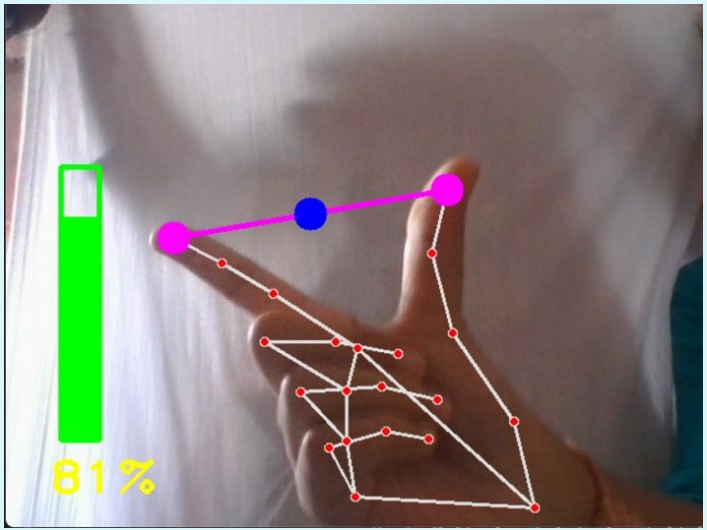
Results screenshots





Results screenshots





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 have 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.

Results comparison with other approaches

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.

Conclusion

The volume control using gestures project presents 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 application.

References

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

- GESTURE VOLUME CONTROL USING OPENCV
(https://www.irjmets.com/uploadedfiles/paper//issue_6_june_2022/27042/final/fin_irjmets1656354635.pdf)