Volume and Brightness Control by Hand Gestures Using Open CV And Media pipe

A project report submitted in partial fulfillment of the requirements for

the award of the degree of

Bachelor of Technology

by

P. Rohan	**:**	**2111CS020407**
A. Rohith Kumar	**:**	**2111CS020409**
K. Rupa Sri	**:**	**2111CS020411**
Rushi Eshwer Reddy Neelam	**:**	**2111CS020412**
D. Rushika	**:**	**2111CS020413**

Under the guidance of

Dr.G.Gifta Jerith

Assistant Professor

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

MALLA REDDY UNIVERSITY

(As per Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

HYDERABAD – 500043

TELANAGANA

INDIA

2023-24

MALLA REDDY UNIVERSITY

(As per Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

HYDERABAD – 500043

TELANAGANA

Certificate

This is to certify that this is the bonafide record of the application development entitled," Volume and Brightness Control by Hand Gestures Using Open CV and Mediapipe submitted by **P. Rohan(2111CS020407), A. Rohith Kumar(2111CS020409), K.Rupa Sri(2111CS020411), Rushi Eshwer Reddy Neelam(2111CS020412), D. Rushika(2111Cs020413)** of B.Tech lII year IInd semester,Department of CSE (AI&ML) during the year 2023- 24.The results embodied in thereport have not been submitted to any other university or institute for the awardof any degree or diploma

INTERNAL GUIDE
Dr.G.Gifta Jerith
Assistant Professor

HEAD OF THE DEPARTMENT
Dr. Thayaba Khatoon
Professor & HoD

ACKNOWLEDGEMENT

We would like to express our gratitude to all those who extended their support and suggestions tocome up with this application. Special Thanks to our mentor Dr. G. Gifta Jerith whose help and stimulating suggestions and encouragement helped us all time in the due course of project development.

We sincerely thank our HOD Dr. Thayyaba Khatoon for her constant support and motivation all the time. A special acknowledgement goes to a friend who enthused us from the back stage. Last but not the least our sincere appreciation goes to our family who has been tolerant understanding our moods, and extending timely support.

Abstract

The purpose of this project is to discuss a volume and brightness control using hand gesture recognition system based on detection of hand gestures. In this the system is consist of a high resolution camera to recognize 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 and by using real time gesture recognition specific user can control a computer by using hand gesture in front of a system video camera linked to a computer. In this project we are developing a hand gesture volume and brightness controller system with the help of OpenCV, Python. In this system can be controlled by hand gesture without making use of the keyboard and mouse.

CONTENTS

1.INTRODUCTION

1.1 Problem Definition

In today's digital age, the interaction between humans and computers has evolved significantly. While traditional input devices like keyboards and mice are widely used, there is a growing demand for more intuitive and hands-free control methods. The aim of this project is to develop a Hand Gesture Volume and Brightness Control System using computer vision techniques and machine learning algorithms.

1.2 Objective of project

The primary objective of this project is to design and implement a system that allows users to control the volume and brightness of a computer or device using hand gestures captured by a camera. The system should achieve the following goals:

- **Gesture Recognition:** Develop algorithms to accurately detect and recognize hand gestures in real-time video streams.

- **Volume Control:** Implement functionality to adjust the volume levels of a device based on recognized hand gestures, providing users with intuitive control over audio output.

- **Brightness Control:** Incorporate features to regulate screen brightness using hand gestures, enhancing user experience and comfort during computer usage.

- **Real-time Operation:** Ensure that the system operates in real-time, enabling instant feedback and responsive interaction with the user's gestures.

- **User-Friendly Interface:** Design a user-friendly interface that displays feedback on gesture recognition and provides visual cues for volume and brightness adjustments.

- **Compatibility:** Ensure compatibility with a wide range of devices and operating systems to maximize accessibility and usability.

1.3 Limitations of the project

- **Limited Gesture Vocabulary:** The system may have a limited vocabulary of recognized gestures, restricting the range of available commands for volume and brightness control. Users may need to learn specific gestures, which could potentially lead to usability issues.

- **Hardware Requirements**: The system requires a high-resolution camera capable of capturing detailed hand movements. Users without access to suitable hardware may be unable to utilize the system effectively.

- **Processing Resources:** Real-time gesture recognition and control require significant computational resources. Low-powered devices or older computers may struggle to handle the processing demands, leading to performance issues or delays in responsiveness.
- **Single User Interaction:** The system may be optimized for single-user interaction, meaning that it may not support simultaneous control by multiple users. This limitation could be problematic in shared or collaborative environments.

- **Environmental Factors:** External factors such as background noise, cluttered backgrounds, or occlusions may interfere with the accurate detection of hand gestures.

These environmental factors could impact the reliability and consistency of the system's performance.

2. LITERATURE SURVEY

1) Gesture Recognition Techniques:

Gesture recognition is a key component of volume control systems using hand gestures. Several approaches have been proposed in the literature, including template matching, machine learning-based classification, and deep learning methods. OpenCV, a popular computer vision library, provides tools for implementing gesture recognition algorithms[1] efficiently. Researchers

have explored different techniques within OpenCV for detecting and recognizing hand gestures, such as contour analysis, histogram-based methods, and feature extraction.

2) MediaPipe Framework:

MediaPipe, developed by Google, is an open-source framework for building real-time multimedia processing pipelines. It offers pre-trained models and efficient implementations for various tasks, including hand tracking and pose estimation. MediaPipe's Hand Tracking module[6] provides robust hand detection and tracking capabilities, which are essential for capturing hand gestures accurately in volume control systems. Integrating MediaPipe with OpenCV enables developers to leverage its advanced features for hand gesture recognition.

3) Volume Control Applications:

Several research studies have investigated the use of hand gestures for controlling volume in multimedia applications. These studies often employ different sensor modalities, such as depth cameras, wearable devices, and vision-based systems. Vision-based approaches[2] offer advantages in terms of cost-effectiveness and ease of deployment, making them suitable for consumer electronics and smart devices. By combining OpenCV and MediaPipe, researchers can develop volume control systems that are responsive, accurate, and adaptable to various environments and user preferences.

4) Image Processing with OpenCV:

OpenCV (Open Source Computer Vision Library) is a widely-used tool for image processing, offering various functions and modules for tasks like filtering, contour detection, and feature extraction. In the context of hand gesture recognition, OpenCV provides essential tools for preprocessing images[4] and extracting relevant features from hand images. Researchers and developers often rely on OpenCV due to its versatility and effectiveness in handling various image processing tasks. However, optimizing OpenCV for real-time applications requires careful consideration of computational efficiency and performance tuning techniques.

5) Volume Control Methods and Audio Signal Processing:

Controlling volume in software applications typically involves adjusting parameters related to audio signal processing, such as amplitude and gain. While traditional methods include sliders, buttons, and keyboard shortcuts, recent research has explored alternative interfaces, including gesture-based approaches and voice commands. Integrating hand gestures or motion control into audio applications presents[4] exciting possibilities for enhancing user experiences and accessibility. Researchers have investigated different techniques for incorporating gesture recognition into audio systems, exploring the potential for intuitive and hands-free interaction.

3. PROPOSED METHODOLOGY

3.1. Existing System

Functionality: The existing system allows users to control system volume using hand gestures captured by a webcam.

Components:

- HCI model
- Raspberry pi
- Computer Vision
- Hand Tracking model
- Control Mechanisms
- Graphical User Interface (GUI)

Features: Real-time hand gesture detection and control. Automatic adjustment of volume and brightness based on hand gestures. Feedback messages for hand detection status.

3.2. Proposed System

Functionality: Expands the existing system to allow users to control both system volume and screen brightness using hand gestures captured by a webcam.

Enhancements:

- Brightness Control
- Low quality Camera supporting
- Integration with existing models

Features: Potentially includes additional features such as user profiles, gesture training, and multi-modal control to improve usability and customization options.

3.3 Modules

OPEN CV: Open CV is a library of python which tackle PC vision issue. It is used to detect the face which is done using the machine learning .It is a very important library and is used in several projects to detect the face and recognize the several frames also it supports several programming languages. It also performs object detection and motion detection. It

also support several type of operating system and can be used to detect the face of the animals also.

NUMPY; NumPy is the module of the Python. The numpy word basically shows Numerical Python and it is utilized. This is the module which is basically written in c language and is said as expansion module . Numpy guarantee remarkable execution speed. Numpy is mostly used for performing calculations, tasks using certain functions it provides like multiply, divide, power etc.

IMAGE FILTERING -HISTOGRAM: Histogram is a type of graph which represents the movement of the pixels power in the portrayal.In this we use to filter the images using histogram and convert them into the rgb in order to process the image in our system . Consequently the power of a pixel is in the range [0,255].

- **MEDIAPIPE**: MediaPipe is a module for processing video, audio and several types of related data across platform like Android, iOS, web, edge device and several applied ML pipeline. Several types of functions are performed with the help of this module , we have used this module in our project to recognize the hand gesture and detect the input from it. • Face Detection • Multi-hand Tracking • Segmentation • Object Detection and Tracking.

3.4. Architecture

1. **Python Language**: Python is chosen as the programming language for developing the project due to its simplicity, versatility, and extensive support for libraries and frameworks. Python's ease of use and readability make it well-suited for rapid prototyping and development of computer vision and machine learning applications.

2. **OpenCV and NumPy Modules**: OpenCV (Open Source Computer Vision Library) is a powerful open-source library designed for computer vision and image processing tasks. It provides a wide range of functionalities for image and video manipulation, including object detection, feature extraction, and gesture recognition. NumPy is a fundamental package for numerical computing in Python, providing support for multi-dimensional arrays, mathematical functions, and linear algebra operations. In the project, OpenCV is used for video input processing, while NumPy is utilized for numerical computations and data manipulation tasks.

3. **Camera Input**: The project utilizes the main camera as the video input source. OpenCV's VideoCapture module is used to access and capture video frames from the camera in real-time.

4. **Gesture Recognition Module**: A gesture recognition module is employed to detect and recognize hand gestures from the video input. This module likely utilizes computer vision techniques such as hand tracking, landmark detection, and gesture classification. The specific implementation may involve machine learning models trained on annotated gesture datasets.

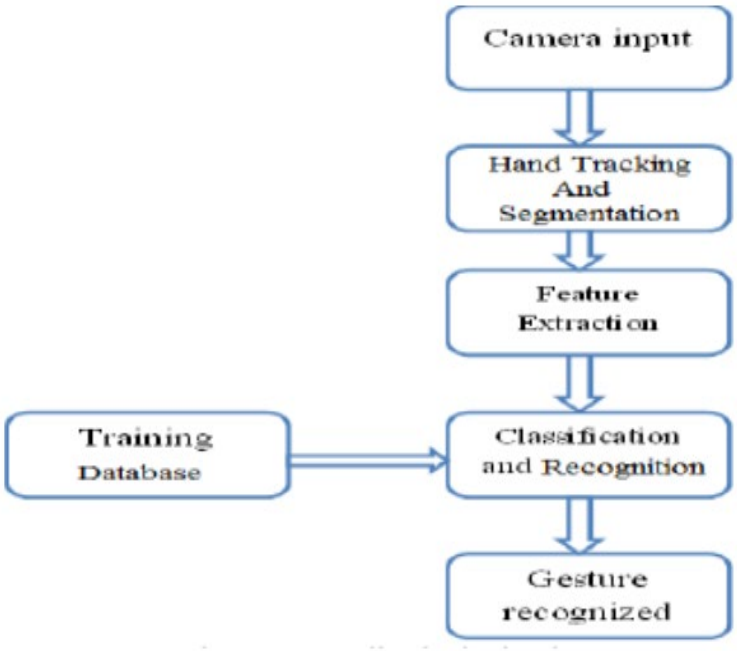
5. **PyCAW for Volume Control**: PyCAW is a Python library that provides access to the Windows Core Audio API, allowing for volume control of audio devices such as speakers and headphones. In the project, PyCAW is used to adjust the system volume based on the recognized hand gestures.
6. **Volume Range Adjustment**: The volume range is specified from the lowest volume to the highest volume level. This range is used to map the detected hand gestures to corresponding volume levels. PyCAW is then utilized to set the master volume level of the audio device within this specified range.

7. **RGB Image Conversion**: The input image from the camera is converted to an RGB image format for further processing. This conversion ensures compatibility with the color

representation expected by the gesture recognition module and other image processing tasks.

8. **Thumb and Finger Points Detection**: Specific thumb and finger points in the input image are detected and extracted as landmarks for gesture recognition. These points are crucial for determining hand poses and gestures, enabling precise control over volume and brightness adjustments.

9. **NumPy Library**: NumPy is indispensable for its powerful N-dimensional array operations, which are extensively used for numerical computations and data manipulation in the project. It provides efficient handling of large datasets and mathematical functions essential for processing hand gesture data and calculating volume and brightness adjustments.



3.5. Methods & Algorithms

- In the Hand Gesture Volume and Brightness Control System, various methods and algorithms are utilized for different tasks, including gesture recognition, hand tracking, volume control, and brightness adjustment. Let's explore the key methods and algorithms employed in each aspect:

- **Gesture Recognition:**
  Hand Tracking: Hand tracking algorithms are used to detect and locate the user's hand in the video feed. Techniques such as background subtraction, skin color detection, or deep learning-based approaches may be employed for hand localization.

- **Landmark Detection:**

Once the hand is tracked, landmark detection algorithms identify key points or landmarks on the hand, such as the fingertips, thumb, and palm. Techniques like convolutional neural networks (CNNs) or cascade classifiers may be used for landmark detection.

- **Gesture Classification:**
  After detecting landmarks, gesture classification algorithms classify the hand pose or gesture based on the positions of the detected landmarks. Machine learning models such as support vector machines (SVMs), decision trees, or deep neural networks (DNNs) are commonly used for gesture classification.

- **Hand Tracking and Pose Estimation:**
  OpenCV and MediaPipe: Libraries like OpenCV and MediaPipe provide pre-trained models and APIs for hand tracking and pose estimation. These libraries offer efficient implementations of hand tracking algorithms, making it easier to detect and track hands in real-time video streams.

- **Convolutional Neural Networks (CNNs):**
  CNN-based models trained on annotated hand pose datasets are used for accurate hand tracking and pose estimation. These models leverage convolutional layers to learn spatial hierarchies of features from input images, enabling robust hand tracking and pose estimation.

- **Volume Control:**
  PyCAW Library: The PyCAW library is utilized to control the system volume based on recognized hand gestures. PyCAW provides access to the Windows Core Audio API, allowing for programmatic volume adjustment of audio devices.

- **Mapping Gestures to Volume Levels:**

The detected hand gestures are mapped to specific volume levels within a predefined range. Linear interpolation or mapping functions may be used to translate gesture positions to corresponding volume levels.

- **Brightness Adjustment:**
    - **Screen Brightness Control:** Screen brightness control algorithms adjust the brightness of the display based on recognized hand gestures. These algorithms typically interface with the operating system's display settings to dynamically adjust the screen brightness.
    - **Brightness Mapping:** Similar to volume control, hand gestures are mapped to specific brightness levels within a predefined range. Linear interpolation or mapping functions are used to convert gesture positions to corresponding brightness levels.

- **Numerical Computation and Data Processing:**

  NumPy Library: The NumPy library is extensively used for numerical computations and data processing tasks. It provides efficient array operations and mathematical functions for processing hand gesture data, calculating volume and brightness adjustments, and performing other numerical tasks required by the system.