A

PROJECT REPORT

ON

MINI PROJECT

ENTITLED

“Finger Counter”

SUBMITTED BY

Singh Piyush R. (216090307096)

Hariyani Kavy D. (216090307084)

Chudasama Harshrajsinh R. (216090307134)

To

COMPUTER ENGINEERING DEPARTMENT

C U SHAH GOVT. POLYTECHNIC – SURENDRANAGAR

GUJARAT TECHNOLOGICAL UNIVERSITY – AHMEDABAD

November – 2023

C.U. SHAH GOVERNMENT POLYTECHNIC - WADHWAN

COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Mini Project entitled “***Finger Counter***” has been carried out by **Singh Piyush R. (216090307084)** under my guidance in partial fulfillment of the degree of Diploma Engineering in **COMPUTER ENGINEERING** 5th Semester of Gujarat Technological University, Ahmedabad during the academic year 2022-23.e

Guides

Mr. J.N Solanki	Mr. K.G.Patel
Internal Guide	Head of Department

C.U. SHAH GOVERNMENT POLYTECHNIC - WADHWAN

COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Mini Project entitled “***<u>Finger Counter</u>***” has been carried out by **<u>Hariyani Kavy D. (216090307096)</u>** under my guidance in partial fulfillment of the degree of Diploma Engineering in **COMPUTER ENGINEERING** 5th Semester of Gujarat Technological University, Ahmedabad during the academic year 2022-23.

Guides

Mr. J.N Solanki	Mr. K.G.Patel
Internal Guide	Head of Department

C.U. SHAH GOVERNMENT POLYTECHNIC - WADHWAN

COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Mini Project entitled "***Finger Counter***" has been carried out by **Chudasama Harshrajsinh R, (216090307096)** under my guidance in partial fulfillment of the degree of Diploma Engineering in **COMPUTER ENGINEERING** 5th Semester of Gujarat Technological University, Ahmedabad during the academic year 2022-23.

Guides

Mr. J.N Solanki	Mr. K. G. Patel
Internal Guide	Head of Department

Introduction to Project Introduction to Project

The "Finger Counter" is an innovative computer vision project developed in Python. Its primary objective is to recognize and count the number of fingers that are presented in front of a camera or within a video stream. By utilizing advanced computer vision techniques, this project has the ability to analyze live or pre-recorded video input and provide real-time feedback regarding the number of fingers detected.

In a world increasingly intertwined with technology, human-computer interaction stands at the forefront of innovation. From voice commands to touchscreens, the way we communicate with our devices has evolved significantly. The Finger Counter project represents a leap forward in this domain, promising to transform the manner in which we interact with computers.

In an era of advanced technology, why develop a Finger Counter in Python? The project's purpose is multifold. It's about making computing more accessible, especially in educational and inclusive settings. It's about breaking language barriers and providing a universal means of communication. The Finger Counter has the potential to empower diverse user groups, from young students learning math to individuals with hearing impairments who rely on sign language.

Key to the Finger Counter's functionality is computer vision and machine learning. Computer vision, a branch of artificial intelligence, empowers computers to interpret visual information, while machine learning enables the system to understand and count fingers. It involves data, training processes, and the use of neural networks. These technologies come together to create a powerful tool for finger counting.

Working of Project

Video Capture: The code starts by capturing a video feed from the default camera (index 0). The width and height of the video feed are set to 640x480 pixels.

Loading Hand Sign Images: The code assumes there is a folder located at "E:\Internship Project\FingerCount" that contains images of hand signs representing numbers (e.g., 1 to 5). These images are loaded into the overlayList for later use in overlaying on the video feed.

Hand Detection: The code uses an external module called handtrackingmodule for hand detection and landmark (finger position) tracking. It is not included in the provided code, but it's assumed to provide the necessary functions for detecting and tracking h ands.

Finger Counting: The code determines the number of fingers displayed by analyzing the positions of specific landmarks on the hand. The tipIds list contains the landmark IDs for the tips of the thumb and fingers.

For each finger (except the thumb), the code checks if the Y-coordinate of the tip landmark is higher than the Y-coordinate of the landmark just below it. If it is, the finger is considered open (1); otherwise, it's closed (0).

Overlaying Hand Signs: Based on the number of open fingers, the code selects the corresponding hand sign image from the overlayList. It overlays this image onto the video feed to represent the finger count.

Displaying Finger Count: The code draws a green rectangle at the top left corner of the video feed and displays the current finger count in red inside the rectangle. The finger count is determined by counting the number of "1" values in the fingers list.

Frame Rate Display: The frames per second (FPS) of the video feed are calculated and displayed at the top right corner of the window.

Terminating the Application: The application can be closed by pressing the "q" key, which is checked by the cv2.waitKey function.

Code of Project

```
import cv2
import os
import time
import handtrackingmodule as htm
wCam, hCam = 640, 480
cap = cv2.VideoCapture(0)
# width of cam
cap.set(3, wCam)
cap.set(4, hCam)

# to store the finger through os library
folderPath = "E:\Internship Project\FingerCount"
myList = os.listdir(folderPath)
print(myList)
overlayList = []
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    # print(f'{folderPath}/{imPath}')
    overlayList.append(image)
 print(len(overlayList))

pTime = 0

detector = htm.handDetector(detectionCon=0.85)
tipIds = [4, 8, 12, 16, 20]
while True:
    success, img = cap.read()
    img = cv2.flip(img, 1)
    img = detector.findHands(img)
    lmList = detector.findPosition(img, draw=False)
    # print(lmList)

    if len(lmList) != 0:
        fingers = []
            # it is for thumb
        if lmList[tipIds[0]][1] < lmList[tipIds[0]-1][1]:
            fingers.append(1)
        else:
            fingers.append(0)
        # it is for 4 finger
        for id in range(1, 5):
            if lmList[tipIds[id]][2] < lmList[tipIds[id] - 2][2]:
                fingers.append(1)
            else:
                fingers.append(0)

        # print(fingers)
        # it shows that how many numbers are 1 in the given list fingers
        totalFingers = fingers.count(1)
        print(totalFingers)
                # print("index finger open")

        # if the size is unknown and all the images size are different then shape
    # method is used which return the value of the height , width and
channel
    h, w, c = overlayList[totalFingers-1].shape
    img[0:h, 0:w] = overlayList[totalFingers-1]

    cv2.rectangle(img, (20, 225), (170, 425), (0,255,0))
    cv2.putText(img, str(totalFingers), (45, 375),
cv2.FONT_HERSHEY_PLAIN
            , 10, (255,0 ,0 ), 25)

  cTime = time.time()
  fps = 1/(cTime-pTime)
  pTime = cTime


  cv2.putText(img, f'FPS:{int(fps)}', (400, 70),
cv2.FONT_HERSHEY_PLAIN,
          3, (255,0, 0), 3)

  cv2.imshow("Image", img)
  if cv2.waitKey(1) & 0xFF == ord("q"):
    break
```

Screenshots

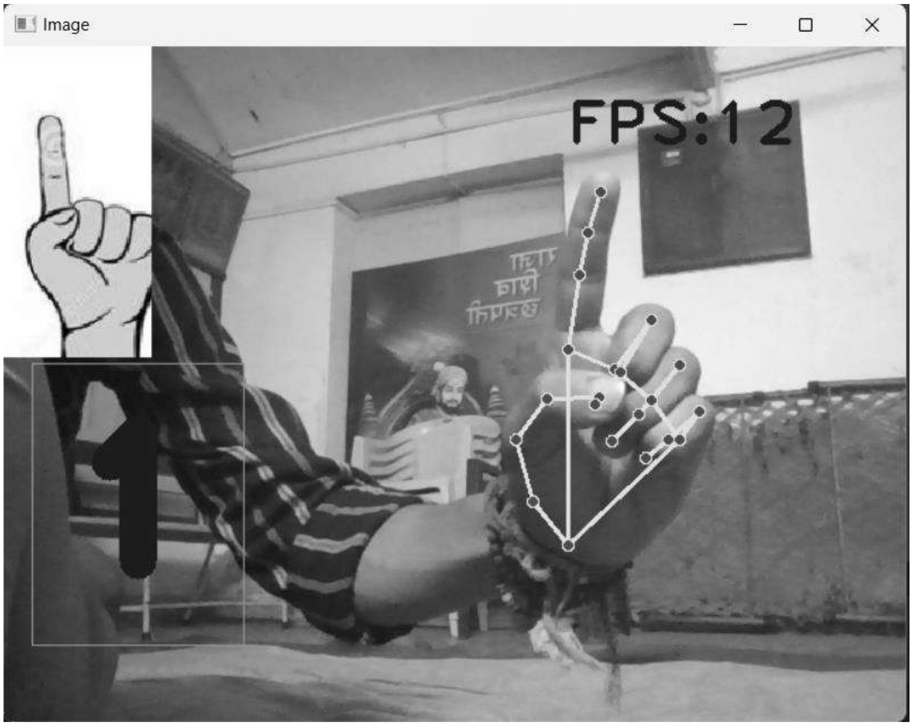
Image
FPS:12
1

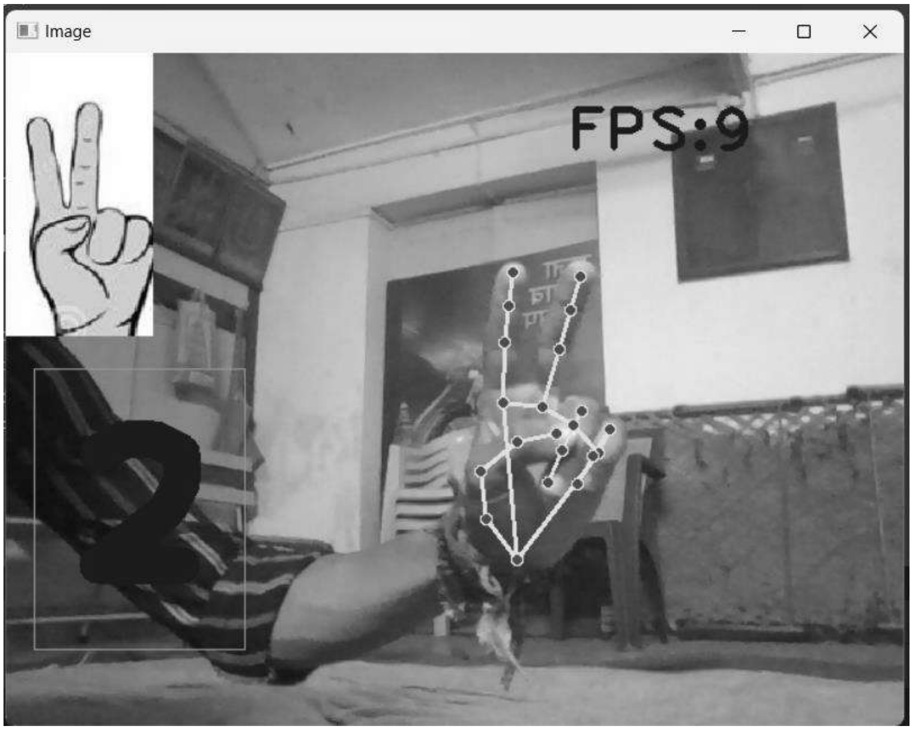
Image
FPS:9
2