# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

### “Jnana Sangama”, Belagavi-560 014, Karnataka



**A Mini Project Report On**

**“GESTURE VOLUME CONTROLLER”**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE COMPUTER GRAPHICS AND IMAGE PROCESSING LABORATORY [21CSL66] OF**

## BACHELOR OF ENGINEERING

**IN**

## Department of Computer Science and Engineering

### Submitted By

1. **PANEENDRA E[1SV21CS053]**

### RAJAN KUMAR SAH [1SV21CS060]

**Under the Guidance of**

**Prof. Shanmukaswamy C.V. B.E.,M.E.,MISTE.**

### Associate Professor, Dept of CSE, SIET, Tumakuru

**and**

**Mrs. Rashmi N B.E., M.Tech.**

### Assistant Professor, Dept of CSE, SIET, Tumakuru



**Department of Computer Science and Engineering**

**SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Affiliated to Visvesvaraya Technological University) Sira Road, Tumakuru–572 106, Karnataka.**

**2023-2024**



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the mini-project entitled “**GESTURE VOLUME CONTROLLER**” has been successfully carried out by 1.PANEENDRA E[1SV21CS053], 2.RAJAN KUMAR SAH[1SV21CS036] in partial fulfillment for the **Computer Graphics and Image Processing Laboratory [21CSL66]** Mini Project of **Bachelor of Engineering in** the **Department of Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the Academic year **2023-24.** It is certified that all the corrections/suggestions indicated for internal assessments have been incorporated into the report. The Mini Project Report has been approved as it satisfies the academic requirements in respect of the Subject of the Bachelor of Engineering Degree.

### Signature of the Guide Signature of the Guide

**Mrs. Rashmi N B. E, M. Tech, Prof. Shanmukaswamy C V B. E, M. E, MISTE**

Assistant Professor, Dept of CSE, Associate Professor, Dept of CSE, SIET, Tumakuru SIET, Tumakuru

### ----------------------------------------------

**Signature of the H.O.D**

**Dr. Basavesha D B.E., M.Tech., Ph. D., MISTE.**

Associate Professor and HOD, Dept of CSE, SIET, Tumakuru.

### Name of the Examiners Signature with Date

**1.**

**2.**



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**DECLARATION**

We **PANEENDRA E** **[1SV21CS053] and RAJAN KUMAR SAH [1SV21CS060]**, students of VIsemester **B.E** in Computer Science & Engineering , at Shridevi Institute of Engineering & Technology, Tumakuru, hereby declare that, the Mini Project work entitled **“GESTURE VOLUME CONTROLLER”**, embodies the report of our Mini-Project work carried out under the guidance of Prof. Shanmukaswamy C. V., Associate Professor, Dept. of CSE, and Mrs. Rashmi N, Assistant Professor, Department of CSE, SIET as partial fulfillment of requirements for the **Computer Graphics and Image Processing Laboratory [21CSL66]** mini project of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi, during the academic year **2023-24**. The Mini Project has been approved as it satisfies the academic requirements in respect to the Mini Project work.

Place: Tumakuru Students’ Name and Signature

### Date: **PANEENDRA E** [1SV21CS053]

**RAJAN KUMAR SAH [1SV21CS060]**

**ACKNOWLEDGEMENT**

This Mini Project would be incomplete without thanking the personalities responsible for this venture, which otherwise would not have become a reality.

We would like to thank our guides **Prof. Shanmukaswamy C. V., Associate Professor** and **Mrs. Rashmi N, Assistant Professor,** Computer Science & Engineering, SIET for their support, sharing their technical expertise timely advice invaluable guidance and assistance throughout this mini-project.

We would like to thank the Head of the Department **Dr. Basavesha D, Associate Professor and Head, Department of Computer Science and Engineering, SIET** for providing all the support and facility.

We express our profound gratitude to **Dr. Narendra Viswanath, Principal**, **SIET**, for his moral support towards completing our mini-project work.

We would like to express our sincere gratitude to all teaching and non-teaching faculty of the Department of CSE for guiding us in this mini-project by giving valuable suggestions and encouragement.

**PANEENDRA E [1SV21CS035]**

**RAJAN KUMAR SAH [1SV21CS036]**

# ABSTRACT

A systems that can recognize a hand motion in real time video is hand gesture recognition. Hand gestures are categorized according to their subject matter. The design of hand gesture recognition is one of the more difficult jobs, as it combines two significant issues. The detection of the hand is the first step and creating a sign that can only be utilized by one hand at a time. It can be used in a variety of settings, including human-computer interaction and sign language. The basic concept of hand segmentation and the hand detection system, which use the Haar-cascade classifier, may be used to construct hand gesture recognition using Python and OpenCV. This paper discuss a way for hand gesture identification based on shape-based features detection. The configuration comprises a single camera that captures the user's gesture and feeds it into the system. A fundamental goal of gesture recognition is to develop a system that can recognize specific human gestures and utilize them to send information for device control. With real-time gesture recognition, a user can operate a computer by making a specific gesture in front of a computer's camera. With the help of the OpenCV module, we will create a hand gesture. Without the use of a keyboard or mouse, the system can be controlled via hand gestures.

**Keywords:** Human Computer Interaction, Structuring Elements, Hand gesture, Region of Interest

|  |  |
| --- | --- |
| **CONTENTS**  **1. INTRODUCTION** | **Page No.**  **1** |
| **2. LITERATURE SURVEY** | **2** |
| **3. SYSTEM ANALYSIS** | **3** |
| 3.1. Existing System | 3 |
| 3.2. Proposed System | 3 |
| **4.SYSTEM METHODOLOGY** | **5** |
|  |  |
| 4.1 System Methodology | 6 |
| 4.2 Implementation | 8 |
| **5.RESULTS** | **11** |
| **6.CONCLUSION** | **13** |
| **7.FUTURE ENHANCEMENT** | **14** |
| **8.REFERENCES** | **15** |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Figure Name** | **Page No.** |
| Figure | 3.2 | Proposed architecture of the system | 6 |
| Figure | 4.1 | Flowchart of the System | 7 |
| Figure | 5.1 | Blood Cancer Detected Benign | 11 |
| Figure | 5.2 | Blood Cancer Detected as Malignant Pre-B | 11 |
| Figure | 5.3 | Blood Cancer Detected as Malignant Pro-B | 12 |
| Figure | 5.4 | Blood Cancer Detected as Malignant early Pre-B | 12 |

# CHAPTER 1

## INTRODUCTION

Hand gesture recognition is a system that can recognize a hand motion in a real-time video in everyday life. Hand gestures are classified according to their subject of interest. One of the goal in this implementation is to design a system for hand gesture recognition. One of the most critical issues in computer vision is the task of recognizing hand motions. Human computer interaction (HCI) systems that entail hand processing tasks such as hand detection and hand gesture recognition have become more advanced as information and media technology has progressed.

Detecting and locating the hand in real-time footage from the webcam is the initial stage in any hand processing system. Because of the variety in position, orientation, placement and scale, detecting a hand might be difficult. Variability is also aided by the varying levels of light in the room. Hand gesture recognition often requires numerous levels of processing, including image acquisition, pre-processing, feature extraction, and gesture identification. Using a webcam, image acquisition entails recording an image in a video frame by frame. The collected images are subjected to colour filtering, smoothing as part of the image pre-processing.

Feature extraction is a technique for extracting features from a hand image, such as hand outlines, whereas gesture recognition is a technique for extracting features and recognizing hand gestures. Designing a hand gesture recognition system is a difficult task that comprises two major issues. The first is the detection of a person's hand.

A webcam is used to detect the user's hand in real-time video in this acquisition some issue arises such as inconsistencies in brightness, noise, resolution, and contrast. To identify the gestures, the technique involves segmentation and edge detection and with help of the openCV module we obtain the hand gesture and able to control the volume

# CHAPTER 2

## LITERATURE SURVEY

Hand gesture detection and utilizing them to control certain set of devices operations and allowing interaction with computer system without the aid of mouse and keyboard. In this paper we draws along the same line but we attributed the use of Haar-cascade classifier to identify hand gesture. Some of the related works in this field are described briefly as follows

1. A non-local algorithm for hand gestures was proposed by A. Buades, B. Coll, and J. Morel. At the moment, finding finger movement algorithms remains a valid task. Functional analysis and statistics collide. Despite the fact that most recently presented approaches have a high level of sophistication, Algorithms have not yet reached a satisfactory degree of performance applicability. All work admirably when the model matches the algorithm assumptions, but they all fail in general, producing defects in analysing the pixels through the camera. The primary goal of this study is to define a generic mathematical and experimental technique for comparing and classifying conventional hand movement recognition algorithms.
2. For the no required elements in the video frame, Golam Moktader Daiyan et al. (2012) suggested a high performance decision based median filter. This technique detects noise pixels iteratively over numerous phases before replacing them with the median value. Noise detection is accomplished by enlarging the field of view. Mask till 77% to keep the extraction of local data going. Furthermore, if the algorithm fails to find a noise-free pixel at 7 7, the processing pixel is replaced by the last processed pixel. If the noise-free median value isn't available in the 7th processing window, the last processed pixel is used to determine if it is noise-free. The method chooses a window size if the last processed pixel is noisy. Calculate the number of 0s and 255s in the processing window using the 1515 dimension. Then, in the selected window, replace the processed pixel with 0 or 255, whichever is higher in number.

[3]Rajeshwari Kumar Dewangan et.al accurate object information and obtain a location using a deep learning object recognition technique.Object recognition algorithms are designed based on theSingle Shot MultiBox Detector (SSD) structure, an object recognition deep learning model, to detect objects using a camera.

1. H. Jabnoun et, al suggested the system that restores a central function of the visual system which is the identification of surrounding objects which is based on the local features extraction concept. Using SFIT algorithm and keypoints matching showed good accuracy for detecting objects.
2. Košale U, Žnidaršic P, Stopar K suggested that Detection of obstacles is performed by Time of Flight (ToF) sensors, whose ranging data is then processed with an on-board microcontroller and send via Bluetooth connection to the belt. The belt is equipped with a second microcontroller, which interprets the data and presents it to the wearer with 15 vibration motors arranged in a square grid. The glasses are worn on the head, whereas the belt is attached around the stomach area.

But the number of sensors detecting the obstacle decreased with the distance. Circle and square were detected better than triangle. This suggests that different shapes trigger different responses of sensors on glasses.

A. Jaiswal et al. [6] proposed an approach that used user generated picture denoising. The remaining task is broken down into four steps. The first image is denoised using a filtering process, and the second image is denoised using a different method. Wavelet-based approaches are used to denoise pictures, filtering, third hard thresholding, and thresholding. Finally, the approach was applied to a noisy image concurrently. PSNR output results are calculated by comparing all cases, the MSE (mean square error) is obtained. On the basis of PSNR, MSE, and image visual quality, experiments are conducted on 512 X 512 noisy images with noise variance of 0.04, output of median filter, Wiener filter, hard thresholding, and hard thresholding with median filtering. When the filtering and wavelet thresholding techniques are combined, they produce excellent results.

# CHAPTER 3

## SYSTEM ANALYSIS

**3.1 Existing system**

**3.1.1. Leap Motion**

* **Overview**: Leap Motion is a hardware sensor device that tracks hand and finger movements. It's often used for gesture control applications, including volume control.
* **Features**: Accurate hand and finger tracking, supports a variety of gestures, integrates with PCs and VR/AR headsets.
* **Use Case**: Users can raise or lower their hand to increase or decrease volume, or use specific hand gestures to mute or unmute.

**3.1.2. Microsoft Kinect**

* **Overview**: Kinect is a motion-sensing device originally developed for Xbox gaming consoles. It uses cameras and infrared sensors to detect body movements and gestures.
* **Features**: Full-body tracking, voice recognition, depth sensing.
* **Use Case**: Users can control the volume by moving their hands up or down in front of the Kinect sensor.

**3.1.3. Hand Gesture Recognition Systems (Generic)**

* **Overview**: Various systems and projects, both commercial and open-source, utilize cameras (e.g., webcam) and machine learning algorithms to recognize hand gestures.
* **Features**: Customizable gesture recognition, integration with different platforms, often built using OpenCV and machine learning libraries.
* **Use Case**: Customizable gestures like swiping up/down to adjust volume, clenching fist to mute.

**3.1.4. Smart Home Devices with Gesture Control**

* **Overview**: Devices like smart TVs or home assistants (e.g., some models of Samsung Smart TVs) come with built-in gesture control capabilities.
* **Features**: Built-in sensors/cameras, integration with smart home ecosystems.
* **Use Case**: Waving hand up/down to control volume, using pre-defined gestures to interact with the device.

**3.1.5. Mobile Applications**

* **Overview**: Apps available on smartphones that utilize the phone’s camera to detect hand gestures.
* **Features**: Use of front or rear camera, machine learning-based gesture recognition.
* **Use Case**: Users can control media volume on their smartphone by performing gestures in front of the camera.

**Example Implementation**

If you're interested in building a simple gesture-based volume controller yourself, here’s a basic outline of how it might work:

**Hardware:**

* A webcam or a Leap Motion device to capture gestures.

**Software:**

* **OpenCV** for video capture and image processing.
* **Machine Learning Models** for gesture recognition (could use pre-trained models or train your own using libraries like TensorFlow or PyTorch).
* **Integration** with the operating system's audio controls to adjust volume.

**Process:**

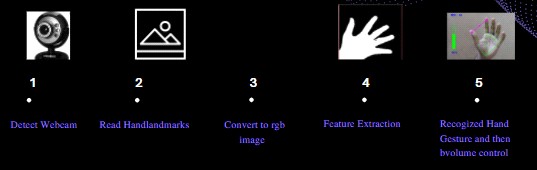
1. **Capture**: Use the camera to capture live video.
2. **Process**: Process the video frames to detect hand gestures.
3. **Recognize**: Use a machine learning model to classify the gesture.
4. **Control**: Map the recognized gestures to volume control commands (e.g., increasing, decreasing, muting).
   1.  **Proposed systems**

fig. 3.2. Proposed architecture of the system

As shown in figure 3.1, which indicates the proposed architecture of the system used in the volume controller. Here our input is hand gesture which is captured using web camera. Then the GUI (graphical user interface) helps to display the hand that conveys information and it processes the actions of the hand gestures of the user. By recognizing the gestures, the user is able to control the volume of the system which is final output.

# CHAPTER 4

## SYSTEM METHODOLOGY

## 

fig. 4.1. Flow chart of the system

The fig 4.1 indicates the flow chart of operation. Firstly, initiate start the program and then import various modules used for the AI recognition and various audio utilities which are of main concern. Next, it capture the area of interest by detecting various contours. Later it execute the loop to detect various hand landmarks. After getting the hand land marks, it verify the distance between the thumb and index figure tip. The frame is displayed giving the final values of the reading with complete decrease and increase in the volume using AI. The program is executed till the loop is iterated. Once it completes the iterations it comes out of the loop and the program stops.

**4.1 Algorithm**

Step 1: Start the Program

Step 2: Importing the Various Modules-Open Vision used for the AI recognition and various audio utilities which are of main concern. Step 3: Capturing the Area of Interest by detecting the various contours and differentiating the white and balck region of the interest Step 4: Executing the Loop to detect the various hand landmarks

Step 5: Getting the Hand Landmarks and verifying the distance between the index and thumb finger based on the algorithm given .

**4.2 Software Implementation**

Its implemented using the IDE software interpreter Pycharm it can also be implemented using the command prompt as well.

Import the open CV Library to python project which is used as a computer vision tool and to read the image which is nothing but hand in this context. Then we have to use MediaPipe which is a cross-platform framework for building multimodal applied machine learning pipelines. It is used for detection purpose. hypot() method returns the Euclidean norm. The Euclidean norm is the distance from the origin to the coordinates given.

Then to Get default audio device using PyCAW we have used comtypes which is a basic library and audio utilities. The video capture object to capture the information if the video cam is open. Then Media Pipe Hands is a high-fidelity hand and finger tracking solution. If the hands are detected then we have drawn the following outline of the hand using the audio utility function.

Obtain the default audio device using PyCAW.After which we have interfaced to the required volume and then found the range which is from 0 to 100, read the frames from the webcam and convert the image to RGB.

If Fit List of lm or glm Objects with a Common Model is null then we detect hands in the frame with the help of “hands.process()” function. Once the hands get detected we will locate the key points and then we highlight the dots in the key points using cv2.circle, and connect the key points using mpDraw.draw\_landmarks. The tip of the index and middle finger then we print(x1, y1, x2, y2).Then we check which fingers are up and we print the fingers. Convert Coordinates and then smoothen the values.

Both Index and middle fingers are close then we reduce the volume and if the index and middle finger are away then we increase the volume. We then find the length of the line through the coordinates. Map the distance of thumb tip and index finger tip with volume range. For our case, the distance between thumb tip and index finger tip was within the range of 15 – 220 and the volume range was from -63.5 – 0.0. Suppose in this case we have taken -8.0 to 194.83366 as maximum volume and similarly the reading for various volumes is taken.

# CHAPTER 5

1. **Results**

# CHAPTER 6

## CONCLUSION

In this paper we have taken up is a vision-based hand Gesture system that runs on a code based on Open CV library of python. It makes used of various algorithms and methods such as tracing significant points in the images and distance calculation between points.Specifically, the system can track the tip positions of the counters and index finger for each hand. It is an efficient and simple way to handle sound devices without much manual work. It does not require any special markers or gloves and can operate in real-time on a commodity PC with low-cost cameras.

The development and implementation of gesture volume controllers demonstrate several key benefits:

1. ****Enhanced User Experience**:** The intuitive nature of gesture controls can lead to a more seamless and enjoyable user experience. Users can adjust volume without direct contact with the device, making it especially useful in situations where hands-free operation is beneficial.
2. ****Accessibility**:** Gesture volume controllers can provide improved accessibility for individuals with physical disabilities or those who find traditional interfaces challenging to use.
3. ****Hygiene and Safety**:** By minimizing the need for physical contact, gesture controls can reduce the risk of spreading germs, which is particularly relevant in public or shared environments.
4. ****Innovation in Technology**:** The development of gesture volume controllers pushes the boundaries of current technology, encouraging further research and innovation in the fields of computer vision, machine learning, and human-computer interaction.

# CHAPTER 7

## FUTURE ENCHANCEMENT

The gesture volume controller, while already a significant innovation, has the potential for numerous enhancements that can further improve its functionality, user experience, and applicability. Some of the possible future enhancements include:

****1.Improved Accuracy and Responsiveness**:**

1. ****Advanced Algorithms**:** Implementing more sophisticated machine learning algorithms and incorporating deep learning techniques can improve the accuracy of gesture recognition, making the system more reliable.
2. ****Real-Time Processing**:** Enhancing the real-time processing capabilities can reduce latency, ensuring that gestures are recognized and responded to instantaneously.

****2. Enhanced Environmental Adaptability**:**

1. ****Robust to Lighting Conditions**:** Developing algorithms that can adapt to various lighting conditions, including low light and backlighting, will make the gesture volume controller more versatile.
2. ****Noise Reduction**:** Improving the system's ability to filter out background noise and accidental movements can enhance its reliability in different environments.

****3.Customization and Personalization**:**

1. ****User-Specific Gestures**:** Allowing users to customize and personalize gestures according to their preferences can make the system more user-friendly and adaptable to individual needs.
2. ****Profile-Based Adjustments**:** Implementing user profiles that remember specific settings and preferences can provide a tailored user experience.

****4.Multi-Device Integration**:**

1. ****Cross-Platform Compatibility**:** Developing the system to be compatible with various devices and operating systems, including smartphones, tablets, TVs, and smart home devices, can broaden its usability.
2. ****Seamless Switching**:** Enabling seamless switching and synchronization between different devices can enhance the user experience in a multi-device environment.

****5.Gesture Feedback Mechanisms**:**

1. ****Haptic Feedback**:** Incorporating haptic feedback to provide tactile responses to gestures can improve user interaction and confirmation.
2. ****Visual and Auditory Feedback**:** Implementing visual and auditory cues to indicate successful gesture recognition can enhance user confidence and usability.

****6.Integration with Emerging Technologies**:**

1. ****Augmented Reality (AR) and Virtual Reality (VR)**:** Integrating gesture volume controllers with AR and VR environments can create more immersive and intuitive user experiences.
2. ****Wearable Devices**:** Expanding compatibility with wearable devices, such as smartwatches and AR glasses, can extend the functionality and convenience of gesture controls.

****7.Energy Efficiency**:**

1. ****Low-Power Consumption**:** Developing more energy-efficient algorithms and hardware can extend battery life for portable devices using gesture volume controllers.
2. ****Optimized Resource Usage**:** Ensuring that the system operates efficiently without significantly impacting the performance of the host device.

# CHAPTER 8

## REFERENCES

1. Coll, Bartomeu, Antoni Buades & Morel, Jean-Michel. (2005). A Review of Image Denoising Algorithms, with a New One. SIAM Journal on

Multiscale Modeling and Simulation. 4. 10.1137/040616024.

2. Daiyan, Golam & Mottalib, M.A.. (2012). High performance decision based median filter for salt and pepper noise removal in images. 107-112.

10.1109/ICCITechn.2012.6509707.

3. Rajeshwari Kumar Dewangan, Siddharth Chaubey, “Object Detection System with Voice Output using Python,” 2021 International Journal of

Science & Engineering Development Research, Vol. 6, Issue 3, pp. 15-20, 2021.

4. H. Jabnoun, F. Benzarti, H. Amiri, "Object detection and identification for blind people in video scene," 2015 15th International Conference on

Intelligent Systems De- sign and Applications (ISDA), pp. 363-367, 2015.

5. Košale U, Žnidaršic P, Stopar K, “Detection of different shapes and materials by glasses for blind and visually impaired,” 2019 6th Student

Computer Science Research Conference, Vol. 57, 2019.

6.Ayushi Jaiswal Ajay Somkuwar Jayprakash Upadhyay ‘Image denoising and quality measurements by using filtering and wavelet based

techniques’ AEU - International Journal of Electronics and Communications 68(8):699-705

7.E. Barri, A. Gkamas, E. Michos , C. Bouras, C. Koulouri, S.A.K. Salgado, “Text to Speech through Bluetooth for People with Special Needs

Navigation,” International Conference on Ne8tworking and Services, 2010.

8. H. A. Karimi, M.B. Dias, J. Pearlman, G. J Zimmerman, “Wayfinding and Navigation for People with Disabilities Using Social Navigation

Networks,” EAI Endorsed Transactions on Collaborative Computing, Vol. 1, Issue 2, e5, October, 2014.

9. Rahman Mohammad Marufur, Md. Milon Islam, Saeed Anwar Khan, Shishir Ahmmed "Obstacle and fall detection to guide the visually

impaired people with real time monitoring," SN Computer Science, Vol. 1, pp 1-10 (219), 2020.