

Rock Paper Scissors with Hand Tracking

Pulkit Gupta
2110637
pulkitgupta.211ch045@nitk.edu.in

Athul Akhileshan
2110033
athulakhileshan.211ch011@nitk.edu.in

Problem Statement -

This project aims at building a real-time hand gesture classification system and stone-paper-scissors game using Python, OpenCV and MediaPipe.

Keywords -

Tracking, OpenCV, MediaPipe, Pycaw, Real-time, Detection

I. DESIGN, PLAN OF THE PROJECT

The model is built using the OpenCV (Library used for vision-based applications) and MediaPipe (Library used for working with audio or video based data) modules. Using the features available to us through these modules, a system is built that helps in tracking the motion of fingers and give back real-time data based on that visual feed.

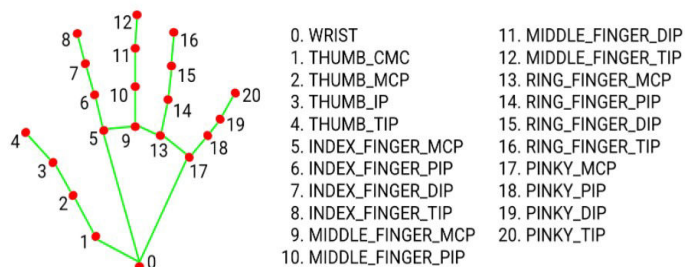


Fig. 1. Hand Landmarks

II. REFERENCES

- [1] <https://www.geeksforgeeks.org/brightness-control-with-hand-detection-using-opencv-in-python/>
- [2] <https://www.section.io/engineering-education/creating-a-hand-gesture-volume-controller-using-python-and-pycharm/>
- [3] <https://www.youtube.com/watch?v=5Hh7tOcgP0U>
- [4] <https://analyticsindiamag.com/real-time-gui-interactions-with-opencv-in-python/>
- [5] <https://www.geeksforgeeks.org/opencv-python-tutorial/?ref=lbp>

III. OBJECTIVE

Our aim is to provide an easier and more comfortable version of the idea and to make the output more pronounced and easy to comprehend.

To play the game (Stone-Paper-Scissors) in a different manner and ease, with kin and kith. The game is one of the oldest one's played and giving it a new look is accomplished through this project. It aims at providing a more interactive way at controlling the properties of one's device (brightness and volume, in this case). A model that helps in practicing, as well as, teaching basic sign-language is created. A more sophisticated model, that helps in interpreting a more wide range of hand-signs can be developed by using this model as a reference. Providing the user with a model that helps them perform all the above said tasks is the ultimate aim of the project.

IV. FEATURES OF THE FINAL DESIGN

1. Stone – Paper – Scissors Game
2. Basic HandSign Recognition
3. Volume Control using hands
4. Brightness Control using hands

IV. FINAL STATUS

1) Stone – Paper – Scissors Game -

Using the libraries, OpenCV and mediapipe, we have been able to develop a program that allows the user to play the age-old game of stone-paper-scissors. In this program, it uses the webcam of the user's device to take visual input, process it and then return the final output.

Here, the input includes two hands and the way they are held up in air (to show the symbol for stone or paper or scissors), since the game requires two people to play it. The program is able to recognise the positions of the 21 landmarks on a hand, and based on that determine what symbol a player is holding up. If there are no hands or the game is not played properly, then a message showing that the game was not played properly is shown. Next, after analysing the inputs of both the users, the game determines who wins and shows it on the screen itself.



Fig. 2. Image taken while playing game

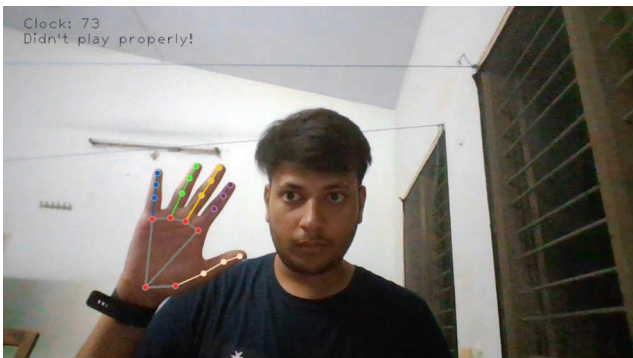


Fig. 3. Image taken while playing game (2)

2) Basic HandSign Recognition

Using the same principle as the previous game, this program is able to recognise basic hand signs and return the meaning of the symbols that a user holds up in the air.

We have implemented six different hand signs that can be recognised by the program. They include “Thumbs Up”, “Thumbs Down”, “One”, “Small”, “Rock!”, “SpideyTime”.

Whenever the user holds up their hand in the action of conveying one of the above messages, the program checks for the landmarks of the hands and determines what message the user wishes to input. It analyses the input, selects the correct message and prints it on the screen.

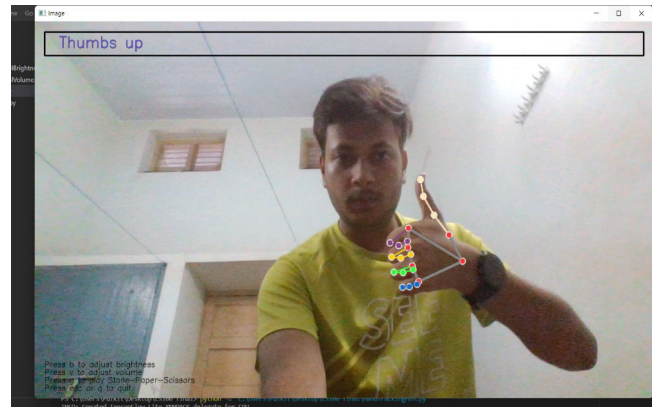


Fig. 4. “Thumbs Up”

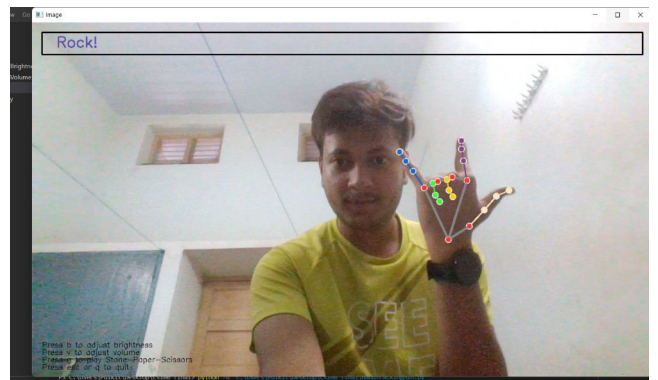


Fig. 5. “Rock!”

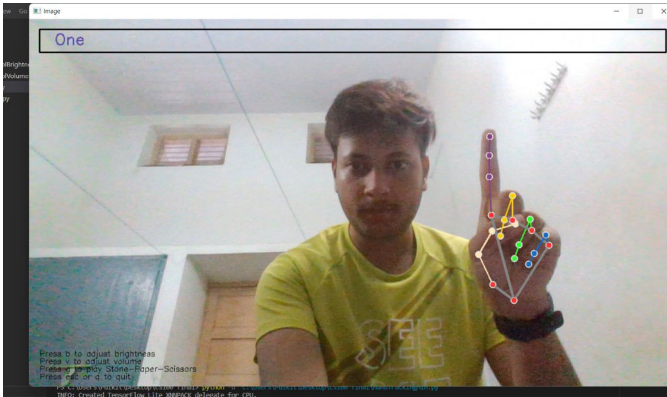


Fig. 6. “One”

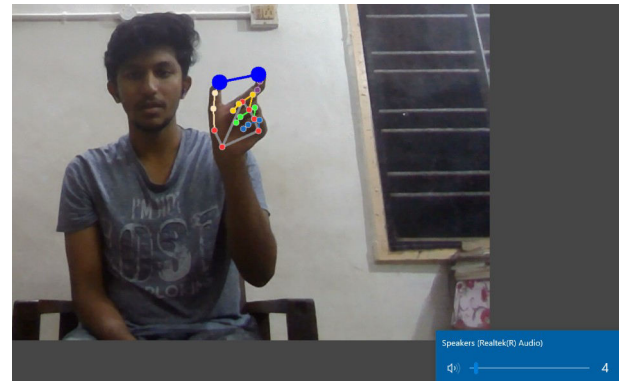


Fig. 8. Volume Control

If no handsign is being put up, the program simply returns “No Handsign”.

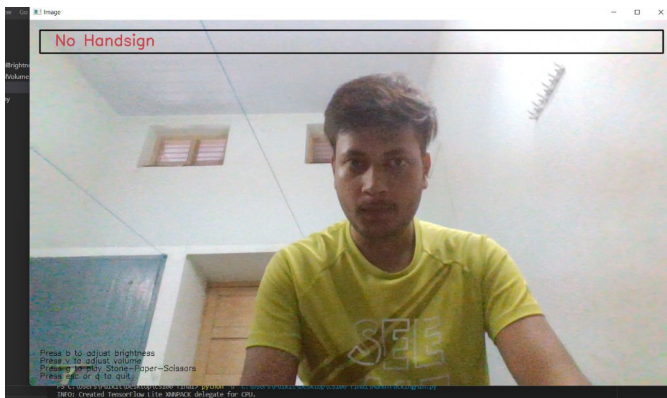


Fig. 7. “No Handsign”

3) Volume Control

Using an additional library by the name pycaw, we are able to use the inputs of the user to alter the volume level of the user’s devices.

What this program does is use the thumb finger and index finger of the user as input. To be more specific, the distance between the tips of the two fingers is calculated by the program, and accordingly, the volume is altered.

The program allows the user to change the volume of the device on a real-time basis.

4) Brightness Control

In the same sense as the previous application, this program allows the user to alter the the brightness level of their device’s screen.

This program uses the distance between the tips of the thumb and index fingers for this purpose as well.

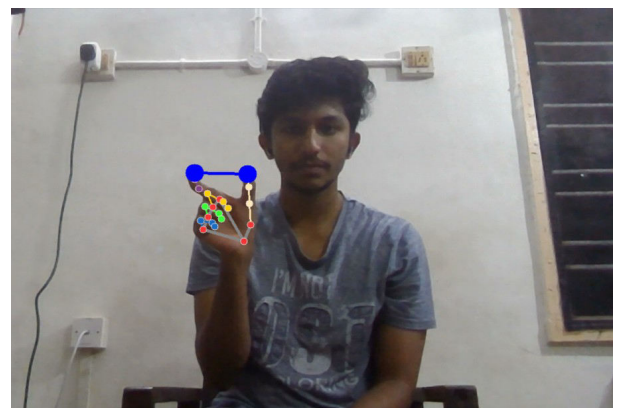


Fig. 9. Brightness Control

To make the entire program more easier to use, we have added an additional feature that allows the user to switch between the four above mentioned “modes” by another input. A key input. It is done by using OpenCV.

Key g – to play the game

Key q – to access handsign recognition program

Key b – brightness control

Key v – volume control

Key esc – to exit the program