BRIGHTNESS AND VOLUME CONTROLLER USING HAND GESTURE RECOGNITION

(Using OpenCV & Python)

A COURSE PROJECT REPORT

By

VARUN KARTHIK (RA2011003011159)
CHANDRASEKHAR REDDY (RA2011003011180)
SAI HARISH (RA2011003011185)

Under the guidance of

DR. M. ARUNA

In partial fulfillment for the Course

of

18CSC305J – Artificial Intelligence

in

Computer Science & Engineering

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chengalpattu District

April 2023

ABSTRACT

Hand Gesture Detection is receiving a lot of attention these days because it has a lot of applications and the ability to connect with machines efficiently via human interaction. We are attempting to gain knowledge of hand gesture detecting systems in this project. In this project, we're attempting to figure out how image processing works and how we can utilize it to create a hand motion recognition system that allow us to operate the computer without touching it. Many studies have been conducted and are continuously being conducted. Many big companies are currently working on this technology so that they can make their products even more useful that they are now because this technology has very high scope in the upcoming future. The people that are not mentally stable or weak from mind can also benefit by technology and can operate the computer. This technology can be used to make computers even more user-friendly.

Table of Contents

5. Conclusion and Future Enhancement

List of Tables

List of Figures

CHAPTER-1

INTRODUCTION

1.1 Introduction

The majority of the time, hands are employed for daily physical activities involving manipulation, although communication is occasionally also done with them. We use hand gestures to help us communicate accurately throughout the day. Hand gestures are essential for sign language communication since mute and deaf persons rely on their hands and gestures to communicate.

In order to communicate with and engage with machines, people most usually use their hands. The two primary input/output devices for computers the mouse and keyboard both need the use of hands to operate. The most crucial and immediate method of communication between a computer and a human is through visual and audible aids, yet this interaction is one-sided.

Gesture recognition helps computers to understand human body language. Instead of using only the simple text user interfaces or graphical user interfaces, this aids in creating a stronger connection between humans and technology (GUIs). In this gesture recognition project, a computer camera interprets the movements of the human body. This information is subsequently used as input by the computer to handle applications.

1.2 Objectives

It would be a major advance in the field of human computer interaction if computers could translate and comprehend hand gestures. The problem is that today's photos are information-rich, and in order to complete this work, substantial processing is needed. Each gesture is distinguished from others by a few unique characteristics.

The aim of this work is to create an interface that dynamically captures human hand gestures and controls volume and brightness level.

Basic Principle

This project makes advantage of our device's camera. It recognizes our hand as having points on it so that it can measure the space between the tips of our thumb and index fingers. The volume of the gadget is exactly proportional to the distance between points 4 (thumb tip) and 8 (index finger tip).

1.3 Literature Survey

Reference	**Methodology**	**Outcome/Results**	**Remarks**
Author name	Algorithms/concepts used		
Devendra Kumar Sharma; Mala Saraswat	Here the concept of hand gesture detection using the methods for extracting data and extraction of characteristics. The system is classified in three steps Extraction Method, features extraction and classification.	Hand gestures detection system is being used for different applications in different fields. Some hand gesture detection application fields are mentioned below. 1.Recognition of Sign Language 2. Controlling Features of computer 3.Virtual Environments	1. Can't be used for long distance 2. Sometimes not accurate 3. Requires a decent camera 4. May be confused by two palms
Munir Oudah Ali Al-Naji and Javaan Chahl	Here the Algorithm depends on extracting the image features in order to model visual appearance such as	The outcome is to recognition and is to introduce a system that can detect specific human gestures and use them to convey	The main remark will be if any person of physically handicapped hand who don't have proper no

1.4 Existing system

The interaction between social life and information technology has grown increasingly intimate in recent years as a result of advancements in computer hardware and software technologies. Future consumer electronics devices' interfaces, particularly those for smart phones, video games, and infotainment systems, will have an increasing number of features and be more complicated. It has become crucial to figure out how to create an easy-to-use human-machine interface (HMI) for every consumer electronics product. Gestures have been an important form of human contact and communication since ancient times. Before the development of language, people could simply convey their ideas through gestures. Many individuals still use gestures in everyday life, and deaf people in particular find gestures to be the most natural and important form of communication. Many human-based electronics items, including computers, TVs, and games, have adopted the gesture control technology in recent years.

1.5 Proposed system

The majority of gesture recognition systems include three primary steps. The detection of objects is the initial stage. This stage's goal is to identify hand items in digital images or videos. At this step, several environment and picture challenges must be resolved in order to ensure that hand contours or areas can be extracted correctly in order to improve recognition accuracy. Unstable brightness, noise, low resolution, and contrast are all common picture issues. These issues can be successfully improved by a better atmosphere and camera technologies. However, it is hard to control when the gesture recognition system is working in the real environment or is become a product. Object recognition is the second stage. To identify the movements, the detected hand objects are identified. Differentiated characteristics and successful classifier selection are important issues in most studies at this point. The third stage is analyzing successive motions in order to determine what users are instructing or doing.

CHAPTER-2

SYSTEM REQUIREMENTS

2.1 Hardware requirements

- RAM minimum of 2 GB.
- Dual Core and up ,15" Monitor.
- Integrated webcam or external webcam (15 -20 fps).

2.2 Software requirements

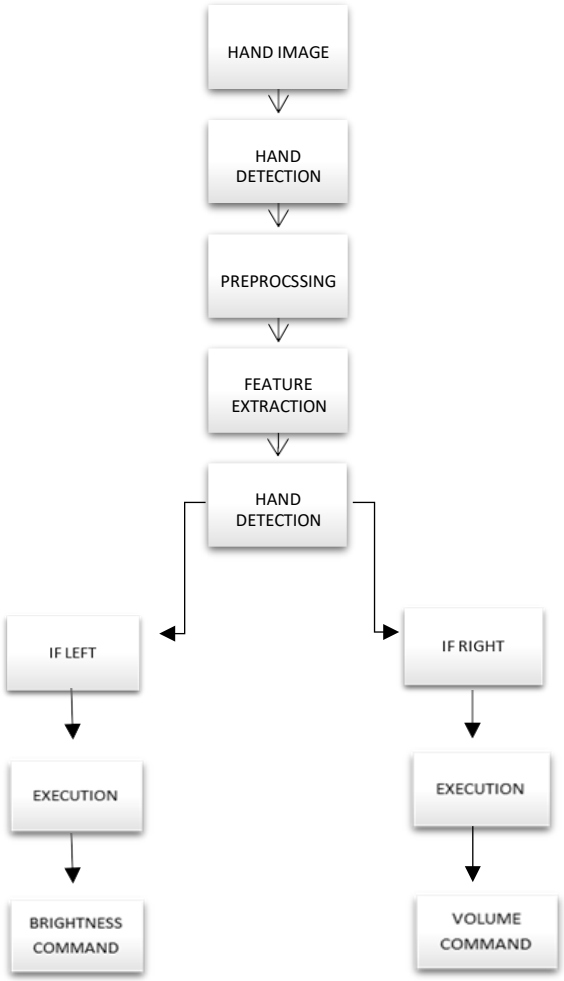
- Operating system- Microsoft Windows 7 or above
- Pycharm or Microsoft Visual Studio 2010
- Supporting Webcam Drivers
- Packages: OpenCV, Numpy, comtyes, pycaw, screen-brightness-control, mediapipe.
- **Python:** Python is a general-purpose, interpreted programming language. published for the first time in 1991 and created by Guido van Rossum
- **OpenCV:** A collection of programming functions with a focus on real-time computer vision is called OpenCV
- **Numpy:** A general-purpose array processing software is called Numpy. It offers a multidimensional array object with outstanding speed as well as capabilities for interacting with these arrays.
- **Mediapipe:** MediaPipe is a module for processing video, audio and several types of related data across platform like Android, iOS, web, edge device and several applied ML pipeline. Several types of functions are performed with the help of this module, we have used this module in our project to recognize the hand gesture and detect the input from it.
    1. Face Detection
    2. Multi-hand Tracking
    3. Object Detection and Tracking

CHAPTER-3

SYSTEM DESIGN

3.1 System architecture

3.2 Algorithms/ Flow charts



CHAPTER-4

IMPLEMENTATION

4.1 Pseudocode

To Detect the hand:

```
class handDetector: def init__(self, mode=False, maxHands=2,
detectionCon=0.7, trackCon=0.5):
    self.results = None self.mode = mode
self.maxHands = maxHands self.detectionCon =
detectionCon self.trackCon = trackCon self.mpHands =
mp.solutions.hands self.hands =
self.mpHands.Hands(self.mode, self.maxHands,
                   self.detectionCon, self.trackCon)
self.mpDraw = mp.solutions.drawing_utils def
findHands(self, img, draw=True):
    imgRGB = cv.cvtColor(img, cv.COLOR_BGR2RGB)
self.results = self.hands.process(imgRGB) if
self.results.multi_hand_landmarks: for handLms in
self.results.multi_hand_landmarks: if draw:
            self.mpDraw.draw_landmarks(img, handLms,
self.mpHands.HAND_CONNECTIONS) return img
```

To Detect the hand positions:

```
  def findPosition(self, img, draw=True): rlmlist
= [] if self.results.multi_hand_landmarks:
if len(self.results.multi_hand_landmarks) == 2:
rlmlist.append('both') elif
len(self.results.multi_hand_landmarks) == 1:
          rlmlist.append(self.results.multi_handedness[0].classification[0].label)
for n in self.results.multi_hand_landmarks:
```

```
        lmList = []              myHand = n
for id1, lm in enumerate(myHand.landmark):
            h, w, c = img.shape                cx,
cy = int(lm.x * w), int(lm.y * h)
lmList.append([id1, cx, cy])              if
draw:
                cv.circle(img, (cx, cy), 15, (255, 0, 255), cv.FILLED)
rlmlist.append(lmList)      return rlmlist
```

To control the volume and brightness using hand gesture

- **Right hand- Volume**
- **Left hand- Brightness**

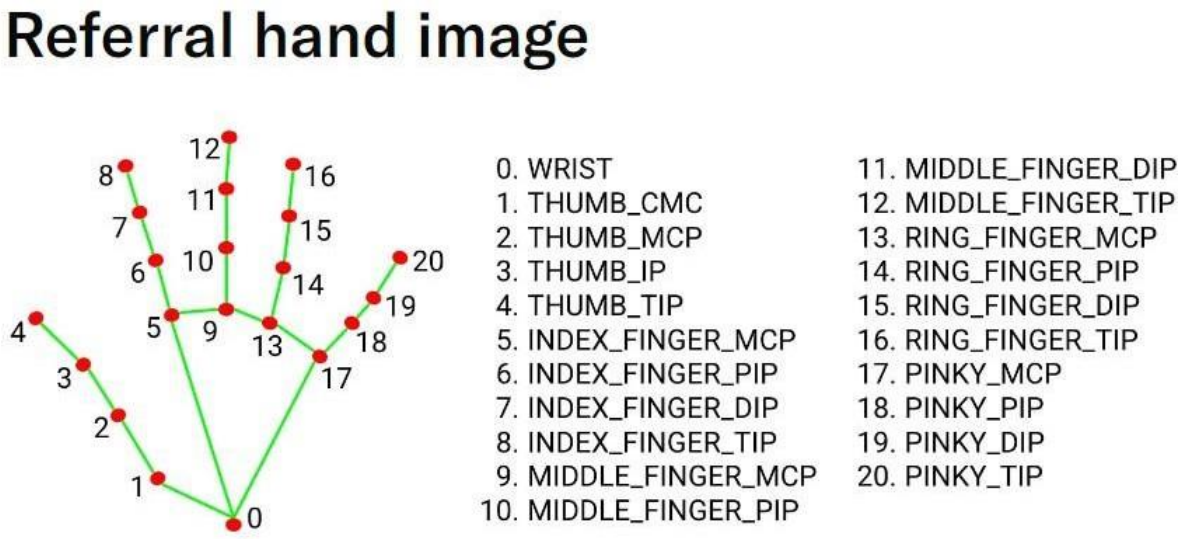


Figure 4.1.0 Hand Landmarks

The figure 4.1.0 displays the MediaPipe point numbers, which are used to identify various hand points. After detecting the palm over the whole picture, our subsequent hand landmark model uses regression, or direct coordinate prediction, to accomplish exact key point localization of 21 (figure 4.1.0) 3D hand-knuckle coordinates inside the identified hand areas.

So, we are using the 4-thumb tip and 8-index finger tip to increase and decrease the volume and brightness.

Function for volume control:

```
def setVolume(dist,frame):
    cv.circle(frame,(xr1,yr1),15,(255,0,255),cv.FILLED)     cv.circle(frame, (xr2, yr2), 15, (255, 0, 255), cv.FILLED)     cv.line(frame,(xr1,yr1),(xr2,yr2),(255,0,255),3)     vol = np.interp(int(dist), [35, 215], [minVolume, maxVolume])
volbar=np.interp(dist,[50,250],[400,150])     volper=np.interp(dist,[50,250],[0,100])
cv.rectangle(frame, (50, 150), (85, 400), (0, 255, 0), 3)     cv.rectangle(frame, (50, int(volbar)), (85, 400), (0, 255, 0), cv.FILLED)
cv.putText(frame,f'{int(volper)}%',(40,450),cv.FONT_HERSHEY_COMPLEX,1,(0,250, 0),3)
    cv.putText(frame, f'RIGHT-VOLUME', (40, 50), cv.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0), 3)
    volume.SetMasterVolumeLevel(vol, None)
```

Function for brightness control:

```
def setBrightness(dist,frame):
    cv.circle(frame, (xr1, yr1), 15, (255, 0, 255), cv.FILLED)
cv.circle(frame, (xr2, yr2), 15, (255, 0, 255), cv.FILLED)     cv.line(frame, (xr1, yr1), (xr2, yr2), (255, 0, 255), 3)
    brightness = np.interp(int(dist), [35, 230], [minBrightness, maxBrightness])
volbar = np.interp(dist, [50, 250], [400, 150])     briper = np.interp(dist, [50, 250], [0, 100])
    cv.rectangle(frame, (50, 150), (85, 400), (0, 255, 0), 3)     cv.rectangle(frame, (50, int(volbar)), (85, 400), (0, 255, 0), cv.FILLED)     cv.putText(frame, f'{int(briper)}%', (40, 450), cv.FONT_HERSHEY_COMPLEX, 1,
(0, 250, 0), 3)             cv.putText(frame, f'LEFT-BRIGHTNESS', (40, 50), cv.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0), 3)
sbc.set_brightness(int(brightness))
```

To capture the video:

```
while True:
```

```
   success,frame = cap.read()     frame = cv.flip(frame, 1)
frame = handlmsObj.findHands(frame,draw=True)
lndmrks = handlmsObj.findPosition(frame, draw=False)
```

To get the landmarks of the Thumb tip and Index finger tip:

```
   if lndmrks:
      # print(lndmrks[4],lndmrks[8])        xr1, yr1
= lndmrks[1][4][1], lndmrks[1][4][2]        xr2, yr2
= lndmrks[1][8][1], lndmrks[1][8][2]
```

To calculate the distance between Thumb tip and Index finger tip:

```
      dist = math.hypot(xr2 - xr1, yr2 - yr1)
```

If the hand is left then brightness function will be called or If the hand is right hand then the volume function will be called:

```
      if lndmrks[0] == 'Left':
         setBrightness(dist,frame)
elif lndmrks[0] == 'Right':
setVolume(dist,frame)
cv.imshow("stream", frame)     if
cv.waitKey(1) & 0xFF == ord('q'):
break
  cv.destroyAllWindows()
```

4.2 Results

The distance between Right hand thumb tip and index finger tip is at maximum

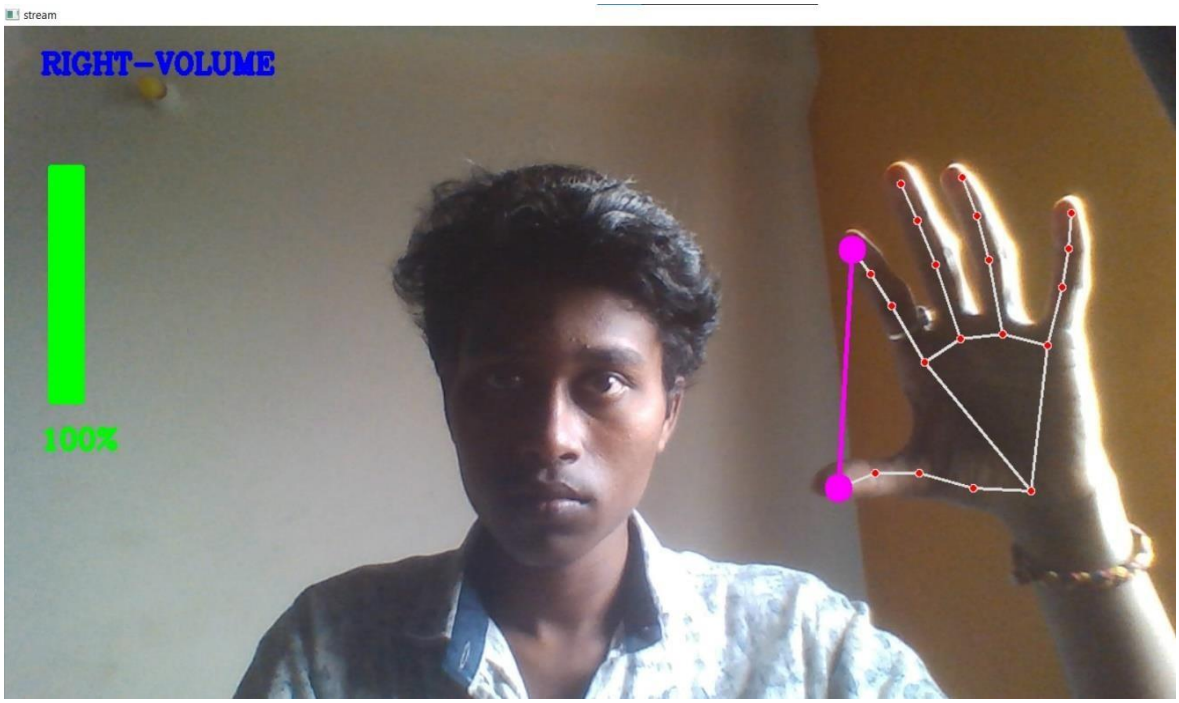


Figure 4.2.0 - Right hand - Volume is at maximum (100%)

The distance between Right hand thumb tip and index finger tip is at minimum

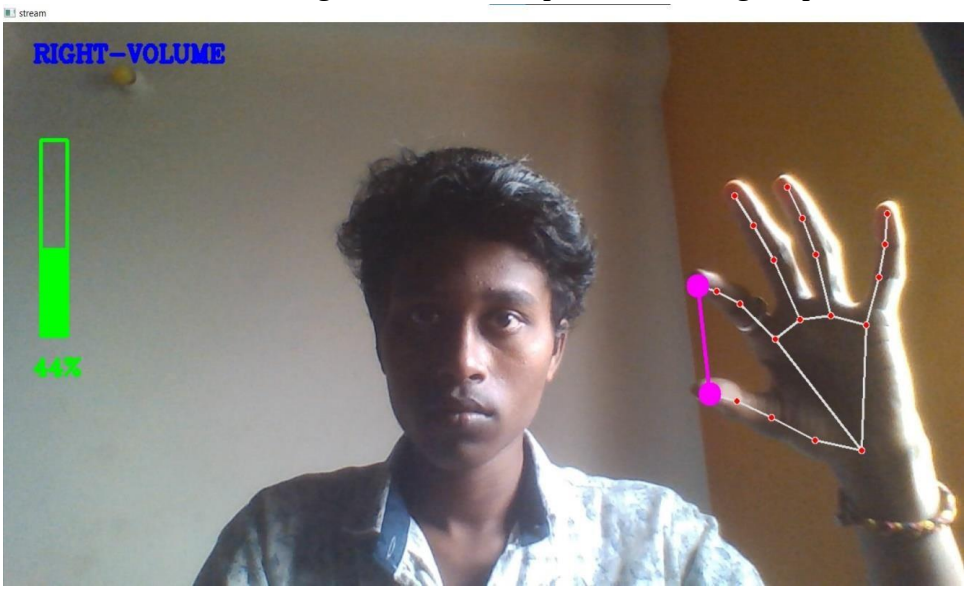


Figure 4.2.1 – Right hand - Volume at minimum (0%)

The distance between Left hand thumb tip and index finger tip is at maximum

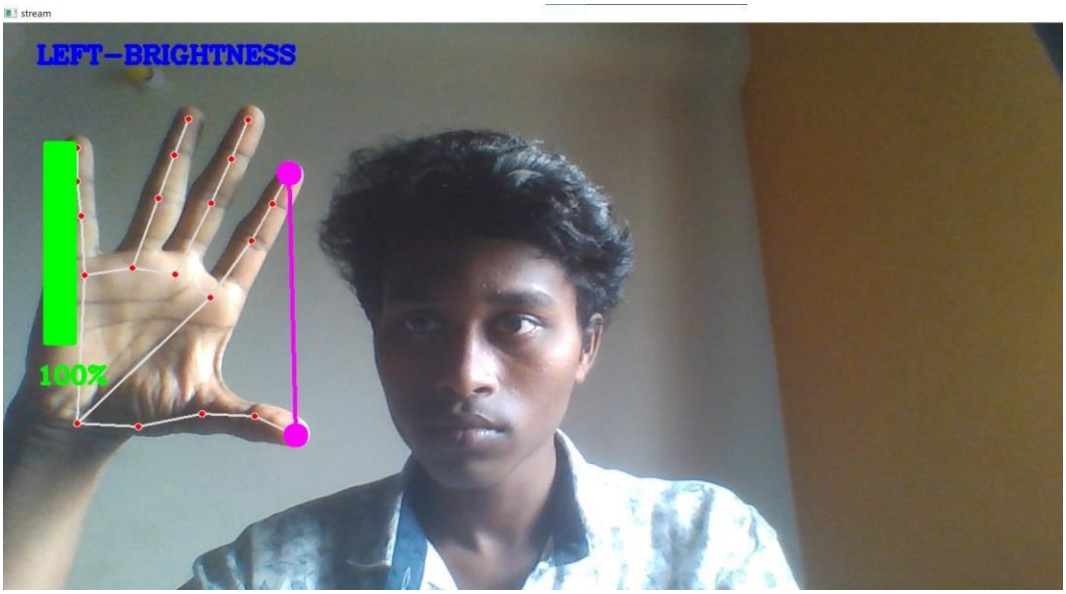


Figure 4.2.3 – Left hand – Brightness at maximum (100%)

The distance between Left hand thumb tip and index finger tip is at minimum

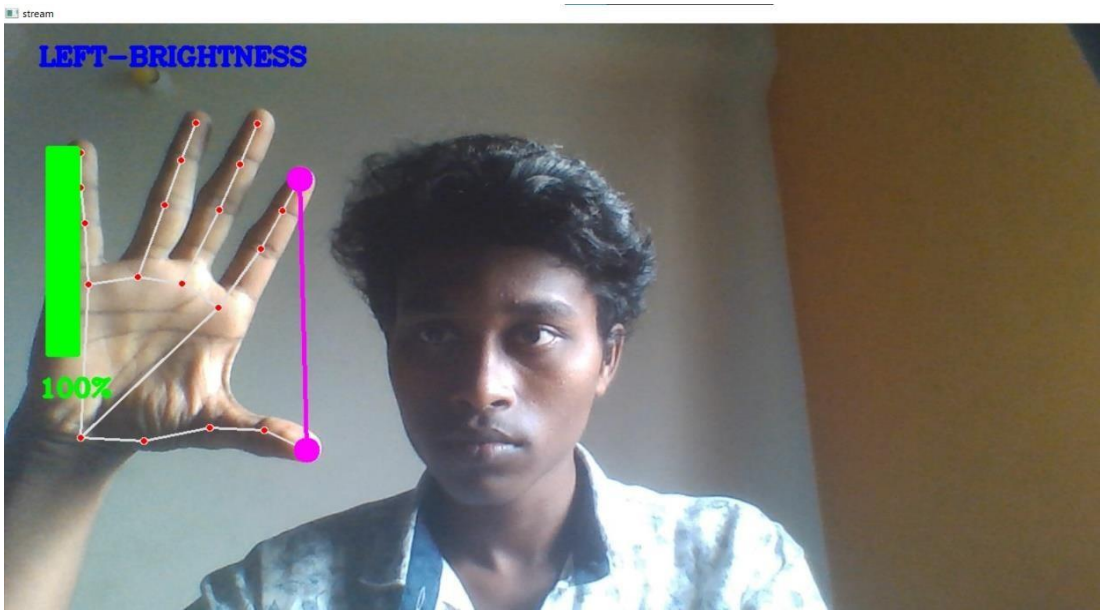


Figure 4.2.4 – Left hand - Brightness at minimum (0%)

CHAPTER-5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

In this project we have planned, designed and implemented the system for Hand gesture recognition system for controlling Volume and brightness using OpenCV and Mediapipe. we coded the program for recognizing the hand gestures and accordingly mapping the identified gestures to specific system operations.

This project showcases a program that enables hand gestures as a practical and simple method of software control. A gesture-based volume and Brightness controller doesn't need any special markers, and it can be used in real life on basic PCs with inexpensive cameras because it doesn't need particularly high-definition cameras to recognize or record the hand movements. The method keeps track of the locations of each hand's index finger and counter tips. This kind of system's primary goal is to essentially automate system components so that they are easy to control. Therefore, using this technique, we have made the concept realistic.

5.2 Future Enhancement

This technique for recognizing hand gestures effectively solves the issue of processing and extracting video frames.

Different hand gestures can be identified and used as computer input in the future. The hand movements used to represent numbers can also be translated in to orders that will carry out relevant action immediately.

Future research can focus on improving the ability to recognize different lighting circumstances and recognize both hands also recognize improper hands for handicap which is a encountered in this project.

In future we would like to improve the accuracy further and add more gestures t

REFERENCES

[1] Devendra Kumar Sharma and Mala Saraswat, "Hand Gesture Detection System" , Advances and Applications in Mathematical Sciences Volume 20, Issue 3, January 2021, Pages 355-360, 2021.

[2.] Munir Oudah ;Ali Al-Naji and Javaan Chahl**,** "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques", Electrical Engineering Technical College, Middle Technical University, Baghdad 10022, Iraq.

[3.] Viraj Shinde; Tushar Bacchav; Jitendra Pawar; Mangesh Sanap, "Hand Gesture Recognition System Using Camera", IJERT**,** Volume 03, Issue 01 (January 2014).

[4.] Salman Shaikh , Raghav Gupta , Imran Shaikh , Jay Borade, "Hand Gesture Recognition Using OpenCV", International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016.

[5.] Soeb Hussain; Rupal Saxena; Xie Han; Jameel Ahmed Khan , " Hand gesture recognition using deep learning" , IEEE, International SoC Design Conference (ISOCC), 2017.

[6.] Akira Utsumi, TsutoniuMiyasato, Fumio KishinoandRyoheiNakatsu, "Real-time Hand Gesture RecognitionSystem," Proc. of ACCV '95, vol. 11, pp. 249-253, Singapore,1995

[7.] Attila Licsár, TamásSzirányi University of Veszprém, "Dynamic Training of Hand Gesture Recognition System" Department of Image Processing andNeurocomputing, H- 8200 Veszpré,2326 Aug. 2004.

[8.] PERALES"Hand tracking and gesture recognition for human-computer interaction",2005.

[9.] Intel Corp, "OpenCV Wiki," OpenCV Library [Online]

[10.] Z. Zhang, Y. Wu, Y. Shan, S. Shafer. Visual panel: Virtual mouse keyboard and 3d controller with an ordinary piece of paper. In Proceedings of Perceptual User Interfaces, 2001

[11.] W. T. Freeman and M. Roth, Orientation histograms for hand gesture recognition. International workshop on automatic face and gesture recognition. 1995, 12: 296- 301.