Department of Artificial Intelligence & Machine Learning

Academic Year 2021-22(EVEN)

Report

for

Mini project-II (20AIM48A)

On

"Sign Language Recognition"

By

Name	USN
Priya N	**1NH20AI080**
Kruthik M	**1NH20AI137**

Under the Guidance of

Mrs. KavyaShree C

Senior Assistant professor,

Dept. of Artificial Intelligence & Machine Learning,

New Horizon College of Engineering,

Bangalore-560103

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

Department of Artificial Intelligence & Machine Learning

CERTIFICATE

Certified that the Mini Project- II with the subject code 20AIM48A work entitled **"Sign Language Recognition"** carried out by Priya N USN 1NH20AI080, Kruthik M USN 1NH20AI137. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work.

Prof. Kavyashree C	**Dr. N V Uma Reddy**
Internal Guide	Head of Department

External Viva

Examiner | Signature with date:

1.

2.

ABSTRACT

Sign Language Recognition is the most essential tool as sign language is the only means of communication for those who are stone deaf. Hand gesture recognition systems have advanced rapidly in recent years, owing to their capacity to successfully collaborate with machines. In a virtual environment, gestures are seen to be the most natural way for humans and computers to communicate. Our project aims to analyze the various researches that have been published on intelligent systems in sign language recognition over the past two decades. To more easily approach the problem and obtain the desired results we experimented with just up to 6 different numbers that is 0 to 5 by creating our own dataset. To extract the hand region in our system, we employed background subtraction. Our PC's camera records a live video in this application, from which a preview is taken using its functions or activities. Then our system gives the output in terms of those numbers that we our showing through our hand. The background does not have to be completely black for the system to function. Any background can be used.

Table of Contents

ACKNOWLEDGEMENT

We have put a lot off effort in this project. However, we would not have been able to complete this project if not for the support of our teachers and parents. We would like to extend our sincere thanks to all of them.

We would like to thank Dr Manjunatha, Principal New Horizon College of Engineering to provide us this opportunity to showcase our skills through the medium of this project .

We also thank Dr. NV Uma Reddy, H.O.D AIML department for providing us facilities and constant motivation during the course of the project.

We owe our sincere thanks and gratitude to Senior Assistant Professor Mrs Kavyashree C ,for her valuable guidance throughout the course of the project.

CHAPTER-1

INTRODUCTION

1.1 Introduction

Artificial intelligence and deep learning can be used to aid the cause of deaf and hard-of-hearing individuals by developing tools that can recognize and interpret sign language. This can be very helpful in communicating with others as it is not something that is commonly taught to everyone.

In the community of the deaf and dumb, sign language is the main form of communication. It is mostly used by their families and/or the deaf and dumb population because the average person cannot understand the syntax or meaning of the numerous gestures that make up sign language.

Making these people feel included in society in today's technologically advanced world requires facilitating their communication. This means that an intelligent computer system needs to be created and trained.

For a while now, researchers have been working on the issue, and the outcomes appear promising. Although interesting technologies are being developed for speech recognition, the industry currently lacks any genuine commercial products for sign recognition.

We would demonstrate a reliable and effective method of sign language detection in our project. We would perform the sign language detection using image processing rather than Data gloves.

The key benefit of employing image processing over data gloves is that whenever a new user logs on, the system does not need to be re-calibrated. This technique is not limited to usage with a black or white background and may be used with any background.

1.2 Objectives

The objective of the project Sign Language Recognition (SLR) is to provide an efficient and accurate method to convert sign language to text or voice. Primary goal of the project is to create systems which will be able to identify specific gestures and use them to convey information.

Although sign language is the most natural way of exchanging information among deaf people it has been observed that they are facing difficulties to interact with normal people, therefore our project SLR systems is built to provide an efficient and accurate way to convert sign language into text or voice.

Objective of our project is not only to assist deaf people but also enable young children to interact with the computer. Through our work children will be able to learn numbers. To simplify the use of sign language. The other objective is to make the result cost effective.

Also the goal of this project is to create a system that will allow stone deaf people to effectively communicate with everyone else by using their natural motion. We frequently use hand gestures to convey information because they are a form of nonverbal communication that allows us to express ourselves freely.

1.3 Literature Survey

Reference	Methodology	Outcome/Results
Ciresan, D., Meier, U., Schmidhuber	Multi-column deep neural networks for image classification.	Identification of images in the data set.
R. B. Abdessalem, S. Nejati, L. C. Briand, and T. Stifter	**Testing vision-based control systems using learnable evolutionary algorithm**	to detect feature interaction failures by casting this problem into a search-based test generation problem, which is able to identify more than twice as many feature interaction failures as two baseline test objectives used in the software testing literature
J. Singha, A. Roy, and R. H. Lashkar	**Dynamic hand gesture recognition using vision-based approach for human–computer interaction.**	To track the Neural Computing and Applications.
Y. Guo, Q. Wang, S. Huang, and A. Abraham	**Flexible Neural Trees for Online Hand Gesture Recognition using surface Electromyography.**	Evaluating the functional status of hands to assist in hand gesture recognition in many fields of treatment and rehabilitation.
Luong Chi Mai	Image pre-processing in python	To read the image as input.
W. Zhou, H. Li, and Q. Tian	Content-based Image Retrieval	Region of interest-based image retrieval techniques

1.4 Existing System

It features a variety of sensors that can recognize finger bending in addition to a variety of motion tracking devices, including magnetic tracking devices and inertial tracking devices. The glove and accompanying software forecast the movements. The existing systems we have the dataset in such a way that it more diverse enough in order to accommodate people with all skin tones and in all environment. Therefore, it uses large amount of dataset.

The existing system uses complicated way to approach the problem. Using of datagloves makes the project expensive. It uses methods where the background is also not segregated. The best virtual reality experience is provided by this ground-breaking flex sensor technology, but it is not practical to utilize it in daily life to interact with computers because of the expensive cost of high-end sensors.

1.5 Proposed System

The learning aids for specially enabled people exists but usage of these aids are very limited. The proposed system would be a real time system wherein live sign gesture would be processed using image processing. Although sign language is the most natural way of exchanging information among deaf people and has been observed that they are facing difficulties in interacting with normal people.

Our project will not only assist deaf people but would also enable very young children to interact with the computer. Our project proposes a camera based assistive sign detection. Here we are using six sign recognitions. We can use various classifiers to differentiate various numbers i.e signs and translate the output in terms for text. We will be using various machine learning algorithm to train on the dataset.

The existing systems have been able to recognize gesture with high latency as it uses only image processing. We work on using hand gestures for the interfacing. In our system we aim

to develop a cognitive system and would be able to give response in such a way that the numbers can be recognized easily.

It is economical as we are not using any additional datagloves for sign language detection. As we use threshold value while converting the image from grayscale to binary form this system need not to be used only in black and white background, it can be used in any background. It is easy to use and needs no prior training to know how to operate.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

“Sign language recognition” the project we have done is a complete software application due to which hardware requirement is minimal.

Processor: intel i5 or AMD Ryzen 5 and above

Ram:8gb or above

Input devices: standard keyboard and mouse

Output devices: high resolution moniter

2.2 Software Requirement

Operating system: windows 10 or above

Programming language: python 3.10

Code editor: juypter notebook

Python additional library Media Pipe, cv2, time, os, hand tracker

OpenCV (Open-Source Computer Vision) a library aimed at present-time computer vision.

CHAPTER 3

SYSTEM DESIGN

3.1 System Architecture

System architecture comprehend the overall working process of the system. From the start that is opening of the webcam till the end if recognizing the number of fingers. It gives the overall flow of testing the system.

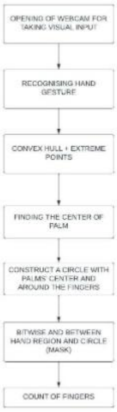

Figure 3.2.1:System architecture depicting the process.

3.2 Algorithms/Flow Charts

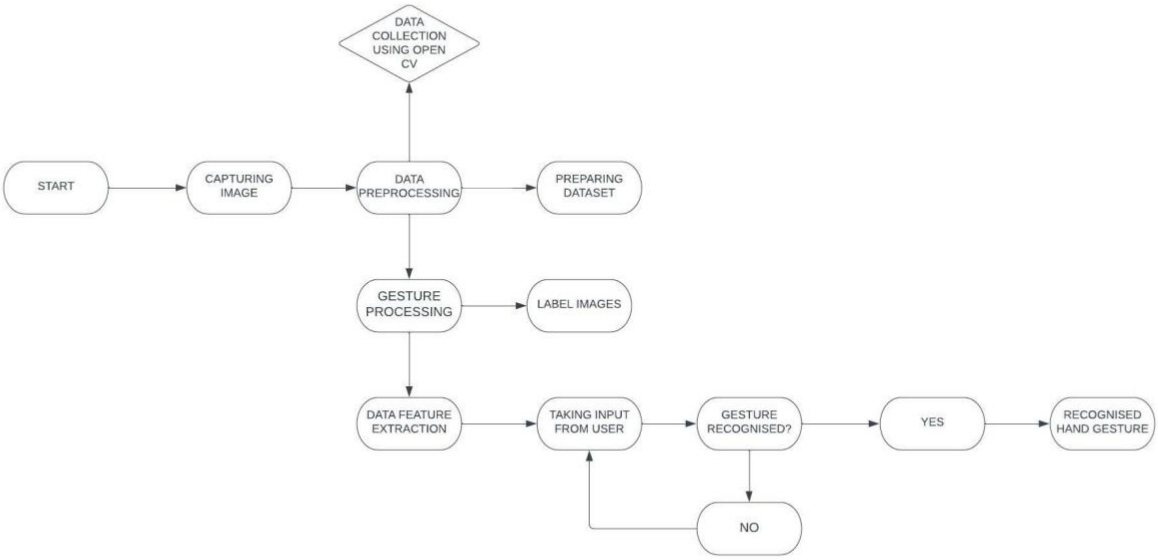

Figure 3.2.2: Flowchart depicting the process of our program

CHAPTER 4

IMPLEMENTATION

4.1 Pseudocode

STEP 1

-Import necessary packages

STEP 2

- Capture the copy from the camera and to define a region of interest.

STEP 3

- Declare the "Media Pipe" objects as well as the coordinates for the thumb and finger.

STEP 4

- Feed the dataset by capturing images.

STEP 5

-Convert the dataset by labeling each image.

-Each image is labelled under specific data or specific sign.

STEP 6

-After the insertion of code run the code.

STEP 7

-A sign is placed in front of the webcam.

- Which is when the result is displayed in real-time on the screen.

Code 1:

```
import cv2
import mediapipe as mp
import time


class handDetector():
    def __init__(self, mode=False, maxHands=2,modelComplexity=1, detectionCon=0.5,
trackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.modelComplex = modelComplexity
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands, self.modelComplex,
                                        self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils

    def findHands(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.hands.process(imgRGB)
        # print(results.multi_hand_landmarks)

        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

    def findPosition(self, img, handNo=0, draw=True):
```

```
lmList = []
if self.results.multi_hand_landmarks:
    myHand = self.results.multi_hand_landmarks[handNo]
    for id, lm in enumerate(myHand.landmark):
        # print(id, lm)
        h, w, c = img.shape
        cx, cy = int(lm.x * w), int(lm.y * h)
        # print(id, cx, cy)
        lmList.append([id, cx, cy]
```

```
                if draw:
                    cv2.circle(img, (cx, cy), 15, (255, 0, 255), cv2.FILLED)

        return lmList



def main():
    pTime = 0
    cTime = 0
    cap = cv2.VideoCapture(0)
    detector = handDetector()
    while True:
        success, img = cap.read()
        img = detector.findHands(img)
        lmList = detector.findPosition(img)
        if len(lmList) != 0:
            print(lmList[4])

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

```
        cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 255), 3)

        cv2.imshow("Image", img)
        cv2.waitKey(1)


if __name__ == "__main__":
    main()
```

code 2:

```
import cv2
import time
import os
import HandTracker as htm

wCam, hCam = 640, 480

cap = cv2.VideoCapture(0)
cap.set(3, wCam)
cap.set(4, hCam)
```

```
folderPath = "FingerImages"
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=0.75)

tipIds = [4, 8, 12, 16, 20]

while True:
    success, img = cap.read()
    img = detector.findHands(img)
    lmList = detector.findPosition(img, draw=False)
    # print(lmList)
```

```
if len(lmList) != 0:
    fingers = []

    # Thumb
    if lmList[tipIds[0]][1] > lmList[tipIds[0] - 1][1]:
        fingers.append(1)
    else:
        fingers.append(0)

    # 4 Fingers
    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)
    totalFingers = fingers.count(1)
    print(totalFingers)

    #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.FILLED)
        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)
    cv2.waitKey(1)
```

4.2 Result

When we run the code with required datasets, we will be getting the below given result.

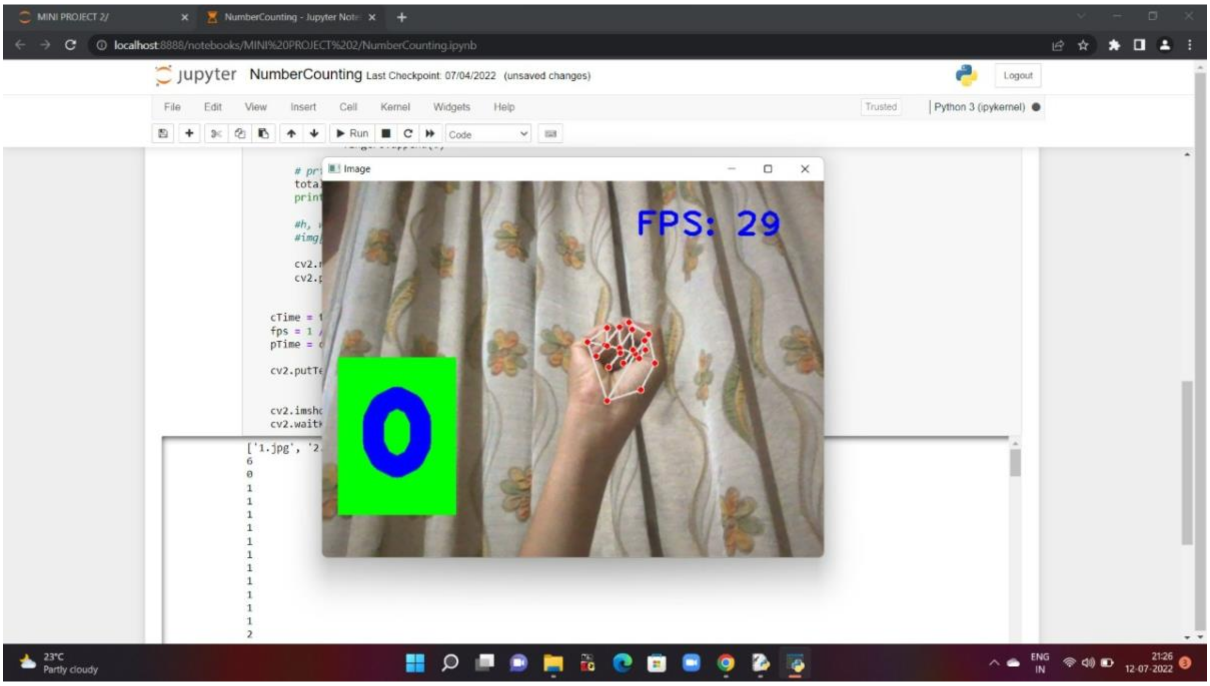

Figure 4.2.1 The output when the sign is recognized as 0 in real-time shown on the screen.

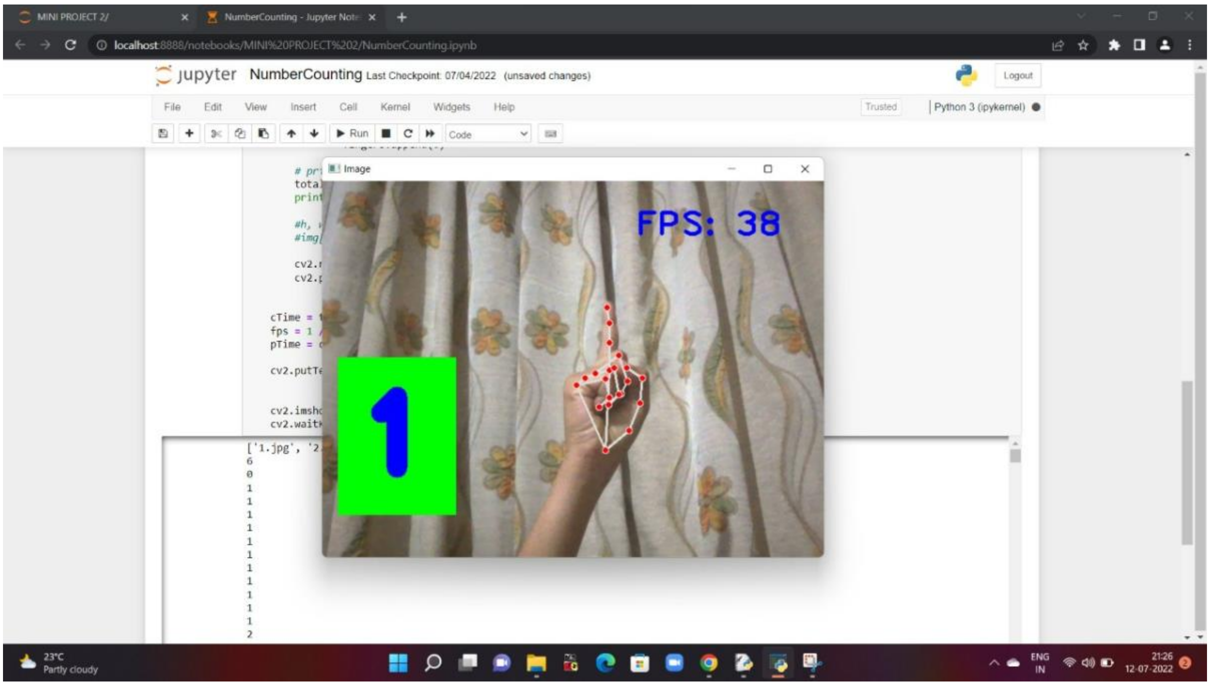

Figure 4.2.2 The output when the sign is recognized as 1 in real-time shown on the screen

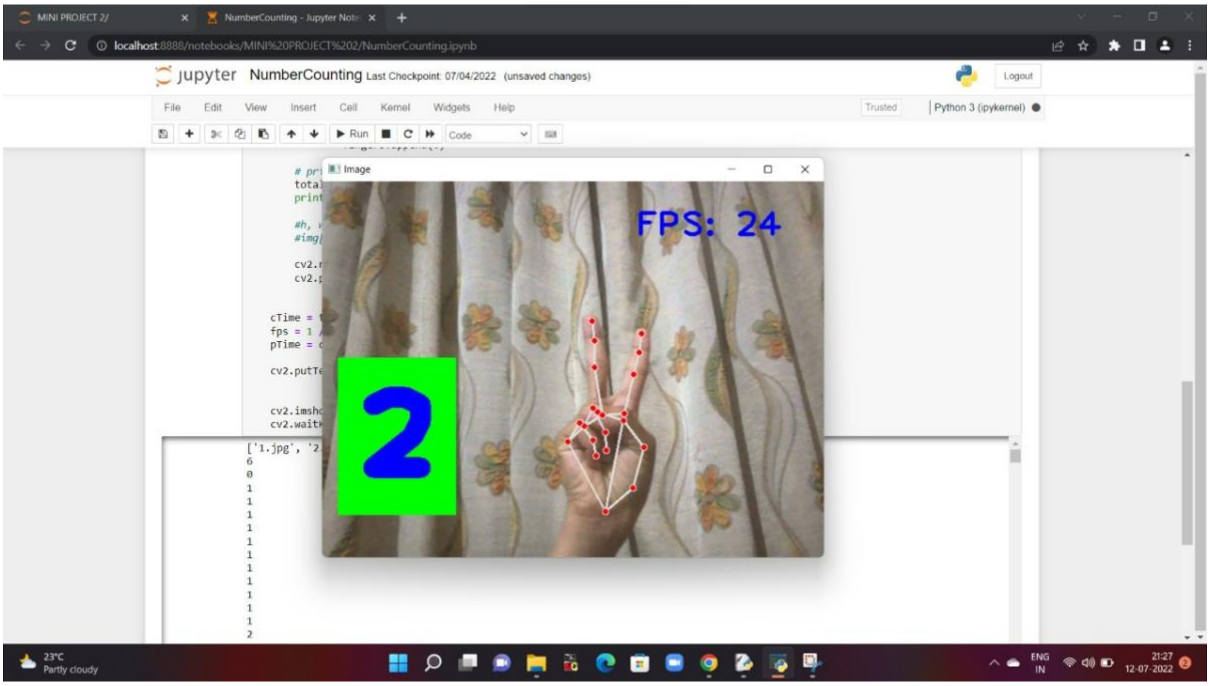

Figure 4.2.3 The output when the sign is recognized as 2 in real-time shown on the screen.

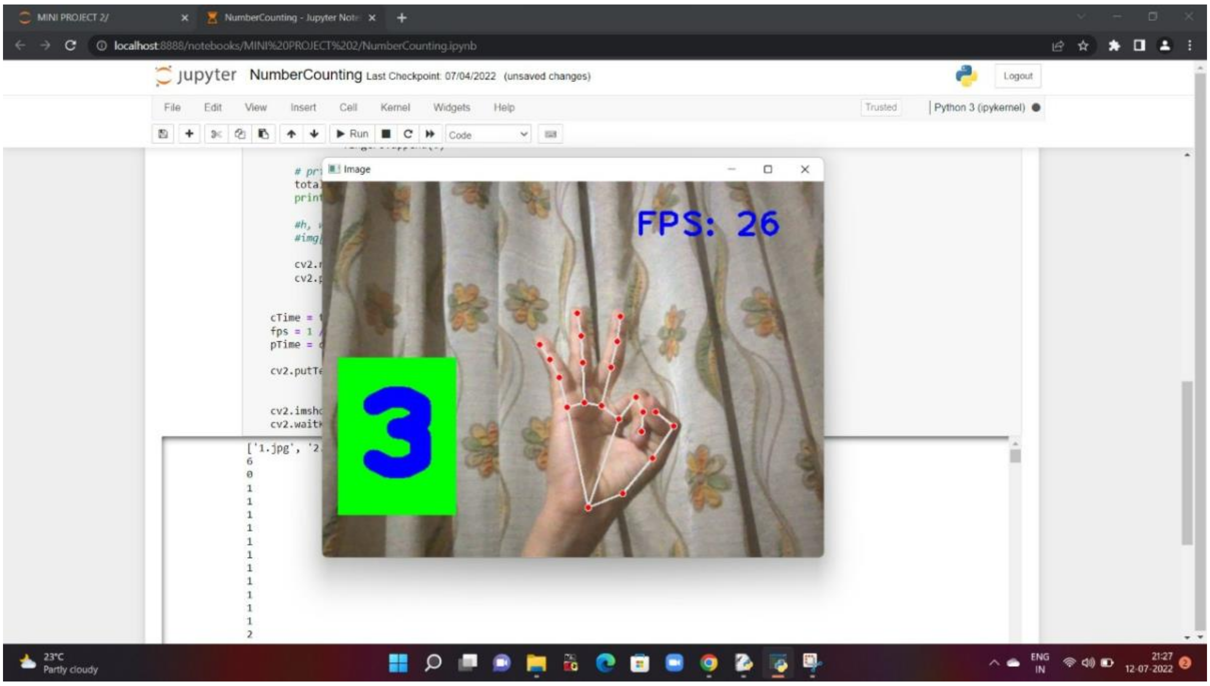

Figure 4.2.4 The output when the sign is recognized as 3 in real-time shown on the screen.

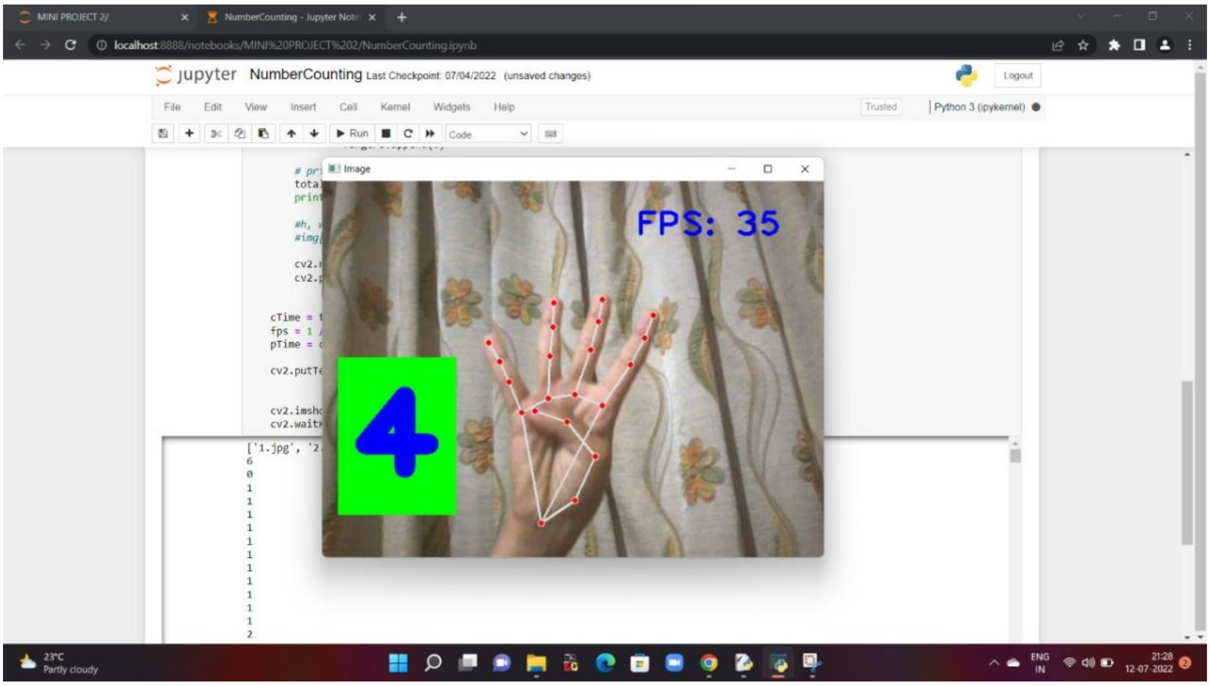

Figure 4.2.5 The output when the sign is recognized as 3 in real-time shown on the screen.

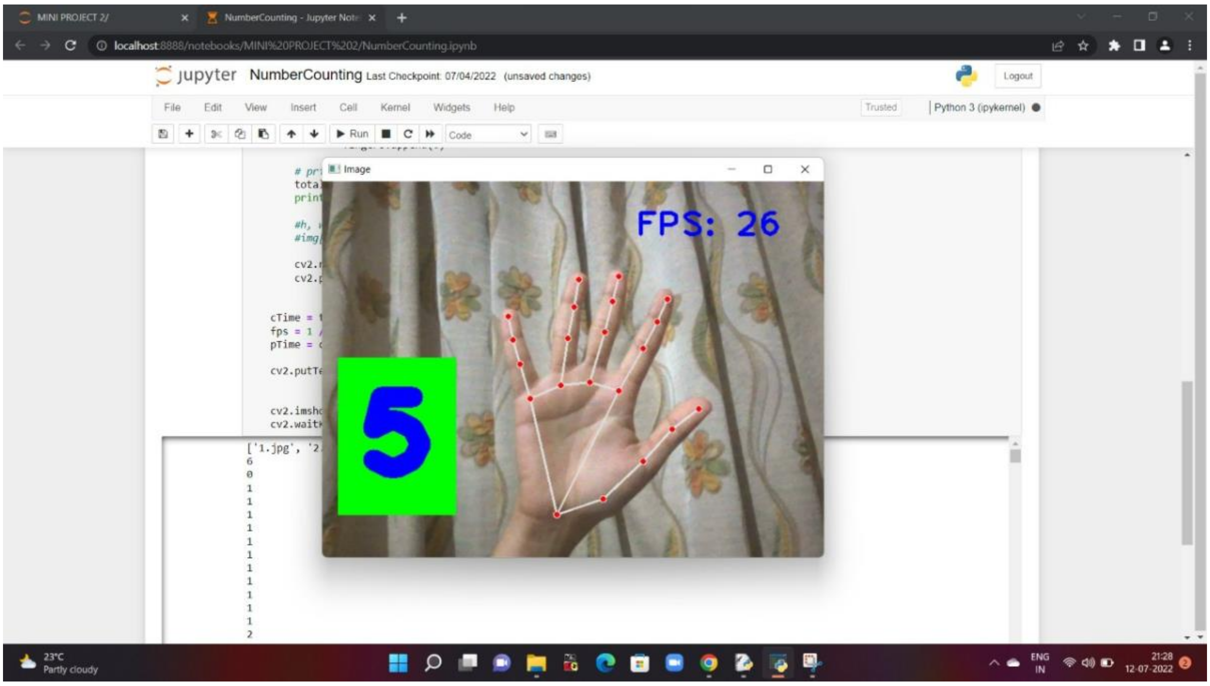

Figure 4.2.6 The output when the sign is recognized as 3 in real-time shown on the screen.

CHAPTER-5

RESULTS AND DISCUSSIONS

From the above images we can see that the signs are recognized in real-time and shown on the screen in a loop. This depicts the results achieved in our program. Further more this results are very accurate and have less chances of error.

Our project shows that CNNs can used to recognize different sign languages accurately. This project aids in communicating to the people having vocal and hearing disabilities. It discusses an improved methods for sign language recognition.

Our project is useful to most of those deaf children who are born into hearing families put of which majority of them do not begin their sign language learning before the age 4-6 in school. So overall this project will help those children to learn the basic numbers. In our project we were able to develop a practical and meaningful sign language recognition system that holds true to our proposed system.

Hence the results shown by our system during the capture of our hand gesture through webcam is very accurate.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

We are going to implement sign language recognition using the concepts of machine learning, open cv and python. This project aims to create a simpler method for sign language recognition helping deaf and dumb individuals and we conclude that the project has achieved its aim.

After our application is implemented, we would like to benefit the children, NGOs and various kinds of people who require special help. Through this project we learnt that basic approaches can work better than the complex approaches. We also realized the difficulties and time constraints involved in creating a dataset from the scratch.

For further enhancement of our project, we can come up with various adoptions like recognition of English sentences and alphabets, we can create a model. A system that can recognize changes in the temporal space will be needed for this.

Raspberry Pi can be used to create and implement the proposed system. Additionally, we can concentrate on translating the serios of motions into texts or words and sentences and finally into audible speech.

The frame work of this project can also be extended to further other so many applications like controlling robot navigation using hand gesture. Hence we can do further more enhancement of our project by taking the required guidance from our mentor.

REFERENCES

1. Identification of images in the data set Ciresan, D., Meier, U., Schmidhuber

2. Dynamic hand gesture recognition using vision-based approach for human–computer interaction. J. Singha, A. Roy, and R. H. Lashkar

3. Image processing in python P Umesh; CSI Communications,2012 – Citeseer

4. Image pre-processing in python Luong Chi Mai

5.https://ieeexplore.ieee.org/document/8798856

6. https://learnopencv.com/introduction-to-mediapipe/

7. https://github.com/topics/sign-language-recognition-system

8.https://www.geeksforgeeks.org/opencv-python-tutorial/#:~:text=OpenCV%20is%20a%20huge%20open,the%20handwriting%20of%20a%20human.