PYTHON PROGRAMMING LABORATORY

COURSE BASED PROJECT

CONTROLLING A MEDIA PLAYER USING HAND GESTURES.

21071A04K4	ADE VASAVI PATEL
21071A04K5	ADUPA SAHASRA
21071A04K6	AKSHITHA GANJI
21071A07N1	GANJI SHIVANI

TAMASŌMA JYOTIRGAMAYA

Department of Electronics & Communication Engineering

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute

(Approved by AICTE, New Delhi, Govt. of TS and Affiliated to JNTUH)

Accredited by NBA and NAAC with A++ Grade

Vignana Jyothi Nagar, Bachupally, Nizampet (S.O), Hyderabad-500090,
Telangana, India.

ABSTRACT

Gesture-based real-time gesture recognition systems received great attention in recent years because of their ability to interact with systems efficiently through human-computer interaction. Human-Computer Interaction can gain several advantages with the establishment of different natural forms of device-free communication. Gestures are a natural form of action that we often use in our daily lives to interact, so to use them as a way of communicating with computers generates a new paradigm of computing interaction. This project implements computer vision and gesture recognition techniques and develops a vision based low-cost input software for controlling the media player through gestures.

INTRODUCTION

Nowadays, gesture recognition plays an important part in the interaction between humans and computers. To facilitate simple yet user-friendly communication between humans and computers hand Gestures can be used which enable us humans to interact with machines without having to use devices like keyboards, laser pens, etc. In the proposed system, users can use four simple gestures to control the Media Player without physically touching the PC. Gesture is a symbol of physical behavior or emotional expression. It includes body gesture and hand gesture. It falls into two categories: static gesture and dynamic gesture. For the former, the posture of the body or the gesture of the hand denotes a sign. For the latter, the movement of the body or the hand conveys some messages. Gesture can be used as a tool of communication between computer and human. It is greatly different from the traditional hardware-based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user intent through the recognition of the gesture or movement of the body or body parts.

In the past decades, many researchers have strived to improve the hand gesture recognition technology. Hand gesture recognition has great value in many applications such as sign language recognition, augmented reality (virtual reality), sign language interpreters for the disabled, and robot control.

PROBLEM STATEMENT

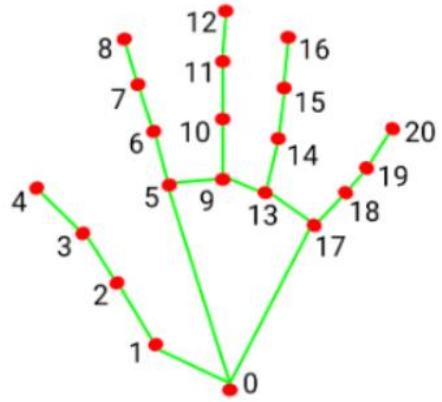
The main aim of this project is to develop an interface between the system and its environment such that the system could identify particular colours to take input as a referral point that can interact with the system to perform some simple tasks such as controlling media player and manipulate its various functions.

APPROACH

Hand gestures
The approach used for hand gesture detection is that you don't need to collect tons of images to train your model, since you rather use landmarks as model inputs.
The workflow is as follows:

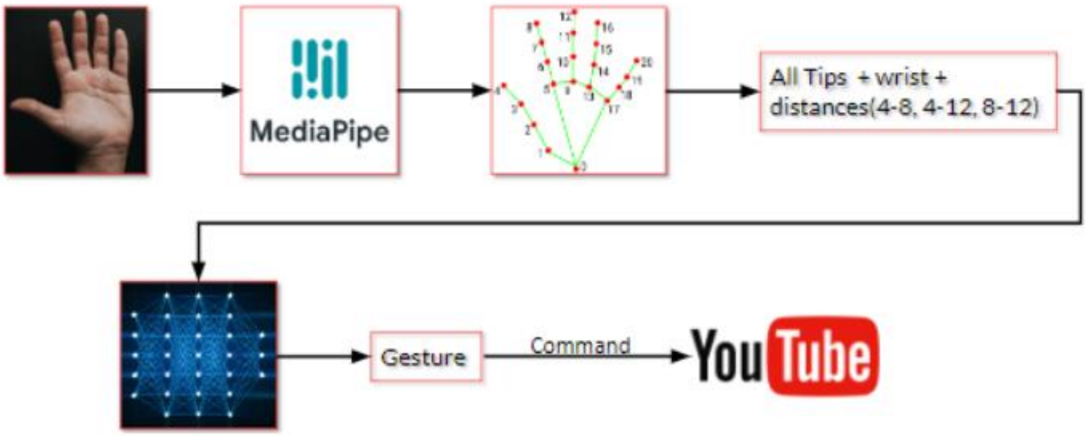
We extracted 2D coordinates from Media Pipe's hand detector. This detector norm ally outputs 21 3D landmarks; the image below shows all the key points.
These new points were then flattened and normalized by the maximum absolute va lue. Also, I computed the distances between key points 4, 8 and 12. Those distance

s were also normalized by the distance between points 0 and 5.



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

Both normalized coordinates and distances were then joined together to formed our feature space, then saved, together with the target, for subsequent training.



LIBRARIES USED:

MEDIAPIPE

MediaPipe is a Framework for building machine learning pipelines for processing t ime-series data like video, audio, etc. This cross-platform Framework works on De sktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano.

MediaPipe powers revolutionary products and services we use daily. Unlike power-hungry machine learning Frameworks, MediaPipe requires minimal resources. It is so tiny and efficient that even embedded IoT devices can run it. In 2019, MediaPipe opened up a new world of opportunity for researchers and developers following its public release.

Pyautogui

Python pyautogui library is an automation library that allows mouse and keyboard control. Or we can say that it facilitates us to automate the movement of the mouse and keyboard to establish the interaction with the other application using the Python script. It provides many features, and a few are given below.

- We can move the mouse and click in the other applications' window.
- We can send the keystrokes to the other applications. For example - filling out the form, typing the search query to browser, etc.
- We can also take snapshots and give an image.
- It allows us to locate a window of the application, and move, maximize, minimize, resizes, or close it.
- Display alert and message boxes.

Opencv

OpenCV is a cross-platform library using which we can develop real-time **computer vision applications**. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

USE CASES OF GESTURE RECOGNITION:

Gesture recognition can be used to control devices or interfaces, such as a computer or a smartphone, through movements or actions, such as hand or body movements, facial expressions or even voice commands.

Gesture recognition has a variety of uses, including:

- **Human-computer interaction:** Gesture recognition can be used to control computers, smartphones, and other devices through gestures, such as swiping, tapping, and pinching.
- **Gaming:** Gesture recognition can be used to control characters and objects in video games, making the gaming experience more immersive and interactive.
- **Virtual and augmented reality:** Gesture recognition can be used to interact with virtual and augmented reality environments, allowing users to control and manipulate objects in those environments.
- **Robotics:** Gesture recognition can be used to control robots, allowing them to perform tasks based on the user’s gestures.
- **Sign language recognition:** Gesture recognition can be used to recognize and translate sign language into spoken or written language, helping people who are deaf or hard of hearing communicate with others.
- **Automotive:** Gesture recognition can be used in cars to control various functions such as radio, AC, and navigation systems.
- **Healthcare:** Gesture recognition can be used in rehabilitation of patients with physical disabilities.

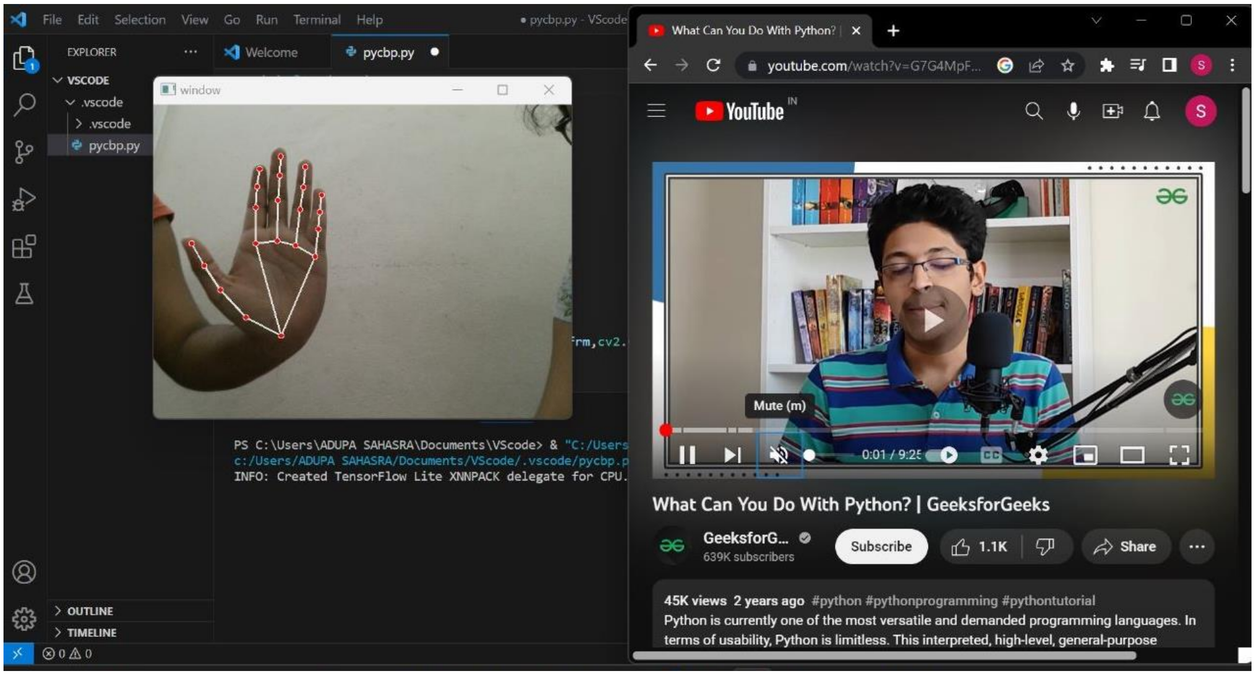
CODE:

```
import cv2
import mediapipe as mp
import pyautogui
import time

def count_fingers(lst):
    cnt=0
    thresh=(lst.landmark[0].y*100-lst.landmark[9].y*100)/2

    if(lst.landmark[5].y*100-lst.landmark[8].y*100)>thresh:
        cnt=cnt+1
    if(lst.landmark[9].y*100-lst.landmark[12].y*100)>thresh:
        cnt=cnt+1
    if(lst.landmark[13].y*100-lst.landmark[15].y*100)>thresh:
        cnt=cnt+1
    if(lst.landmark[17].y*100-lst.landmark[20].y*100)>thresh:
        cnt=cnt+1
    if(lst.landmark[5].x*100-lst.landmark[4].x*100)>5:
        cnt=cnt+1
    return cnt


cap=cv2.VideoCapture(0)

drawing=mp.solutions.drawing_utils
hands=mp.solutions.hands
hand_obj=hands.Hands(max_num_hands=1)

start_init=False
prev=-1
while True:
    end_time=time.time()
    _, frm=cap.read()
    frm=cv2.flip(frm,1)


    res=hand_obj.process(cv2.cvtColor(frm,cv2.COLOR_BGR2RGB))
```

```
  if res.multi_hand_landmarks:

        hand_keyPoints=res.multi_hand_landmarks[0]
        cnt=count_fingers(hand_keyPoints)

        if not(prev==cnt):

            if not(start_init):
                start_time=time.time()
                start_init=True
            elif(end_time-start_time)>0.2:
                if cnt==1:
                    pyautogui.press("right")
                elif cnt==2:
                    pyautogui.press("left")
                elif cnt==3:
                    pyautogui.press("up")
                elif cnt==4:
                    pyautogui.press("down")
                elif cnt==5:
                    pyautogui.press("space")

                prev=cnt
                start_init=False

        drawing.draw_landmarks(frm,hand_keyPoints,
hands.HAND_CONNECTIONS)
    cv2.imshow("window",frm)
    if cv2.waitKey(1)==27:
        cv2.destroyAllWindows()
        cap.release()
        break
```

OUTPUT:

1. Play the video OR Pause the video

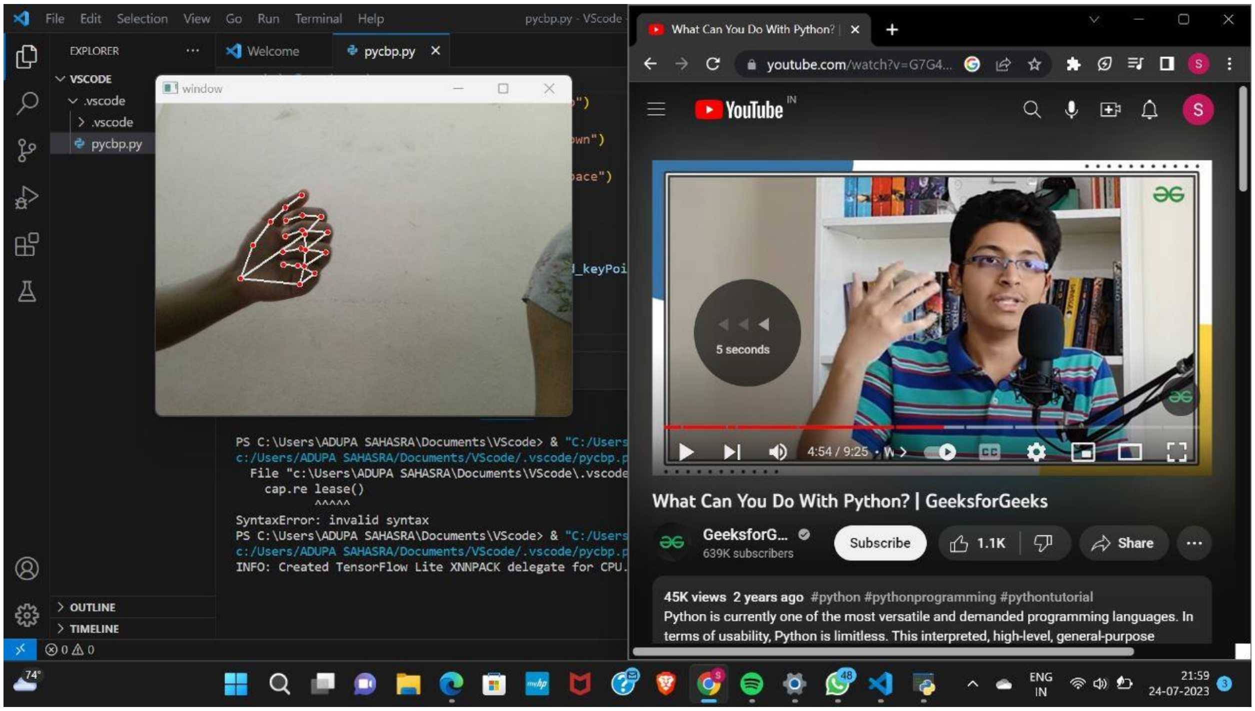


```
PS C:\Users\ADUPA SAHASRA\Documents\VScode> & "C:/Users
c:/Users/ADUPA SAHASRA/Documents/VScode/.vscode/pycbp.p
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

What Can You Do With Python? | GeeksforGeeks

45K views 2 years ago #python #pythonprogramming #pythontutorial
Python is currently one of the most versatile and demanded programming languages. In terms of usability, Python is limitless. This interpreted, high-level, general-purpose

2. Backward the video for 5 seconds

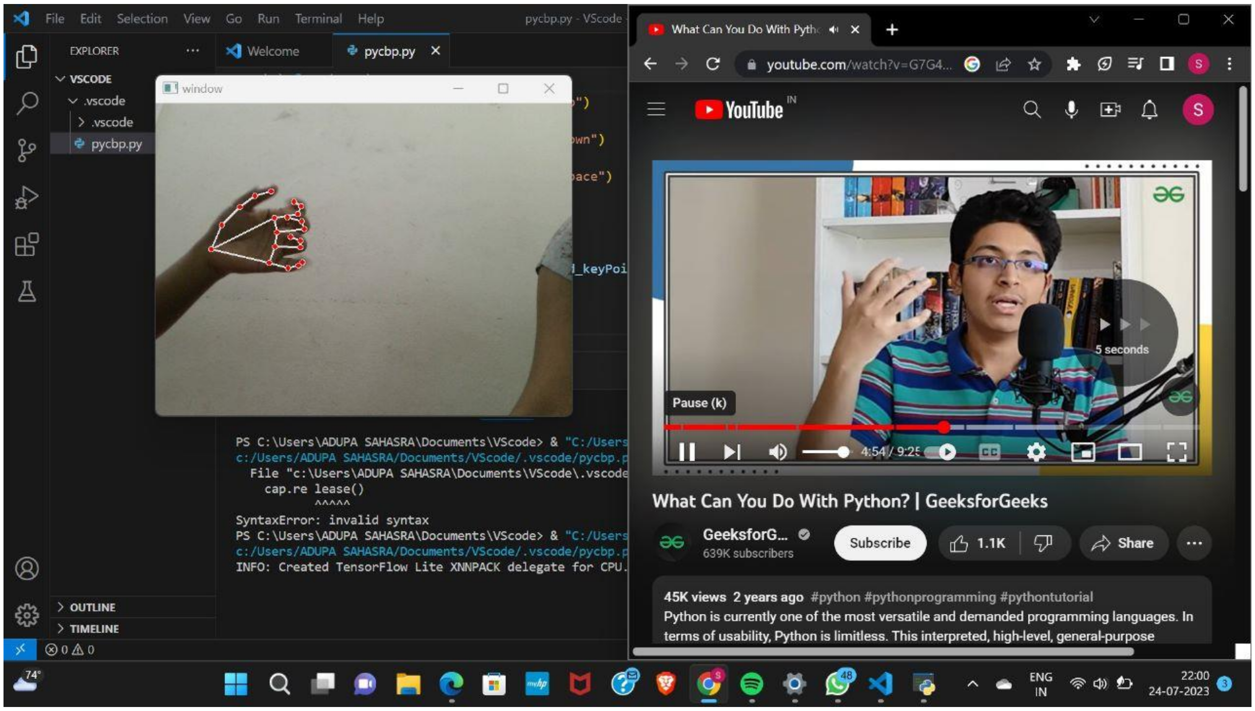


```
PS C:\Users\ADUPA SAHASRA\Documents\VScode> & "C:/Users
c:/Users/ADUPA SAHASRA/Documents/VScode/.vscode/pycbp.p
  File "c:\Users\ADUPA SAHASRA\Documents\VScode\.vscode
    cap.re lease()
           ^^^^^
SyntaxError: invalid syntax
PS C:\Users\ADUPA SAHASRA\Documents\VScode> & "C:/Users
c:/Users/ADUPA SAHASRA/Documents/VScode/.vscode/pycbp.p
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

What Can You Do With Python? | GeeksforGeeks

45K views 2 years ago #python #pythonprogramming #pythontutorial
Python is currently one of the most versatile and demanded programming languages. In terms of usability, Python is limitless. This interpreted, high-level, general-purpose

3. Forward the video for 5 seconds



```
PS C:\Users\ADUPA SAHASRA\Documents\VScode> & "C:/Users
c:/Users/ADUPA SAHASRA/Documents/VScode/.vscode/pycbp.p
  File "c:\Users\ADUPA SAHASRA\Documents\VScode\.vscode
    cap.re lease()
          ^^^^^
SyntaxError: invalid syntax
PS C:\Users\ADUPA SAHASRA\Documents\VScode> & "C:/Users
c:/Users/ADUPA SAHASRA/Documents/VScode/.vscode/pycbp.p
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

What Can You Do With Python? | GeeksforGeeks

45K views 2 years ago #python #pythonprogramming #pythontutorial
Python is currently one of the most versatile and demanded programming languages. In terms of usability, Python is limitless. This interpreted, high-level, general-purpose

CONCLUSION:

With the advancements in technology to provide novel, convenient and fast methods of human computer interaction, gesture recognition has received wide appreciation. The various existing systems have good working features but have not been well received by customers. The main problem lies in the fact that these systems have low accuracy rates and complex algorithms. The proposed system will aim to combat these issues and stand out in the crowd of gesture recognition systems.

It will provide a touchless user interface for controlling multimedia files and applications such as video players and music players. It will act as a helping aid for manipulation of systems for people who have disabilities, who cannot access their input devices or anyone who prefers this more natural method of communication compared to other methods.

The gesture would function as the direct command for operations consisting of play or pause the video based on the person's gestures onto the display. The Hand

Gesture recognition is moving at incredible velocity for the futuristic services and products and main businesses are developing technology-based at-hand gesture devices.

REFERENCES:

1. https://www.ijert.org/controlling-media-player-with-hand-gestures
2. https://ijrpr.com/uploads/V3ISSUE5/IJRPR4107.pdf
3. https://wjarr.com/sites/default/files/WJARR-2022-0565.pdf
4. https://pypi.org/project/mediapipe/
5. https://www.ijraset.com/research-paper/media-control-using-hand-gestures