



ISHARA

VOCALIZING THE SILENCE

Sign Language Recognition System that converts Sign Language to Text



TEAM ISHARA

SIDDHARTHA SHARMA 20BAI10044

SHIVAM SHARMA 20BAI10058

AVRODEEP SAHA 20BAI10041

MANSI SAHU 20BAI10158

UNDER THE SUPERVISION OF : DR. NILAMADHAV MISHRA

PROBLEM STATEMENT



Problem Statement

The easiest and prominent way for communication of deaf & mute people is by performing certain actions by hand to express different gestures, to express their ideas with other people i.e. sign language.

However, due to the lack of awareness of sign language, these people have hard times communicating. They still face various psychological, educational, employment, and social problems.

With this project, we aim to give a right to speak to those who cannot speak. Our objective is to design a solution that is intuitive and simple. Communication for the majority of people is not difficult. It should be the same way for the deaf.



SIMPLIFIES
SIGN
LANGUAGE
USAGE



GIVES EQUAL
RIGHT TO
SPEECH TO
EVERYONE

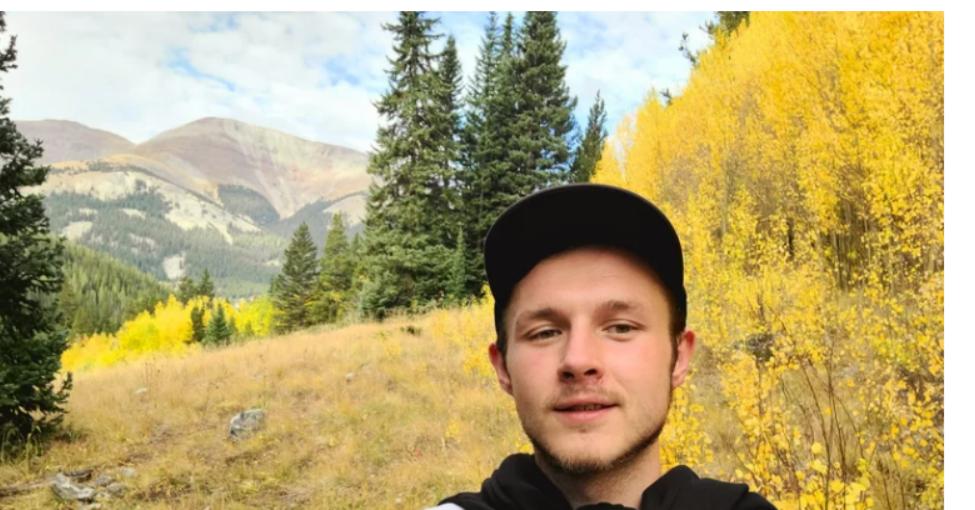
Our project aims at helping deaf and mute people to actually communicate with people without abilities and create an inclusive environment that further prevents happenings like these.

LAW

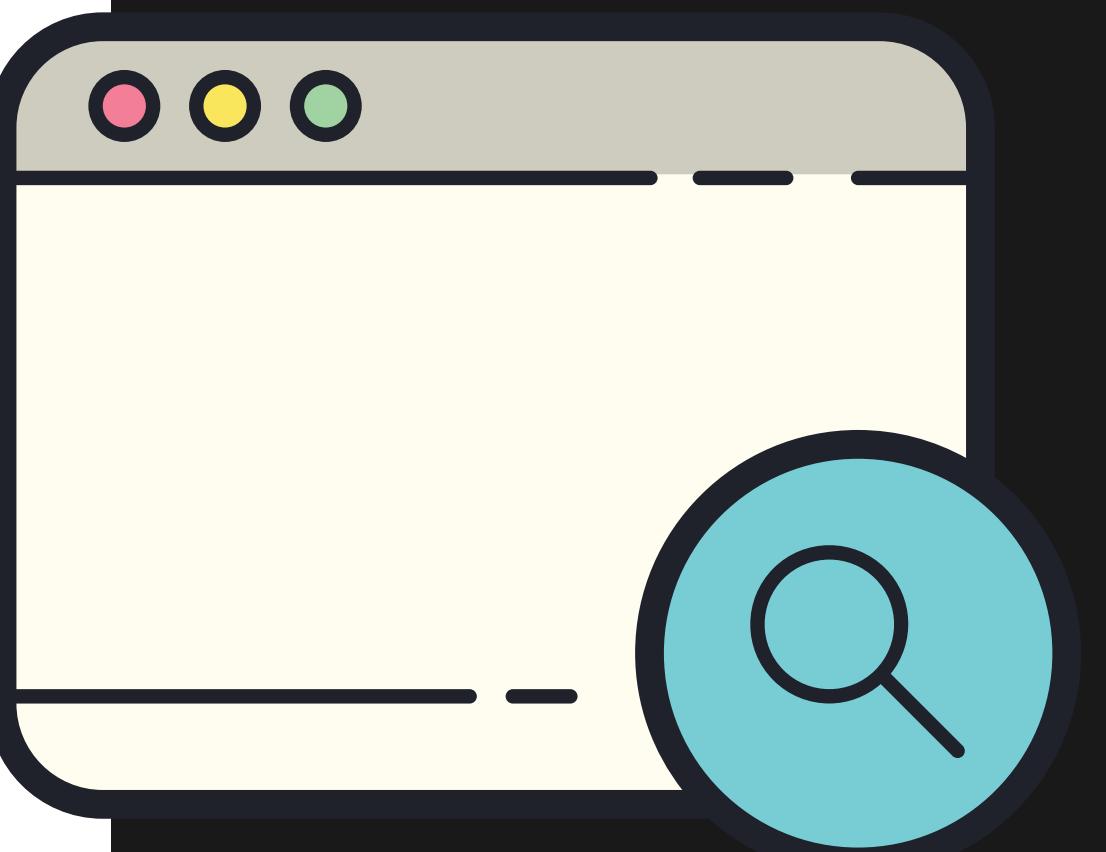
A Deaf Man Who Couldn't Hear Police Commands Was Tased And Spent 4 Months In Jail

September 29, 2021 · 4:19 PM ET

JOE HERNANDEZ



Real-Life Application



PRE-REQUISITES

System & Hardware Requirements



Python (3.6.6)

Numpy (version 1.15.3)

cv2 (version 81.3.4.3)

Time

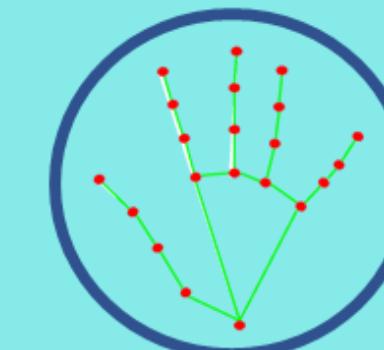
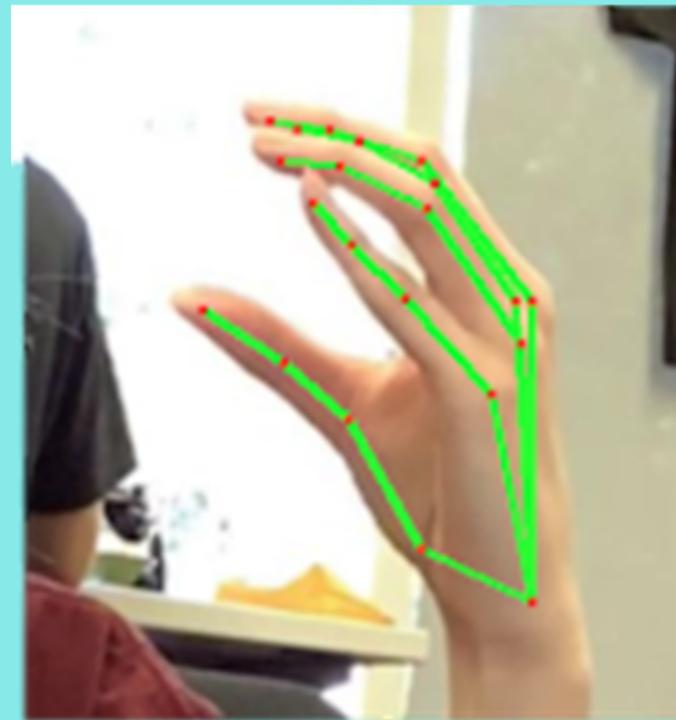
Math

Mediapipe

Autocorrect

Translate

FLOW DIAGRAM



Capturing

The foremost requirement is to capture the image and that can be done by scanning using cameras



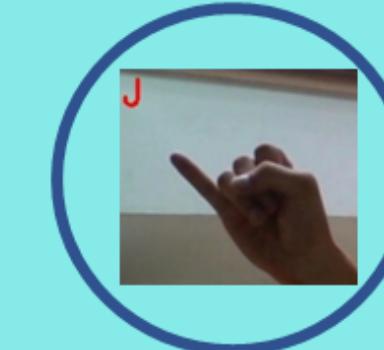
Extracting

Unique gestures is then extracted from the Database matching with the gesture input.



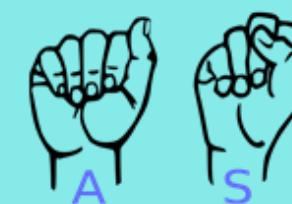
Comparison

The gestures are then compared on the basis of the standard ISL gestures and gesture meaning is returned.



Checking

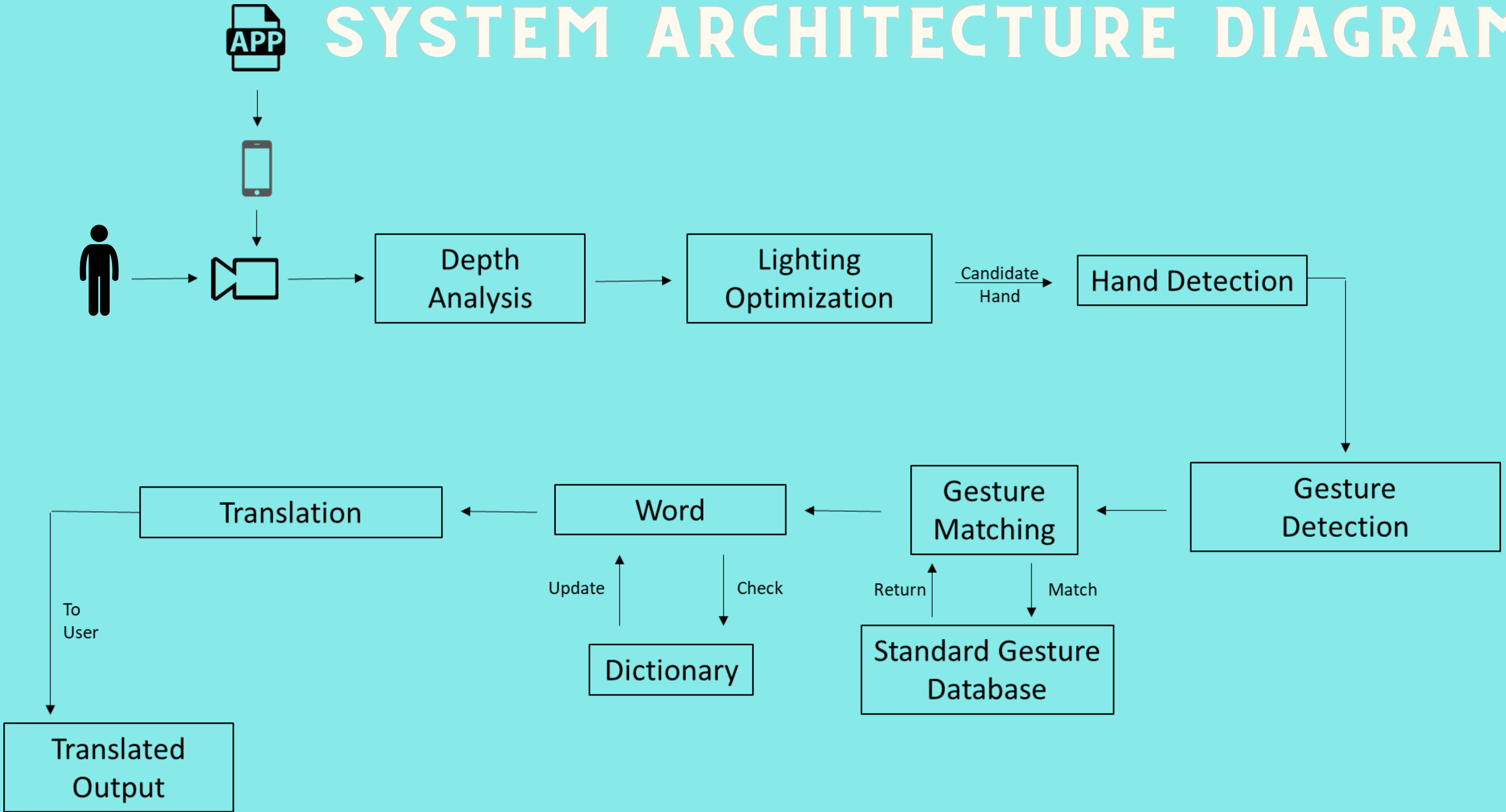
The output is now checked based on the existing and understandable words and grammar.



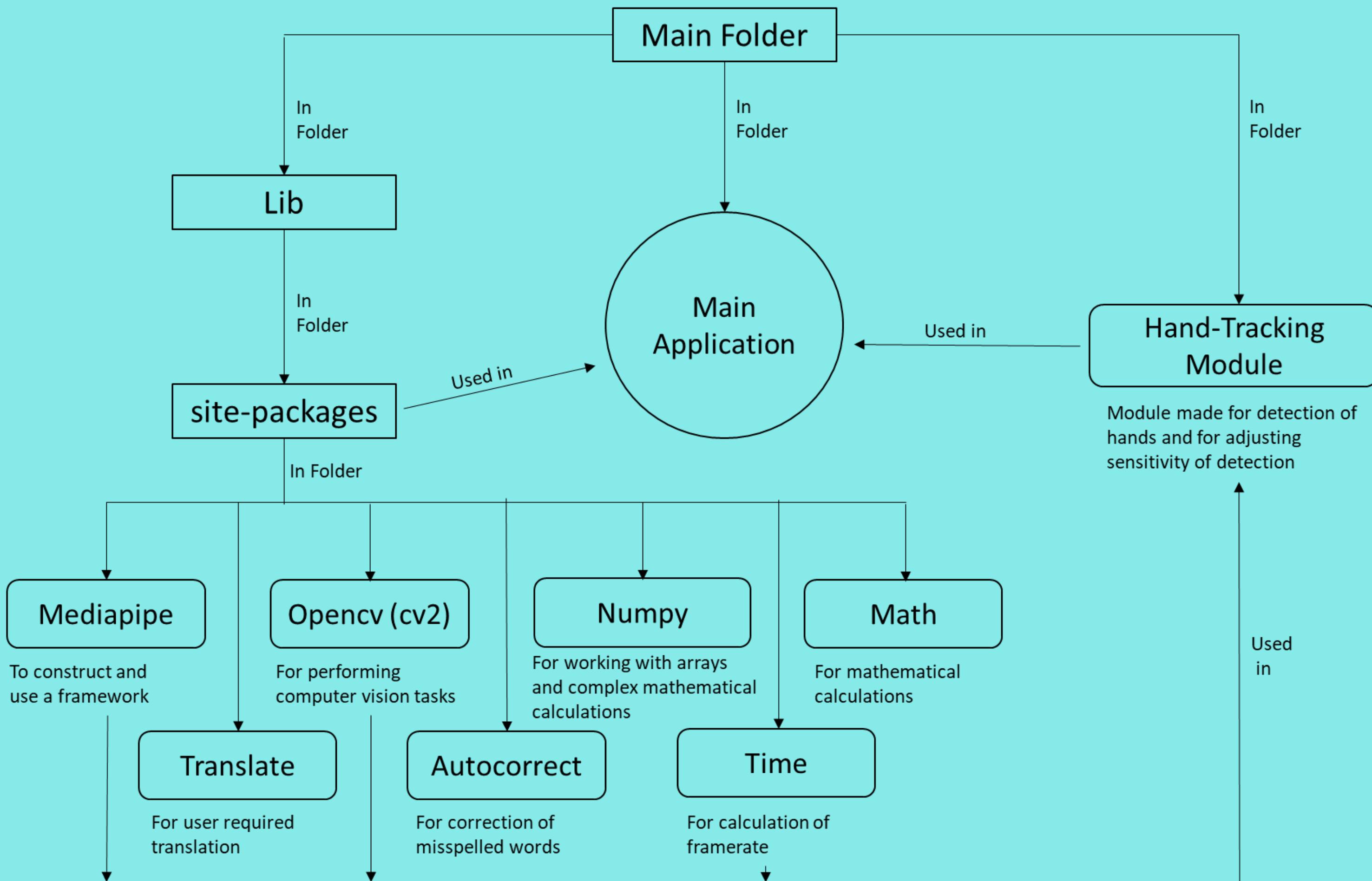
Output



SYSTEM ARCHITECTURE DIAGRAM



ORGANISATION OF MODULES



EXPLANATION

```
import cv2  
import time  
import numpy as np  
import Hand_Tracking_Module as htm  
import math
```

Importing
Modules

Setting up camera and using
hand detection module to detect hands

```
wCam, hCam = 640, 480  
cap = cv2.VideoCapture(0)  
cap.set(3, wCam)  
cap.set(4, hCam)  
pTime = 0  
detector = htm.handDetector(detectionCon=0.7)
```

EXPLANATION

```
count = 0  
CN=25  
word = []  
pword = ""  
str = ""
```

Defining a function
to append gesture words



Setting up
Global variables

```
def call(let):  
    global pword  
    if (pword != let):  
        word.append(let)  
    pword = let
```

EXPLANATION

```
while True:  
    success, img = cap.read()  
    img = detector.findHands(img)  
    lmList = detector.findPosition(img, draw=False)  
  
    cv2.line(img, (0, 400), (640, 400), (0, 0, 0), 3)
```

Initiating for loop and running cv2 as well as hand detection to identify our hands gestures

Setting up variables to store the pattern postions

```
40 if len(lmList) != 0:  
41  
42     Tx, Ty = lmList[4][1], lmList[4][2]  
43     T2x, T2y = lmList[3][1], lmList[3][2]  
44     Ix, Iy = lmList[8][1], lmList[8][2]  
45     Sx, Sy = lmList[20][1], lmList[20][2]  
46     M4x, M4y = lmList[9][1], lmList[9][2]  
47     M3x, M3y = lmList[10][1], lmList[10][2]  
48     Mx, My = lmList[12][1], lmList[12][2]  
49     Px, Py = lmList[0][1], lmList[0][2]  
50     I3x, I3y = lmList[6][1], lmList[6][2]  
51     Rx, Ry = lmList[16][1], lmList[16][2]
```

EXPLANATION

```
53     INTH = math.hypot(Ix - Tx, Iy - Ty)
54     M4TH = math.hypot(M4x - Tx, M4y - Ty)
55     M3TH = math.hypot(M3x - Tx, M3y - Ty)
56     MI = math.hypot(Mx - Ix, My - Iy)
57     IP = math.hypot(Ix - Px, Iy - Py)
58     I3TH = math.hypot(I3x - Tx, I3y - Ty)
59     RS = math.hypot(Rx-Sx, Ry-Sy)
60     ST = math.hypot(Sx-Tx, Sy-Ty)
61     T2M = math.hypot(T2x-Tx, T2y-Ty)
62     MT = math.hypot(Mx-Tx, My-Ty)
63     SI = math.hypot(Sx-Ix, Sy-Iy)
64     MR = math.hypot(Mx-Rx, My-Ry)
65     RI = math.hypot(Rx-Ix, Ry-Iy)
66     RT = math.hypot(Rx-Tx, Ry-Ty)
```

Setting up
Pattern relations

Defining letters and words
for gesture matching

```
if (IP >= 75 and IP <= 110 and ST>160 and ST<200):
    count = count + 1
    if count > CN:
        let = "A"
        count = 0
        call(let)
```

EXPLANATION

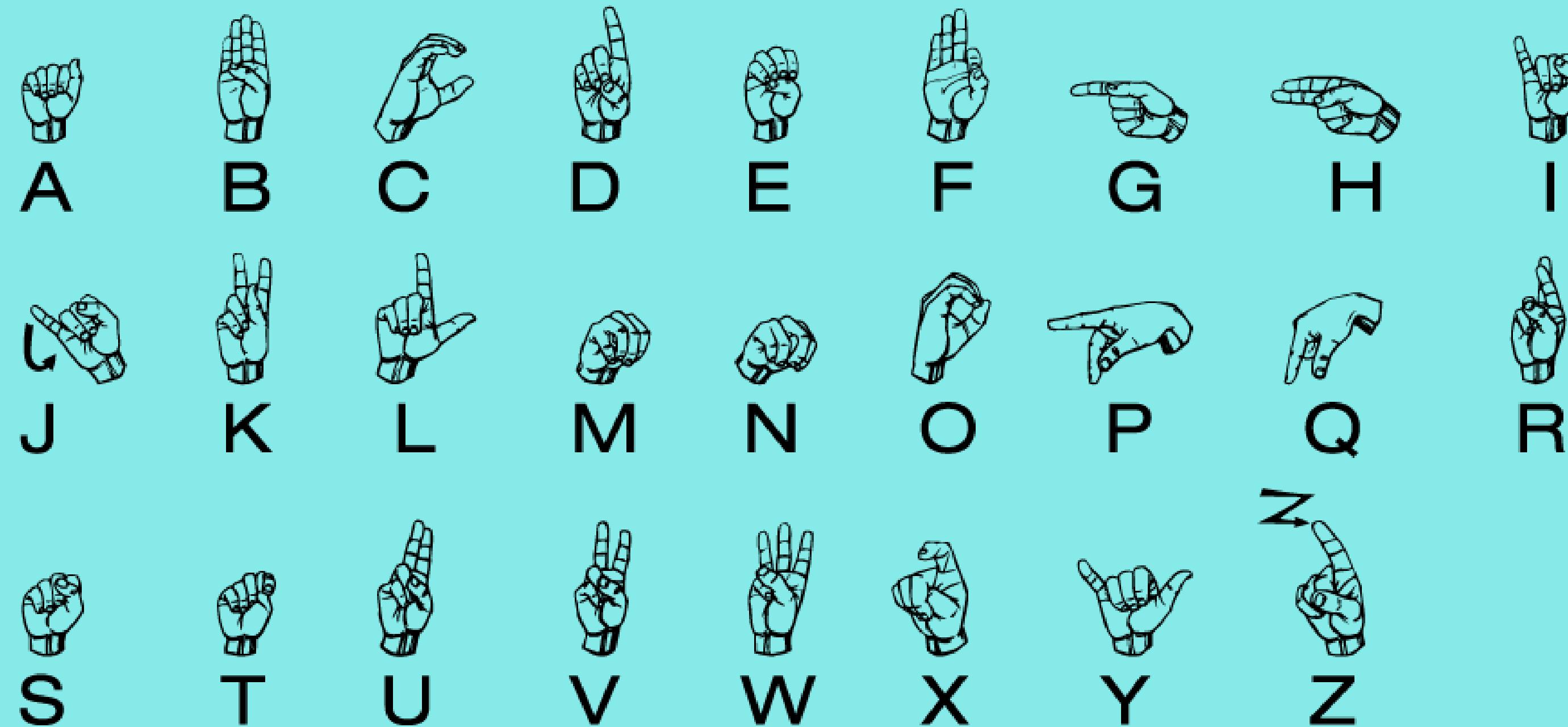
```
cTime = time.time()  
fps = 1 / (cTime - pTime)  
pTime = cTime  
  
cv2.putText(img, f'FPS: {int(fps)}', (450, 50),  
            cv2.FONT_HERSHEY_PLAIN, 2, (0, 0, 0), 2)
```



To display
FPS

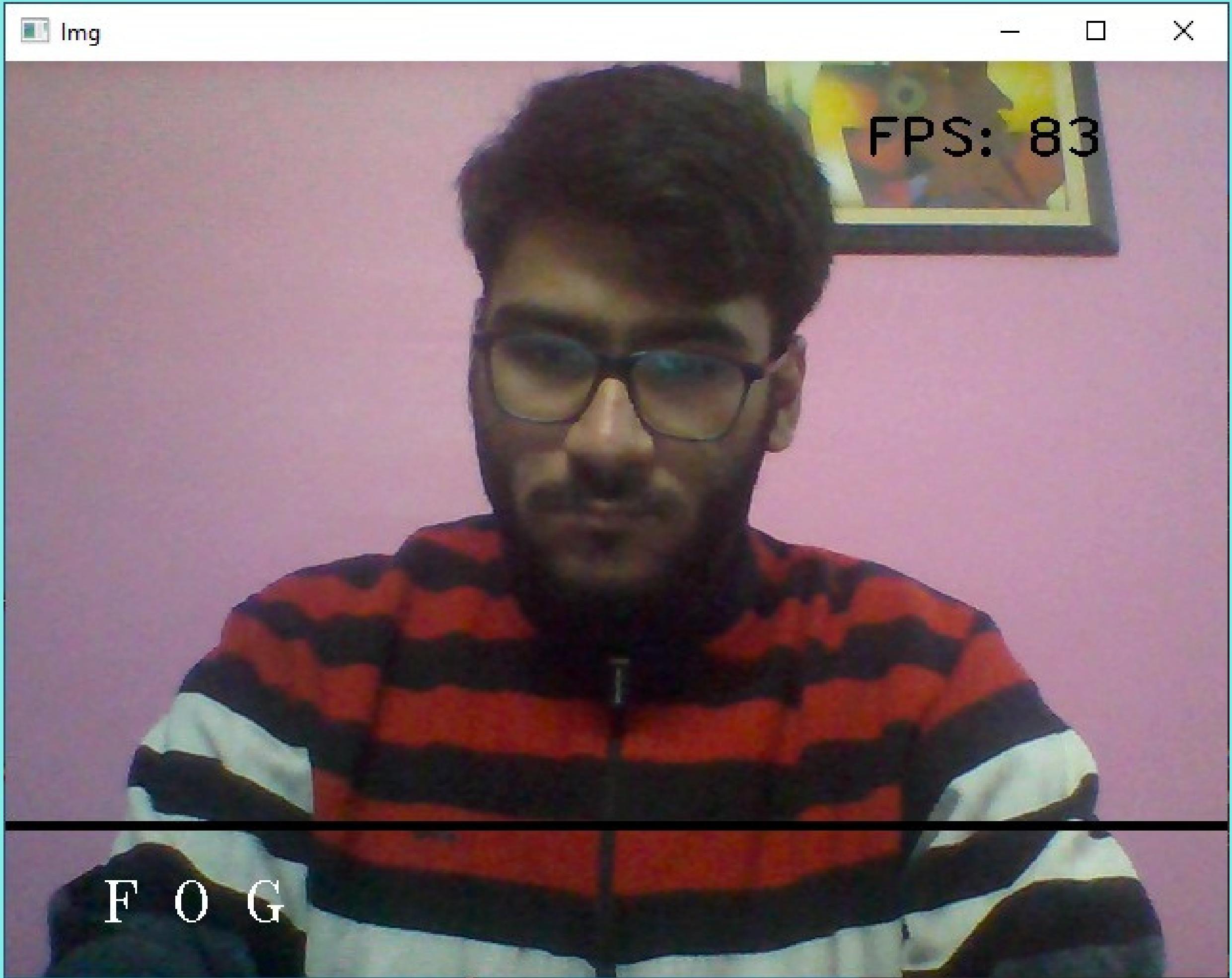
IMPLEMENTATION

In order to type a letter we need to make sure that we put a symbol according to the below mentioned schema which every mute or deaf person will understand as they would have learnt it.



Based on this we can form a letter through our program a screenshot of the same is given here:





When we write a few letters together we are able to make a word that is corrected by the autocorrect library as and when needed.

DEMO VIDEO

The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Hand_Tracking_module.py - Ishara - Visual Studio Code.
- Editor Area:** Displays Python code for a hand tracking module. The code imports cv2 and mediapipe, defines a handDetector class with __init__ and findHands methods, and handles multi-hand landmarks.
- Explorer:** Shows the project structure under "OPEN EDITORS" and "ISHARA".
- Terminal:** Shows command-line output related to file sizes: [4, 256, 332], [4, 258, 322], [4, 258, 328], [4, 255, 328], [4, 277, 344], [4, 100, 403].
- Status Bar:** Python 3.9.7 64-bit, 0 errors, 0 warnings, 0 info, 0 tips. Includes file navigation icons, search bar, and system status (18°C, ENG, 7:42 PM, 11/24/2021).
- Bottom Right:** A context menu is open with options: cmd, Debug Ter..., and Code.

Link For the video : <https://drive.google.com/file/d/11yOM9mzzEwHXYkcbXUTbvDyP5Hgknnx-/view?usp=sharing>



AUTO CORRECTION

The autocorrect library is not great. We are looking for alternatives as to how can we improve the library being used for correction.



CONFIGURATION OF BACKSPACE

Space and backspace are the things we are still developing along with letters from P to Z.



CAMERA REQUIREMENTS

Our program requires a webcam with > 8mp lens and memory space > 150mb

LIMITATIONS

Future Scope

What are we planning to do next?

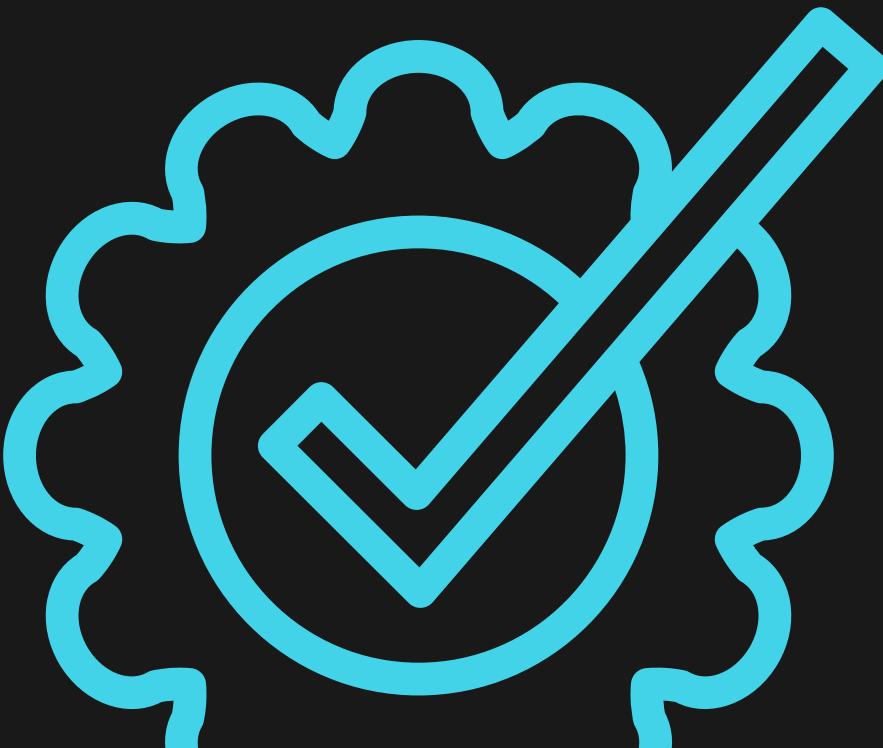
There are plenty of features we plan to add to our project such as the integration of a virtual assistant with minimal modifications.

The system can be extended to incorporate the knowledge of facial expressions and body language too so that there is a complete understanding of the context and tone of the input speech.

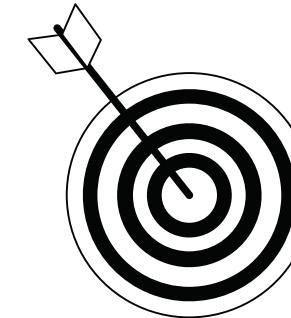
We're also planning to create a web-based application for our program in order to increase the reach. Further, it's our goal to develop text to sign language (vice versa) features for two-way communication.



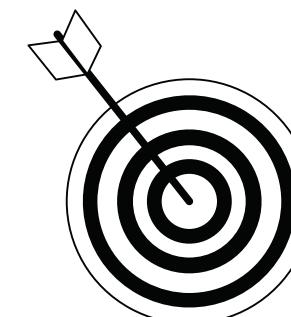
Various Advantages



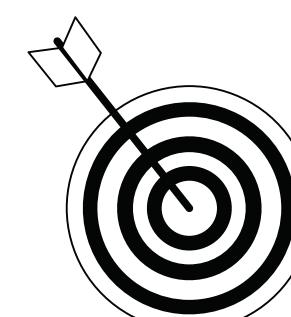
Various Advantages of our Program



**HELPS DEAF & MUTE
PEOPLE TO
COMMUNICATE**



**FAST, COMPATIBLE &
EFFICIENT**



UNIQUE & RELIABLE

Thank you

