Project Report

Conversion of Sign Language to text

Name - Shubhadeep Chakraborty

Registration No - 1252011400012

Name - Sourav Dhali

Registration No - 1252011400001

Subject Code - CMSADSE06P

COURSE : B.Sc. (HONOURS) IN COMPUTER SCIENCE

Supervisor: Mr. Sumit Bagchi

Department of Computer Science
Mrinalini Datta Mahavidyapith
Birati, Kolkata 700051

Declaration

I confirm that the project is my original work, developed according to guidelines. Where some source materials have been used, these have been acknowledged.

Signature: ____________________

Introduction

A person with speaking disorders face major problems of expressing their emotions as freely in this world.

“Not able to utilize” some of the new technologies.

The primary objective of this project is to develop a Python program that can identify the hand sign of dumb & deaf person based on specific hand sign and show the letter of the hand sign in display for easy understand.

Methodology

Data Collection:

In the first step we have to collect data of hand signs.
These genareted signs can be collected from our own hands.

Alphabet Identification:

Next, we need to identify a list of alphabet keywords that are commonly used in communication with each other
the alphabets will be used by the program to classify the hand signs user showing on the camera.

Program Development:

Using Python, we will develop a program that scan the sign content of each genareter hand and searches for the matching alphabet. Based on these input data, the program will classify the sign. The program should be updateable for new alphabets or word signs in future.

Accuracy:

The program will then create separate image for each
hand genaretion on 60 FPS and scan classified image into their respective CNN model.

Implementation

Required Libraries:

1. `import cv2 ------------ #For camera`
2. `import tensorflow ------------ #For classifier`
3. `import math --------#For many purpoes`

Starting the Webcam:

```
import cv2
cap=cv2.VideoCapture(0)
while True:
        success,img=cap.read()
        cv2.imshow("^-^",imgOutput)
        cv2.waitKey(1)
```

If there is no hand on the camera then the display not showing alphabet signs

Testing Code:

```
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
classifier=Classifier('model/keras_model.h5','model/labels.txt')
labels=['A','B','C','D','E','F','G','I','K','L','O','P',
                        'Q','W','X','Y','Like','Dislike','Rock','Thank you']
```

Then we take all alphabet and scan the hand sign, make a secparate image of hand sign and the model classified the image scan and output the alphabet.

Our full Code:

```
import cv2
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import numpy as np
import math
cap=cv2.VideoCapture(0)
detector=HandDetector(maxHands=1)
classifier=Classifier('model/keras_model.h5','model/labels.txt')
offset=20
imgSize=300
labels=['A','B','C','D','E','F','G','I','K','L','O','P',
 'Q','W','X','Y','Like','Dislike','Rock','Thank you']
while True:
    success,img=cap.read()
    imgOutput=img.copy()
    hands,img=detector.findHands(img)
    if hands:
        hand=hands[0]
        x,y,w,h=hand['bbox']
        imgWhite=np.ones((imgSize,imgSize,3),np.uint8)
        imgCrop=img[y-offset:y+h+offset,x-offset:x+w+offset]
        aspectRatio=h/w
        if aspectRatio>1:
            k=imgSize/h
            wCal=math.ceil(k*w)
            imgResize=cv2.resize(imgCrop,(wCal,imgSize))
            imgResizeShape=imgResize.shape
            wGap=math.ceil((imgSize-wCal)/2)
            imgWhite[:,wGap:wCal+wGap]=imgResize
            prediction,index=classifier.getPrediction(imgWhite,draw=False)
        else:
            k=imgSize/w
            hCal=math.ceil(k*h)
            imgResize=cv2.resize(imgCrop,(imgSize,hCal))
            imgResizeShape=imgResize.shape
            hGap=math.ceil((imgSize-hCal)/2)
            imgWhite[hGap:hCal+hGap,:]=imgResize
            prediction,index=classifier.getPrediction(imgWhite,draw=False)
        cv2.rectangle(imgOutput,(x-offset,y-offset-50),
                           (x-offset+90,y-offset-50+50),(255,0,255),cv2.FILLED)
        cv2.putText(imgOutput,labels[index],(x,y-26),
                      cv2.FONT_HERSHEY_COMPLEX,1.7,(255,255,255),2)
        cv2.rectangle(imgOutput,(x-offset,y-offset),
                                                  (x+w+offset,y+h+offset),(255,0,255),4)
    cv2.imshow("^-^",imgOutput)
    cv2.waitKey(1)
```

Conclusion:

The hand genaretion Classification System aims to improve the communication of dumb and deaf people

This project will not only enhance the accessibility of alphabet and help in to say the word by just sign hand. The system can be expanded to new words and alphabets.

Bibliography:

We use some documentation on python for implementation of hand genaretion clasifier .

Our Supervisor "Sumit bagchi" helped us in every step and provided feedback for this project.