GESTURE VOLUME CONTROL

Presented By:

Lokam Teja Ram – B200793CS

E Krishna Prasad – B200803CS

Anagha M V – B200762CS

P Gnana Prakash - B200840CS

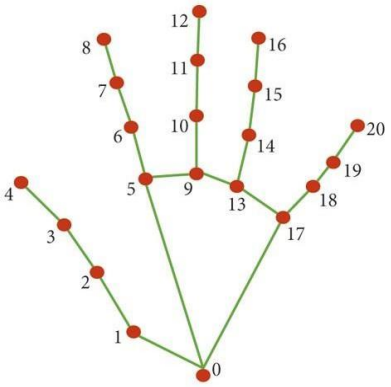
Abhina Sunny – B200692CS

PROPOSED METHODOLOGY:

Mediapipe:

MediaPipe is a software framework developed by Google that helps computers see and understand the world through video input. It's like giving computers the ability to analyse and interpret images and videos in real-time. MediaPipe is often used for tasks like recognizing and tracking objects, detecting hand movements, or identifying facial expressions. It's a powerful tool for creating applications that involve computer vision and understanding the visual world.

Single-shot detector model is used for detecting and recognizing a hand or palm in real time. single shot detector model is used by the MediaPipe. First, in the hand detection module, it is first trained for a palm detection model because it is easier to train palms. Furthermore, the non-maximum suppression works significantly better on small objects such as palms or fists. A model of hand landmark consists of locating 21 joint or knuckle co-ordinates in the hand region, as shown in figure.



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

OpenCV:

OpenCV is a computer vision library which contains image-processing algorithms for object detection. OpenCV is a library of python programming language, and real-time computer vision applications can be developed by using the computer vision library. OpenCV library is used in image and video processing and also analysis such as face detection and object detection.

The various functions and conditions used in the system are explained in the flowchart of the real-time gesture volume control system as shown in Figure.

The proposed gesture volume control system is based on the frames that have been captured by the webcam in a laptop or PC. By using the Python computer vision library **OpenCV**, the video capture object is created and the web camera will start capturing video.

Capturing the Video and Processing:

The Gesture volume control system uses the webcam where each frame is captured till the termination of the program. The video frames are processed from BGR to RGB colour space to find the hands in the video frame by frame.

Once we get the 21 points of the hand, we take co-ordinates of thumb and index finger then find the distance between those two points using the formula

$$\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

Then according to the distance we change the volume of the device using pyautogui .

STEPS INVOLVED:

Data Preparation:
No specific dataset preparation is required as the system operates in real-time, capturing frames from the webcam.

Feature Extraction:
Utilize OpenCV and MediaPipe Hands library to detect hand landmarks in each frame.
Extract relevant features such as the distance between specific landmarks representing fingertips.

Gesture Classification:
Define thresholds for the distances between landmarks to classify gestures (e.g., volume up, volume down, neutral).

Volume Control:
Based on the detected gestures, adjust the system volume accordingly using PyAutoGUI or other appropriate libraries.

Real-time Processing:
Continuously capture frames from the webcam.
Process each frame to detect hand gestures and adjust the volume accordingly.

Performance Evaluation:
Validate the system's performance in real-time by interacting with the gesture volume control system

TOOLS AND TECHNIQUES:

OpenCV (cv2):
OpenCV (Open Source Computer Vision Library) is a popular library used for computer vision tasks such as image and video manipulation, object detection, and more.

Mediapipe:
Mediapipe is a framework developed by Google that provides tools for building machine learning pipelines for various perception tasks, including body pose estimation, hand tracking, facial recognition, and more.Here we utilize the mediapipe module for detecting and tracking hand landmarks in the webcam feed.

pyautogui:
PyAutoGUI is a Python library for automating keyboard and mouse interactions. In this code, it's used to simulate key presses based on hand gestures detected by the hand tracking model.here it is used to simulate pressing the volume up or volume down keys.

Image Processing:
Techniques such as converting the color space of the image from BGR to RGB and extracting hand landmarks coordinates from the detected hand landmarks are used for processing the webcam frames.

Hand Landmarks Detection:
The code detects hand landmarks using the pre-trained hand tracking model provided by Mediapipe.It then extracts the coordinates of specific landmarks (such as fingertips) to determine hand gestures.

EXISTING APPROACHES:
Here are some of the existing approaches for gesture volume control.

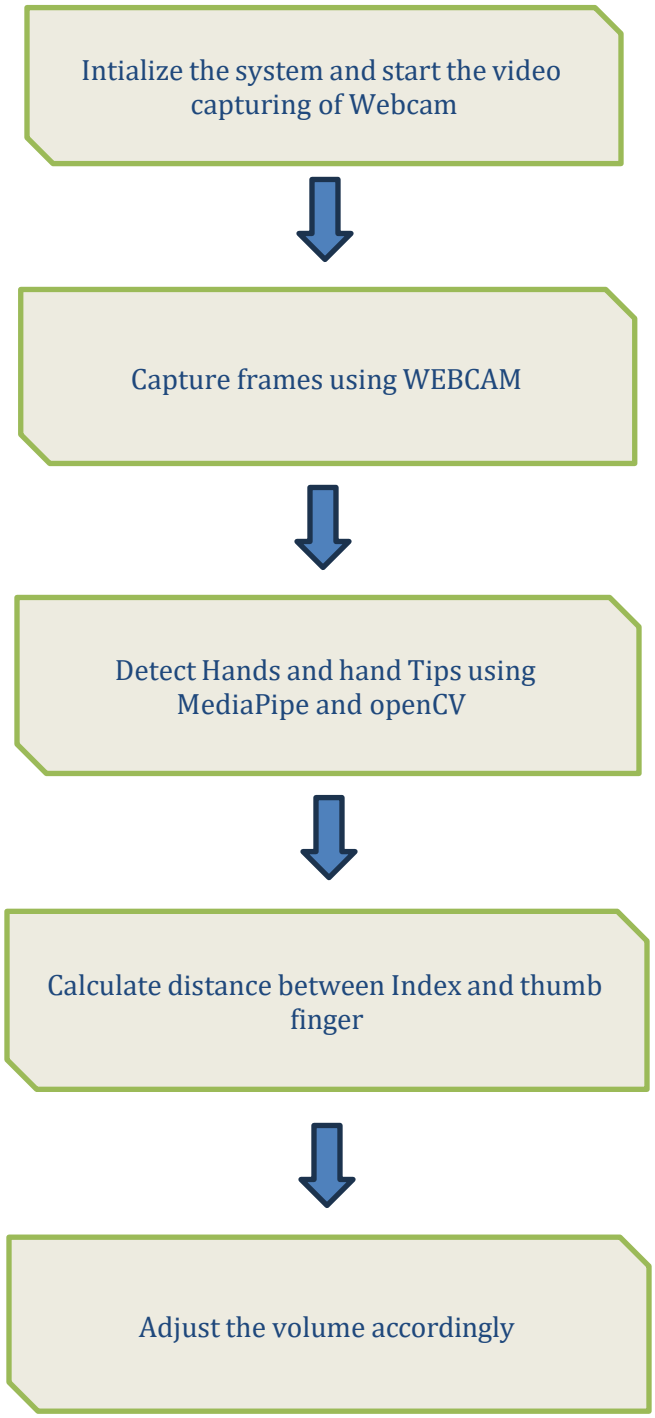
Depth Sensors(e.g.,Kinect):
Depth sensors like Kinect can track hand movements in 3D space. Users can perform gestures such as moving their hand up or down to control volume.

Wearable devices:
Wearable devices equipped with motion sensors, such as smartwatches or wristbands, can detect gestures like wrist rotations or taps to adjust volume on connected devices.

Smartphone Apps:
Some smartphone apps utilize the device's built-in accelerometer and gyroscope to recognize gestures for volume control. For instance, users can tilt their phone up or down to adjust volume.

Gesture Recognition Software:
Dedicated gesture recognition software can be installed on computers or embedded in devices to interpret specific hand movements as volume control commands. This software often requires training to recognize and map gestures accurately.

DESIGN:


Intialize the system and start the video capturing of Webcam
Capture frames using WEBCAM
Detect Hands and hand Tips using MediaPipe and openCV
Calculate distance between Index and thumb finger
Adjust the volume accordingly

CODING:

```
import cv2
import mediapipe as mp
import pyautogui

x1 = y1 = x2 = y2 = 0

webcam = cv2.VideoCapture(0)
my_hands = mp.solutions.hands.Hands()
drawing_utils = mp.solutions.drawing_utils
while True:
    _, image = webcam.read()
    image = cv2.flip(image,1)
    frame_height,frame_width, _ = image.shape

    rgb_image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
    output = my_hands.process(rgb_image)
    hands = output.multi_hand_landmarks
    if hands:
        for hand in hands:
            drawing_utils.draw_landmarks(image,hand)
            landmarks = hand.landmark
            for id,landmark in enumerate(landmarks):
                x=int(landmark.x * frame_width)
                y=int(landmark.y * frame_height)
                if id == 8:
                    cv2.circle(img=image,center=(x,y),radius=8,color=(0,255,25
5),thickness=3)
                    x1=x
                    y1=y
                if id == 4:
                    cv2.circle(img=image,center=(x,y),radius=8,color=(0,0,255)
,thickness=3)
                    x2=x
                    y2=y
        dist=((x2-x1)*2 +(y2-y1)2)*(0.5)//4
        cv2.line(image,(x1,y1),(x2,y2),(0,255,0),5)
        if dist > 30 :
            pyautogui.press("volumeup")
        else:
            pyautogui.press("volumedown")
    cv2.imshow("Volume control using python", image)
    key = cv2.waitKey(10)
    if key == 27:
        break
webcam.release()
cv2.destroyWindow()
```