**ACKNOWLEDGEMENT**

we extend our sincere and heartfelt thanks to our esteemed guide, Ms. Priyanka Shukla and for her exemplary guidance, monitoring and constant encouragement throughout the course at crucial junctures and for showing us the right way.

We would like to extend thanks to our respected Head of the division, Dr. Munish Sabharwal for allowing us to use the facilities available. We would like to thank other faculty members also, Last but not the least,

We would like to thank our friends and family for the support and encouragement they have given us during the course of our work.

**Abstract**

The idea behind this project is that, with the help of Thumb, index and pinky fingers we will increase and decrease the volume of our computer/laptop.

When the pinky finger down then we will increase and decrease the volume with the help of thumb and index finger. Once pinky finger up then it will set the volume of the computer, means that we will change volume with thumb and index finger but we will set the volume with pinky finger.

* It helps computer to understand human body language.
* It is very easy to use and hassle free.
* Calculate the distance between thumb tip and index finger tip.

**Table of Contents**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | |  | | | **Page No.** |
| **Candidates Declaration** | | | | | **I** |
| **Acknowledgement** | | | | | **II** |
| **Abstract** | | | | | **III** |
| **Contents** | | | | | **IV** |
|  | | | | |  |
| **Chapter 1** | **Introduction** | | | | **1** |
|  |  | |  | |  |
|  |  | |  | |  |
|  |  | |  |  |  |
| **Chapter 2** | **Software used in the project** | | | | **2** |
|  |  | |  |  |  |
|  |  | |  |  |  |
| **Chapter 3** | **Language Used** | | | | **3** |
|  |  | |  |  |  |
|  |  | |  |  |  |
| **Chapter 4** | **Libraries used** | | | | **4** |
|  |  | |  |  |  |
|  |  | |  |  |  |
| **Chapter 5** | **Algorithm of the project** | | | | **14** |
|  | 5.1 | | Code explanation | | **22** |
|  |  | |  | |  |
|  | **Reference** | | | |  |
|  | **Conclusions**  **Project Sample Images** | | | |  |
|  |  | |  |  |  |

1. **INTRODUCTION**

PROJECT:

VOLUME CONTROL USING HAND GESTURE

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. The objective

of this project is to develop an interface which will capture human hand gesture

dynamically and will control the volume level.

Use gesture control to change the volume of a computer. First, we look into hand

tracking and then we use hand landmarks to find gestures of our hand to change

volume. Distance between index finger and thumb is used to change volume level

**SOFTWARE USED IN THIS PROJECT:**

**PYCHARM (2021.3):**

PyCharm is compatible with Linux, macOS, and Windows platforms. Sitting

gracefully among the best Python IDEs, PyCharm provides support for both

Python 2 (2.7) and Python 3 (3.5 and above) versions.

PyCharm comes with a plethora of modules, packages, and tools to hasten Python

development while cutting-down the effort required to do the same to a great

extent, simultaneously. Further, PyCharm can be customized as per the

development requirements, and personal preferences call for. It was released to the

public for the very first time back in February of 2010. In addition to offering code

analysis, PyCharm features:

1. A graphical debugger
2. An integrated unit tester
3. Integration support for version control systems (VCSs)
4. Support for data science with Anaconda

**LANGUAGE USED:**

**PYTHON:**

Python is an interpreted high-level general-purpose programming language. Its

design philosophy emphasizes code readability with its use of significant

indentation. Its language constructs as well as its object- oriented approach aim to

help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple

programming paradigms, including structured (particularly procedural), object-

oriented and functional programming. It is often described as a "batteries included"

language due to its comprehensive standard library.



**PYTHON LIBRARIES**

We know that a module is a file with some Python code, and a package is a

directory for sub packages and modules. But the line between a package and a

Python library is quite blurred.

A Python library is a reusable chunk of code that you may want to include in your

programs/ projects.

Compared to languages like C++ or C, a Python library do not pertain to any

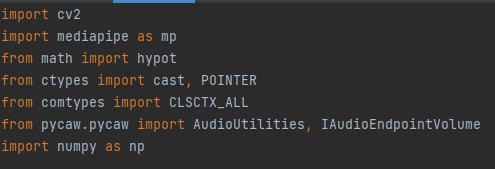
specific context in Python. Here, a ‘library’ loosely describes a collection of core

modules.

Essentially, then, a library is a collection of modules. A package is a library that

can be installed using a package manager like rubygems or npm.

LIBRARIES USED IN THIS PROJECT:



**OPENCV:**

OpenCV is a Python open-source library, which is used for computer vision in Artificial intelligence, Machine Learning, face recognition, etc.

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

OpenCV is available for free of cost.

1. Since the OpenCV library is written in C/C++, so it is quite fast. Now it can be used with Python.
2. It requires less RAM to usage, it maybe of 60-70 MB.
3. Computer Vision is portable as OpenCV and can run on any device that can run on C.

CV:

The CV is an abbreviation form of a computer vision, which is defined as a field of study that helps computers to understand the content of the digital images such as photographs and videos. The purpose of computer vision is to understand the content of the images.

### CV2:

cv2 (old interface in old OpenCV versions was named as cv) is the name that OpenCV developers chose when they created the binding generators. This is kept as the import name to be consistent with different kind of tutorials around the internet.

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. cv2.putText () method is used to draw a text string on any image. Syntax: cv2.putText (image, text, org, font, fontScale, color [, thickness [, lineType [, bottomLeftOrigin]]])



# Import NUMPY as np:

#### NUMPY is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

NumPy’s high level syntax makes it accessible and productive for programmers from any background or experience level.



NumPy is the fundamental package for array computing with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data- types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**IMPORT MATHS:**

Python maths module is a standard module and is always available in python to do mathematical operations easily. To Import math in python is to give access to the mathematical functions, which are defined by the C standard. In this tutorial, you will learn about some important math module functions with examples in python.

To use Python math functions, you have to import the module using the import math line at the starting of the program to get the math class object. Using a math class object can access any math function in python.

he functions under the math module do not support a complex number or you can’t use. For use a complex number with this function you have to use other modules in python.



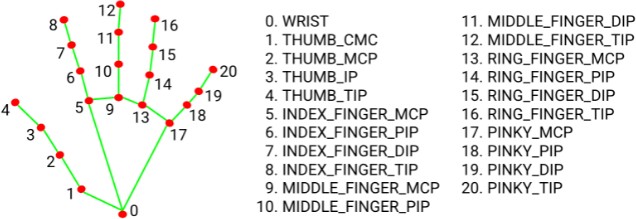


**IMPORT MEDIAPIPE AS MP:**

Hand Landmark Model:

After the palm detection over the whole image our subsequent hand landmark model performs precise key point localization of 21 3D hand- knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions.

To obtain ground truth data, we have manually annotated ~30K real- world images with 21 3D coordinates, as shown below (we take Z-value from image depth map, if it exists per corresponding coordinate). To better cover the possible hand poses and provide additional supervision on the nature of hand geometry, we also render a high-quality synthetic hand model over various backgrounds and map it to the corresponding 3D coordinates.



## **Solution APIs**

STATIC\_IMAGE\_MODE:

If set to false, the solution treats the input images as a video stream. It will try to detect hands in the first input images, and upon a successful detection further localizes the hand landmarks.

## MAX\_NUM\_HANDS:

Maximum number of hands to detect. Default to 2.

MODEL\_COMPLEXITY:

Complexity of the hand landmark model: 0 or 1. Landmark accuracy as well as inference latency generally go up with the model complexity.

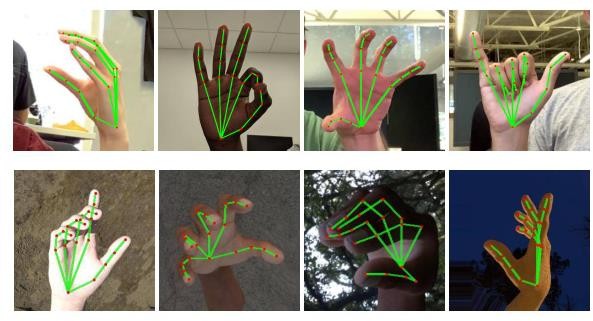
Default to 1.

MIN\_DETECTION\_CONFIDENCE:

Minimum confidence value ([0.0, 1.0]) from the hand detection model for the detection to be considered successful. Default to 0.5.

MIN\_TRACKING\_CONFIDENCE:

Minimum confidence value ([0.0, 1.0]) from the landmark-tracking model for the hand landmarks to be considered tracked successfully.



Using MediaPipe Hand Landmark Model for identifying Hands:

**IMPORT PYCAW:**

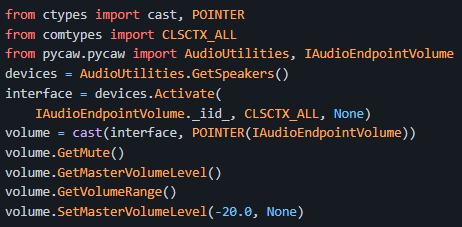
Pycaw is a Python Core Audio Windows Library, working for both Python2 and Python3.

It generally get and set the windows audio volume.

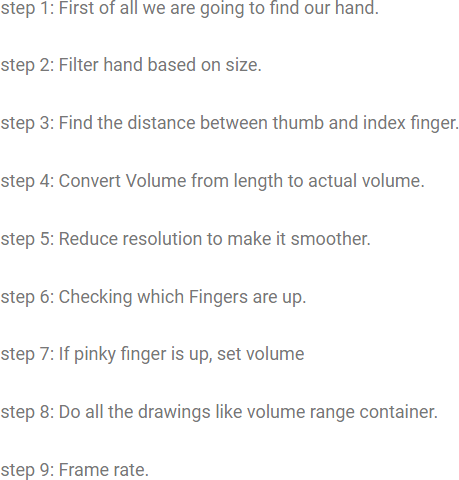
INSTALL:



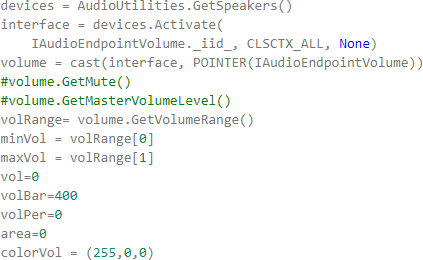
USAGE:



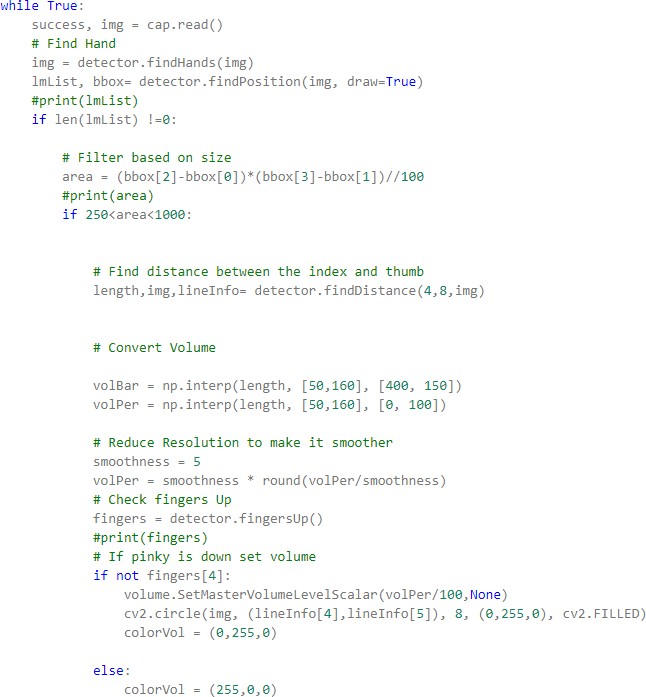
## **ALGORITHM OF THIS PROJECT:**

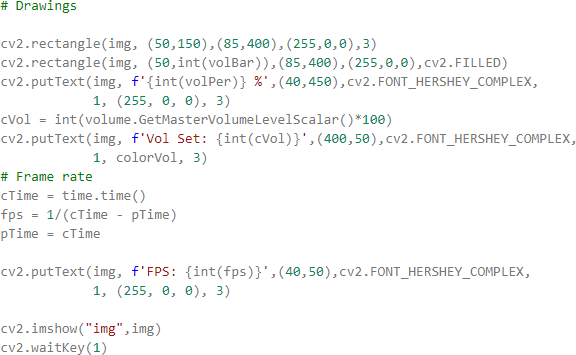


In this project there are two python files one is VolumeControl.py and another is HandTrackingModule.py

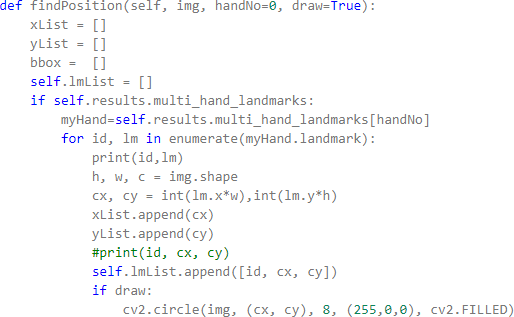


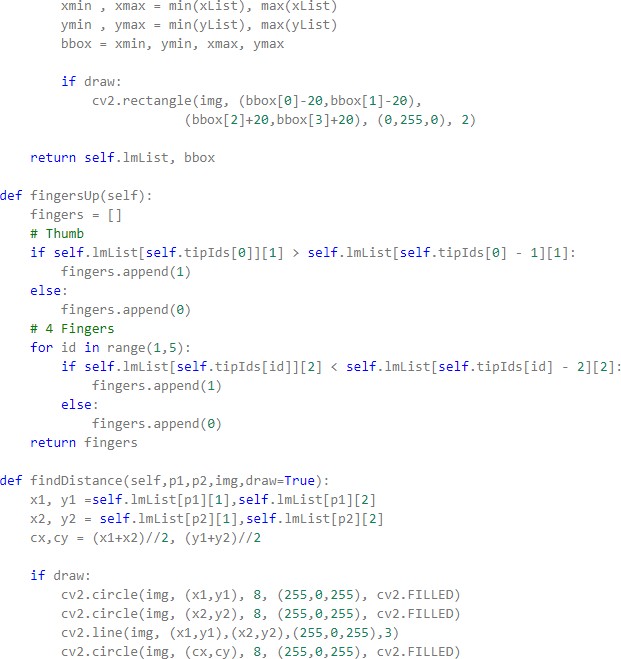


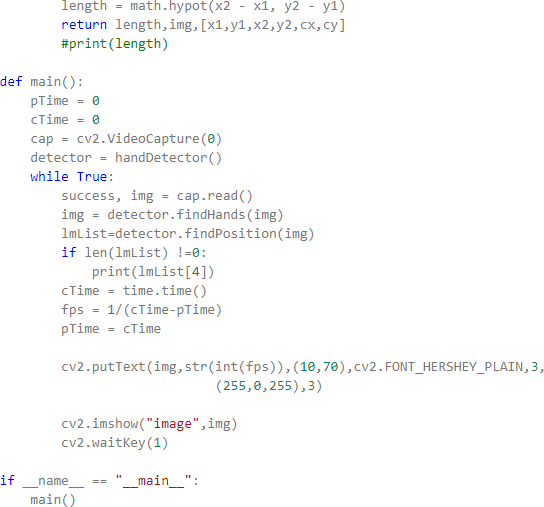






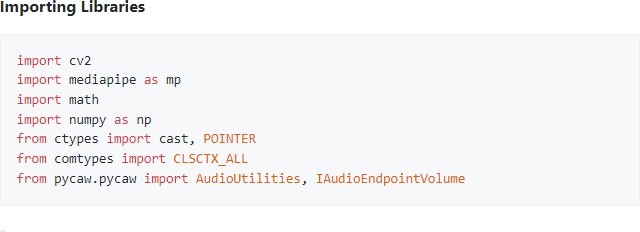


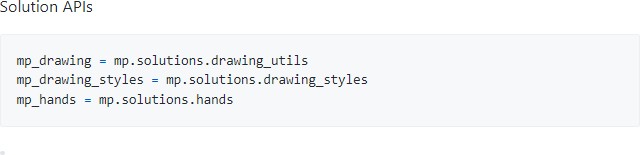




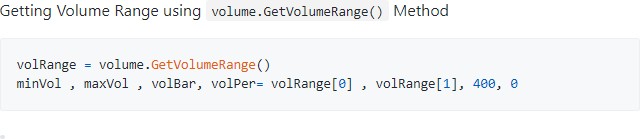
END OF THE CODE

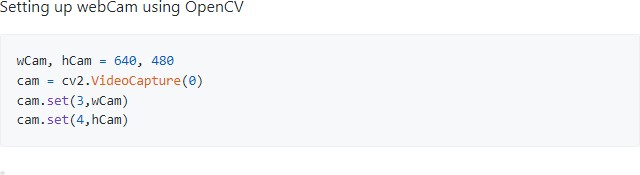
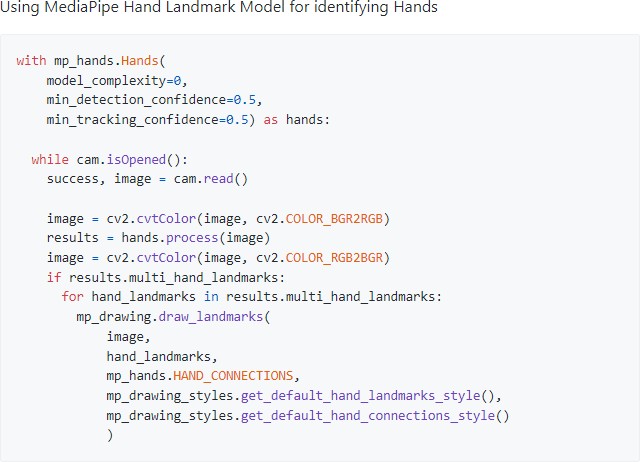
# CODE EXPLANATION:

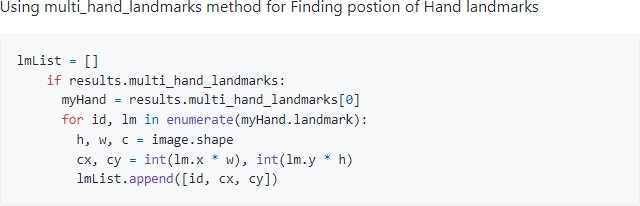


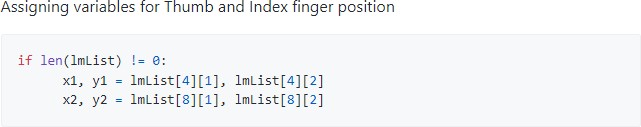


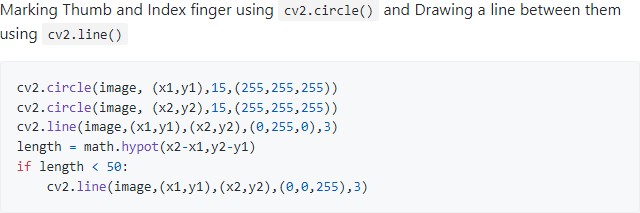


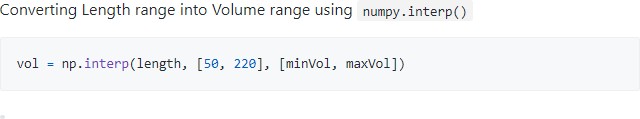












END OF CODE EXPLANATION

# REFERENCES

1. PAGE NO. 11,21 (MEDIAPIPE AND CODE EXPLANATION)

*Pratham bhatnagar*

[*GitHub - pratham-bhatnagar/Gesture-Volume-Control at*](https://github.com/pratham-bhatnagar/Gesture-Volume-Control?ref=pythonawesome.com)[*pythonawesome.com*](https://github.com/pratham-bhatnagar/Gesture-Volume-Control?ref=pythonawesome.com)

1. PAGE NO. 7 (OPENCV)

[*OpenCV Tutorial | OpenCV using Python - javatpoint*](https://www.javatpoint.com/opencv)

1. PAGE NO. 10 (IMPORT MATH)

*Rohit*

[*Python math module | Python import math | Function |*](https://tutorial.eyehunts.com/python/python-math-module-import-math-function-operator/)[*Operator - EyeHunts*](https://tutorial.eyehunts.com/python/python-math-module-import-math-function-operator/)

1. PAGE NO. 6 (PYTHON LIBRARIES)

[*Python Libraries - Python Standard Library & List of Important*](https://data-flair.training/blogs/python-libraries/#%3A~%3Atext%3DImportant%20Python%20Libraries.%201%201.%20Matplotlib.%20Matplotlib%20helps%2C4%204.%20NumPy.%205%205.%20SQLAlchemy.%20More%20items)[*Libraries - DataFlair (data-flair.training)*](https://data-flair.training/blogs/python-libraries/#%3A~%3Atext%3DImportant%20Python%20Libraries.%201%201.%20Matplotlib.%20Matplotlib%20helps%2C4%204.%20NumPy.%205%205.%20SQLAlchemy.%20More%20items)

CONT. ….

…. CONT.

1. PAGE NO. 15 (IMPORT PYCAW)

[*GitHub - AndreMiras/pycaw: Python Core Audio Windows*](https://github.com/AndreMiras/pycaw)[*Library*](https://github.com/AndreMiras/pycaw)

1. PAGE NO. 16 (ALGORITHM AND CODE)

*Ashwin*

[*https://github.com/Ashwin543/Gesture-Volume-Control*](https://github.com/Ashwin543/Gesture-Volume-Control)

1. PAGE NO. 26 (REFERENCE)

[*Mini Project Report Format - Ajay Bolar (weebly.com)*](https://ajaybolar.weebly.com/mini-project-report-format.html)

1. PYCHARM

[*PyCharm: the Python IDE for Professional Developers by*](https://www.jetbrains.com/pycharm/)[*JetBrains*](https://www.jetbrains.com/pycharm/)

1. YOUTUBE

*Murtaza's Workshop - Robotics and AI*

[*Gesture Volume Control | OpenCV Python 2021 | Computer*](https://www.youtube.com/watch?v=9iEPzbG-xLE&list=PLiVdlBrJ2CXmz8DGPSJeBp45tvPtiKDIM&index=7&ab_channel=Murtaza%27sWorkshop-RoboticsandAI)[*Vision - YouTube*](https://www.youtube.com/watch?v=9iEPzbG-xLE&list=PLiVdlBrJ2CXmz8DGPSJeBp45tvPtiKDIM&index=7&ab_channel=Murtaza%27sWorkshop-RoboticsandAI)

## **CONCLUSIONS**

WE LEARN ABOUT HOW, OPENCV WORKS, HOW COMPUTER DEALS WITH THE OPEN WORLD APPLICATION, AND OBJECT RECOGNITIONS.

WE LEARNED SOME BASIC OPENCV LIBRARIES, FOR COMPUTER VISION APPLICATIONS.

LOOKING FORWARD FOR MORE PROJECTS ON OPENCV LIKE:

1. DETECTING PNEUMONIA USING EDGE ML
2. OPENCV-POWERED MOTION SENSOR FOR SAMSUNG
3. SELF DRIVING CARS
4. SMART ATTENDENCE MODEL
5. FACE AND VOICE RECOGNITION FOR THE VISUALLY IMPAIRED ETC.

PROJECT SAMPLE IMAGES:

Image 1 : 100% volume

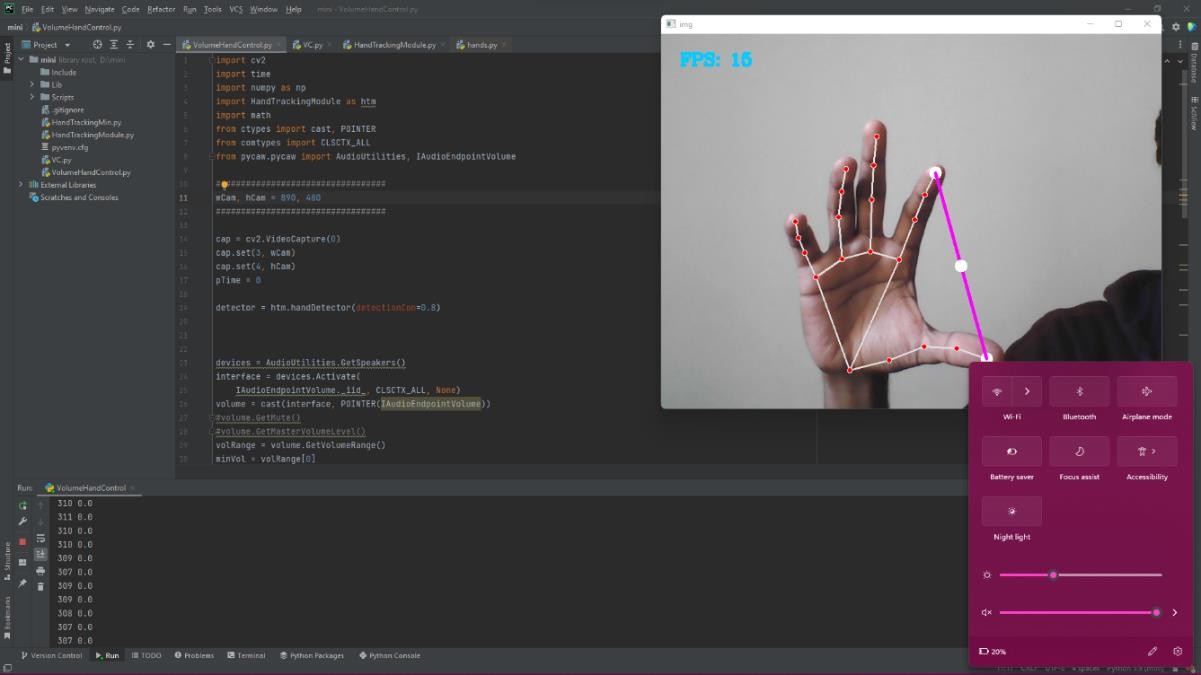


Image 2 : 0% volume

