Project Report

On

SYSTEM VOLUME CONTROL USING HAND GESTURE RECOGNITION

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Engineering (Data Science)

by

Devansh Jain

(20001016015)

under the supervision of

Ms. Monika Gupta

Department of Computer Engineering

J. C. BOSE UNIVERSITY OF SCIENCE & TECHNOLOGY, YMCA

FARIDABAD-121006

August 2022

CANDIDATE'S DECLARATION

I hereby certify that the work which is being carried out in this Project titled **"System Volume Control using Hand Gesture Recognition"** in fulfillment of the requirement for the degree of Bachelor of Technology in **Computer Engineering (Data Science)** and submitted to "**J. C. Bose University of Science and Technology, YMCA, Faridabad**", is an authentic record of my own work carried out under the supervision of **Ms. Monika Gupta**.

Devansh Jain

20001016015

CERTIFICATE

This is to certify that the work carried out in this project titled **“System Volume Control using Hand Gesture Recognition”** submitted by **Devansh Jain** to “**J. C. Bose University of Science and Technology, YMCA, Faridabad**” for the award of the degree of Bachelor of Technology in **Computer Engineering (Data Science)** is a record of bonafide work carried out by them under my supervision. In my opinion, the submitted report has reached the standards of fulfilling the requirements of the regulations to the degree.

Ms. Monika Gupta
(Mentor)
Assistant Professor
Department of Computer Engg.
J. C. Bose University of Science and Technology, YMCA, Faridabad

Dr. Komal Kumar Bhatia
Chairman,
Department of Computer Engg.,
J. C. Bose University of Science and Technology, YMCA, Faridabad

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

INTRODUCTION

The **Volume Control With Hand Detection OpenCV Python** was developed using **Python OpenCV**, In this **Python OpenCV Project** we are going **Building a**
Volume Controller with OpenCV, to change the volume of a computer. We first look into hand tracking and then we will use the hand landmarks to find gesture of our hand to change the volume. This project is module based which means we will be using a previously created hand module which makes the hand tracking very easy.
Hand detection is a simple idea, but it requires some advanced tools to do it. This project is one of the system that can detect the gesture of hand in a real time video. A **Hand Detection** is one of the most viable and popular solution for improving human computer interaction.

PROBLEM IDENTIFICATION

What is OpenCV and MediaPipe ?

OpenCV is short for Open Source Computer Vision. Intuitively by the name, it is an open-source Computer Vision and Machine Learning library. This library is capable of processing real-time image and video while also boasting analytical capabilities. It supports the Deep Learning frameworks.

A **OpenCV Python Hand Landmark Detection**, We will be using **OpenCV** to read the image and displaying it and **MediaPipe** to perform the hand detection and landmark estimation. In short, **MediaPipe** is a free and open-source framework that offers cross-platform, customizable Machine Learning solutions for live and streaming media.

OBJECTIVE

- The main objective of this project is to relieve the user from using mouse, keyboard as input method to control the System Volume.
- Gesture recognition helps computers to understand human body language. This helps to build a more potent link between humans and machines, rather than just the basic text user interfaces or graphical user interfaces (GUIs).
- In this project for gesture recognition, the human body's motions are read by computer camera. The computer then makes use of this data as input to handle applications.
- Hand gesture recognition comes under the field of machine learning.
- Machine Learning is the need of the generation. In near future we are likely to witness nearly everything to be working on Machine Learning and Artificial Intelligence. So, this project is basically a starting guide as to HOW are we going to feed in the data to the machine, run it during the training and the machine creates its own program(logic) which can be evaluated while testing. There are already a lot of contributors in the open-source market creating libraries which makes the task easier and make proper working machine learning models. Libraries such as OpenCV, MediaPipe are widely used in such areas and we are also going to use the same to implement this project.

CHAPTER 2: DESCRIPTION AND METHODOLOGY

FLOW OF WORK :-

Step 1. Detect Hand landmarks

Step 2. Calculate the distance between thumb tip and index finger tip.

Step 3. Map the distance of thumb tip and index finger tip with volume range. For this case, distance between thumb tip and index finger tip was within the range of 13 – 280 and the volume range was from -65.25 – 0.0.

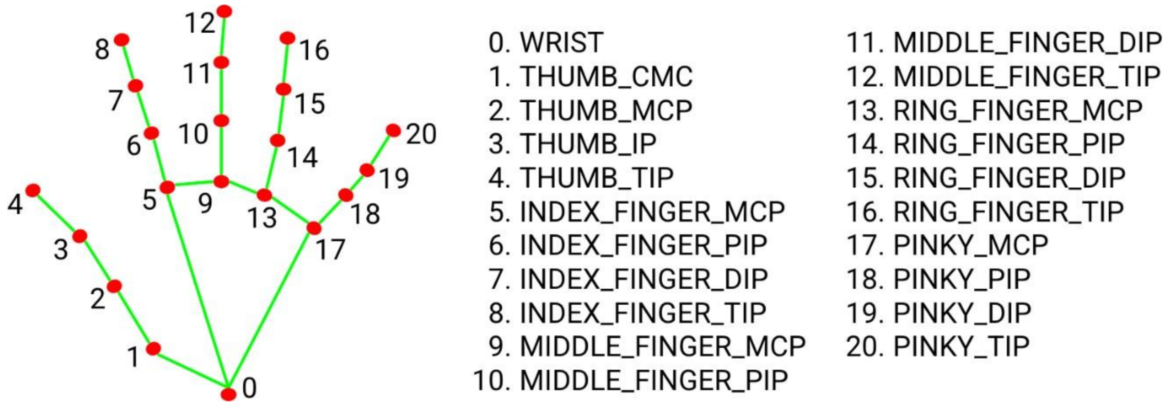
HOW TO CARRY FORWARD ?

The project is carried forward in 2 steps :-

1. We will first look into how to create hand tracking module.
2. Then we will use hand landmarks to find the gesture of our hands and integrate this module into our real working model and make changes according to our desired needs.

This hand tracking module acts as a base module in various other models like – brightness control, media player, VR games, etc.

21 HAND LANDMARKS



CHAPTER 3: HARDWARE AND SOFTWARE REQUIREMENTS

Hardware and software requirements of any project are must to be satisfied, so that the virtual environment can be set up on any machine to run the project. So, in this section the software and the hardware requirements are discussed completely.

Software Requirements

Code Editor – VS Code

OS – Windows 10

Language – Python

Libraries – OpenCV, MediaPipe, NumPy, math, pycaw

Hardware Requirements

a. A computer

b. Minimum System Requirements: -

   i. O.S. -Windows 10

   ii. Quad-core i5 CPU minimum

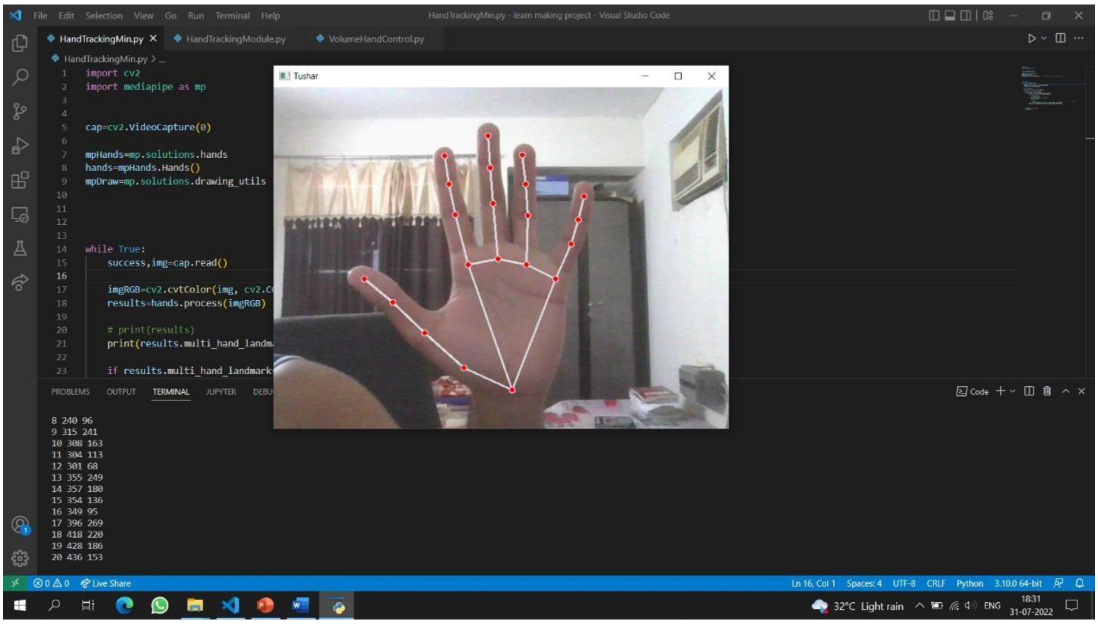
   iii. 8 GB RAM

   iv. 256GB SSD

c. Webcam

CHAPTER 4: RESULTS (SCREENSHOTS)

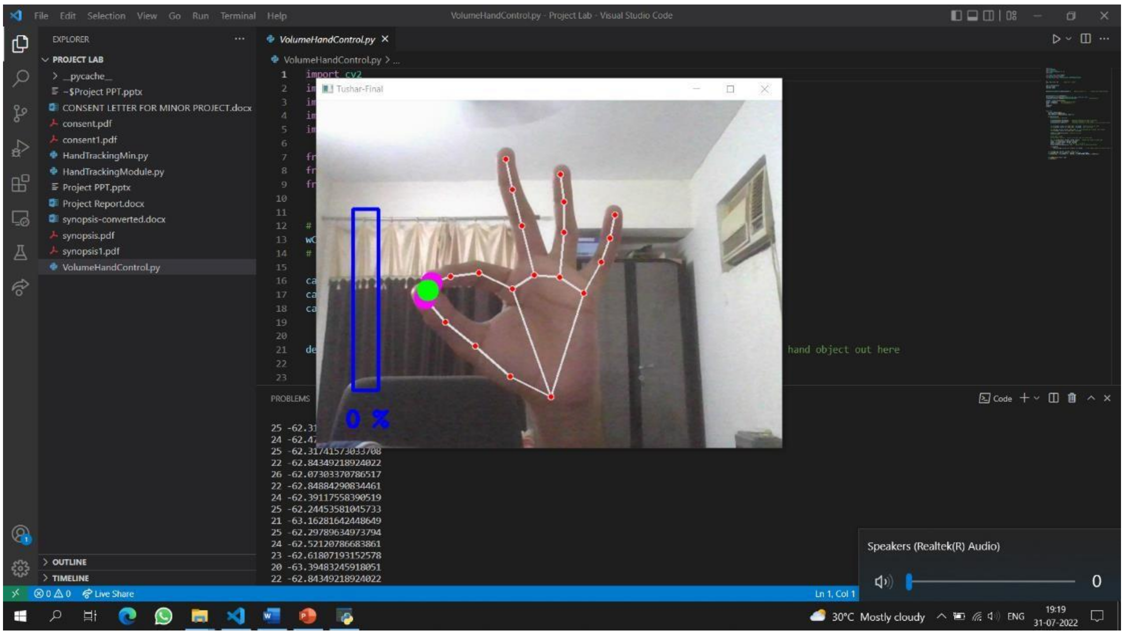
Hand Tracking Module


File Edit Selection View Go Run Terminal Help
HandTrackingMin.py - learn making project - Visual Studio Code
HandTrackingMin.py
HandTrackingModule.py
VolumeHandControl.py
HandTrackingMin.py > ...
Tushar
PROBLEMS OUTPUT TERMINAL JUPYTER DEBU
Code
0 0 Live Share
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.0 64-bit
32°C Light rain ENG 18:31 31-07-2022

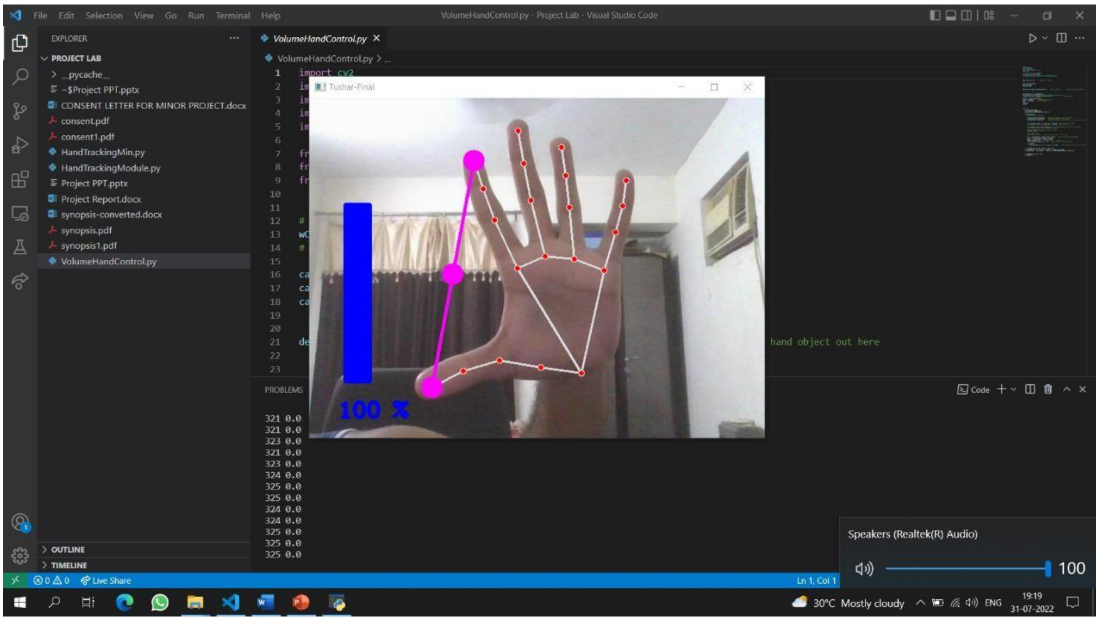
```
import cv2
import mediapipe as mp


cap=cv2.VideoCapture(0)

mpHands=mp.solutions.hands
hands=mpHands.Hands()
mpDraw=mp.solutions.drawing_utils




while True:
    success,img=cap.read()

    imgRGB=cv2.cvtColor(img, cv2.C
    results=hands.process(imgRGB)

    # print(results)
    print(results.multi_hand_landm

    if results.multi_hand_landmark
```

```
8 240 96
9 315 241
10 308 163
11 304 113
12 301 68
13 355 249
14 357 180
15 354 136
16 349 95
17 396 269
18 418 220
19 428 186
20 436 153
```

Controlling Volume



```
25 -62.31
24 -62.47
25 -62.31741573033708
22 -62.84349218924022
26 -62.07303370786517
22 -62.84884290834461
24 -62.39117558390519
25 -62.24453581045733
21 -63.16281642448649
25 -62.29789634973794
24 -62.52120786683861
23 -62.61807193152578
20 -63.39483245918051
22 -62.84349218924022
```



```
321 0.0
321 0.0
323 0.0
321 0.0
323 0.0
324 0.0
325 0.0
325 0.0
324 0.0
324 0.0
325 0.0
325 0.0
325 0.0
```

CHAPTER 5: CONCLUSION

EXPECTED OUTCOMES

1. We will have a complete working application for the model.
2. Webcam will be continuously running to capture hand gestures.
3. Application shows all the plotting points on your hands to find out the relative distance between your thumb and index finger.
4. Application can detect your both hands in the frame but can only allow gestures from one hand to control the volume.
5. You can now control volume of your system with your hand gestures without using any other input methods like mouse, keyboard.

BRIEF PROFILE OF STUDENT

Name	:	Devansh Jain
Roll No	:	20001016015
Branch	:	Computer Engineering (Data Science)
Email Id	:	20001016015@jcboseust.ac.in

Brief About Project:

The Project revolves around basics of Hand Tracking. The objective of the project is to use machine learning tools to build a working model to control system volume. We use MediaPipe library to train this model and use webcam as input source to give our hand data instead of using traditional mouse/keyboard as an input.

Future Scope:

The project includes controlling volume only. The project can be extended to feature proper media control options like- play/pause, mute, next, previous, fast forward, rewind, etc.