Minor Project Report

on

Sign Language Recognition

A Project Report/Synopsis submitted in partial fulfilment of the requirements for the award of

Bachelor in Engineering
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by
Priyadarshini
(Roll no: LCO17373)

Under the supervision of
Dr. Dheerendra Singh

CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY (DEGREE WING)

Government Institute under Chandigarh (UT) Administration, Affiliated to Panjab University, Chandigarh

Sector-26, Chandigarh. PIN-160019

CONTENTS

CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY (DEGREE WING)
Government Institute under Chandigarh (UT) Administration | Affiliated to Panjab University , Chandigarh
Sector-26, Chandigarh. PIN-160019 | Tel. No. 0172-2750947, 2750943 Ibsite:
www.ccet.ac.in | Email: principal@ccet.ac.in | Fax. No. :0172-2750872

Department of Computer Sc. & Engineering

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled "**Sign Language Recognition**" in fulfilment of the requirement for the award of the degree Bachelor of Engineering in Computer Science & Engineering, submitted in CSE Department, Chandigarh College of Engineering & Technology (Degree wing) affiliated to Punjab University, Chandigarh, is an authentic record of my/our own work carried out during my degree under the guidance of **Dr. Dheerendra Singh**. The work reported in this has not been submitted by me for award of any other degree or diploma.

Date: 2nd May, 2021 Name: Priyadarshini

Place: Chandigarh Roll No.: LCO17373

CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY (DEGREE WING)
Government Institute under Chandigarh (UT) Administration | Affiliated to Panjab University , Chandigarh
Sector-26, Chandigarh. PIN-160019 | Tel. No. 0172-2750947, 2750943 |
Ibsite: www.ccet.ac.in | Email: principal@ccet.ac.in | Fax. No. :0172-2750872

Department of Computer Sc. & Engineering

CERTIFICATE

This is to certify that the Project work entitled "**Sign Language Recognition**" submitted by **Priyadarshini, LC017365** in fulfilment for the requirements of the award of Bachelor of Engineering Degree in Computer Science & Engineering at Chandigarh College of Engineering and Technology (Degree Wing), Chandigarh is an authentic work carried out by him/her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree.

Date: 2th May,2021

Place: Chandigarh

Dr. Dheerendra Singh

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my mentor “**Dr. Dheerendra Singh**” for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of my project. I deeply express our sincere thanks to our Head of Department “Dr Prof. S.K. Singh” for encouraging and allowing me to present the project on the topic **“Sign Language Recognition’’** at department premises for the partial fulfillment of the requirements leading to the award of B-Tech degree. I take this opportunity to thank all our lecturers who have directly or indirectly helped me in my project for sharing their valuable knowledge and time. I pay my respects and love to my parents and all other family members and friends for their love, help and encouragement throughout our career. Last but not the least I express my thanks to my friends for their cooperation and support.

Priyadarshini

LCO17373

BE-CSE 7th semester

Chapter 1

Introduction

1.1 INTRODUCTION

This project aims to identify Sign Language. I intend to solve this problem using state-of-the-art vision computers and learning algorithm algorithms. The problem I am investigating is the recognition of sign language through unsupervised learning. I am developing a sign language recognition system using a small vocabulary and personalized database. Deaf and mute people rely on sign language interpreters for communication. However, finding experienced and trained translators in their daily affairs of life is a daunting task. Therefore, the computer-to-computer system will be a reliable and consistent solution for such people.

Sign language is an important means of communication for the deaf. Since sign language is a Ill-organized form of code, each action has a given meaning. It can be used to convey complex meanings by combining basic elements. Sign-language vision is a problem for computer viewing while it is very useful for deaf and dumb people to communicate with. The problem I am investigating is the recognition of sign language through unsupervised learning. I am developing a sign language recognition system using a small vocabulary and personalized database. Deaf and mute people rely on sign language interpreters for communication. However, finding experienced and trained translators in their daily affairs of life is a daunting task.

While sign language is very essential for deaf-mute people, to communicate both with normal people and with themselves, is still getting less attention from the normal people. The importance of sign language has been tending to ignored, unless there are areas of concern with individuals who are deaf-mute. One of the solutions to talk with the deaf-mute people is by using the mechanisms of sign language.

Hand gesture is one of the methods used in sign language for non-verbal communication.It is most commonly used by deaf & dumb people who have hearing or talking disordersto communicate among themselves or with normal people. Various sign language systems have been developed by many manufacturers around the world but they are neither flexible nor cost-effective for the end users.

1.2 SCOPE

One of the solutions to communicate with the deaf-mute people is by using the servicesof sign language interpreter. But the usage of sign language interpreters could be expensive. Cost-effective solution is required so that the deaf-mute and normal peoplecan communicate normally and easily.

This strategy involves implementing such an application which detects pre-defined American sign language (ASL) through hand gestures. For the detection of movement of gesture, I would use basic level of hardware component like camera and interfacing is required.

I want to introduce software that will introduce a type of system that will automatically detect sign language to help deaf and mute people communicate better with other people or ordinary people.

It is difficult to find a knowledgeable and educated sign language translator at all times and everywhere but this personal and computer communication system can be installed wherever possible.

This application will comprise of two core module one is that simply detects the gesture and displays appropriate alphabet. The second is after a certain amount of interval period the scanned frame would be stored into buffer so that a string of character could be generated forming a meaningful word.

1.3 Problem Statement:

Given a hand gesture, implementing such an application which detects pre-defined American sign language (ASL) in a real time through hand gestures and providing facility for the user to be able to store the result of the character detected in a txt file, also allowing such users to build their customized gesture so that the problems faced by persons who aren’t able to talk vocally can be accommodated with technological assistance and the barrier of expressing can be overshadowed.

Chapter 2

System Specification

2.1 SYSTEM REQUIREMENT

2.1 SOFTWARE REQUIREMENTS

The prerequisites software & libraries for the sign language project are:

Python (3.7.4)

IDE (Jupyter)

Numpy (version 1.16.5)

cv2 (openCV) (version 3.4.2)

Keras (version 2.3.1)

Tensorflow (as keras uses tensorflow in backend and for image preprocessing) (version 2.0.0)

2.2 SYSTEM FEATURES

User-friendly based GUI .

Real time American standard character detection based on gesture made byuser.

Customized gesture generation.

Chapter 2

System Design

3.1 SYSTEM ARCHITECTURE

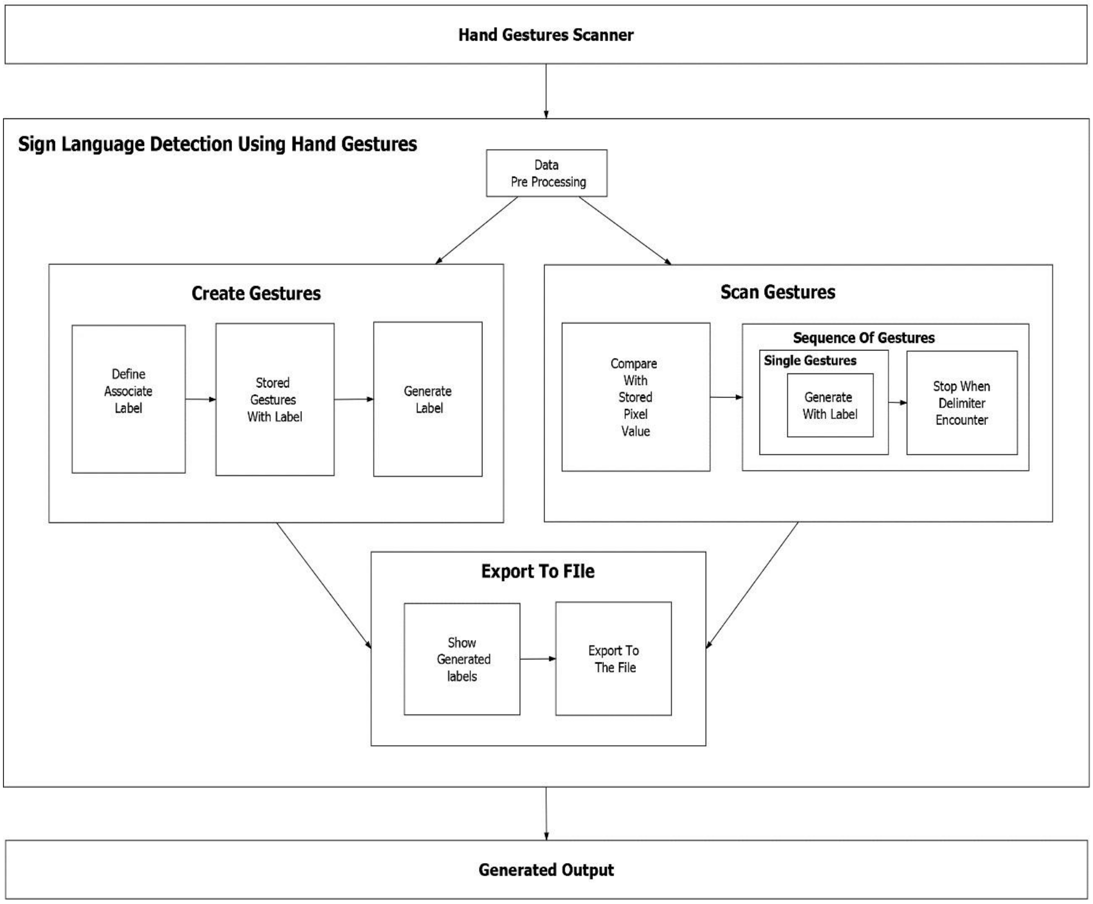


fig 1: System Architecture for Sign Language Recognition Using Hand Gestures.

3.2 MODULES IN THE SYSTEM

Data Pre-Processing – In this module, based on the object detected in front of the camera its binary images is being populated. Meaning the object will be filled with solid white and background will be filled with solid black. Based on the pixel's regions, their numerical value in range of either 0 or 1 isbeing given to next process for modules.

Scan Single Gesture – A gesture scanner will be available in front of the end user where the user will have to do a hand gesture. Based on Pre-Processed module output, a user shall be able to see associated label assigned for each hand gestures, based on the predefined American Sign Language (ASL) standard inside the output window screen.

Create gesture –A user will give a desired hand gesture as an input to the system with the text box available at the bottom of the screen where the user needs to type whatever he/she desires to associate that gesture with.

Formation of a sentence – A user will be able to select a delimiter and until that delimiter is encountered every scanned gesture character will be appended with the previous results forming a stream of meaning-full words and sentences.

Exporting – A user would be able to export the results of the scanned character into an ASCII standard textual file format.

3.3 USE CASE DIAGRAM

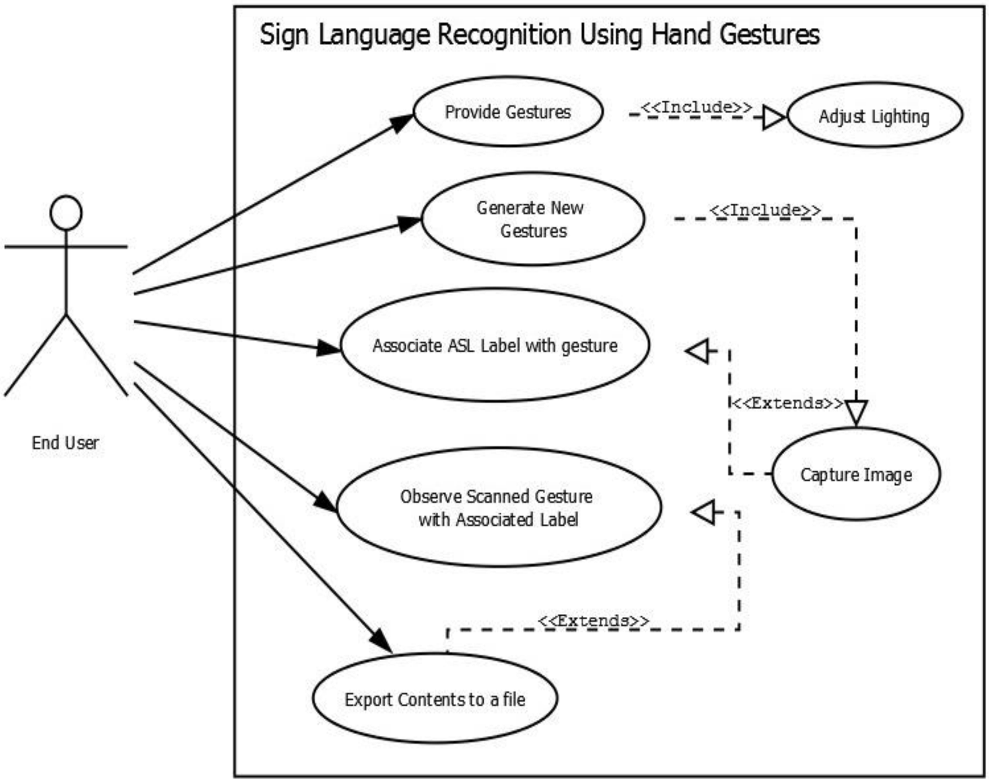

Sign Language Recognition Using Hand Gestures
Provide Gestures
<<Include>>
Adjust Lighting
Generate New Gestures
<<Include>>
Associate ASL Label with gesture
<<Extends>>
Capture Image
End User
Observe Scanned Gesture with Associated Label
<<Extends>>
Export Contents to a file

Fig 2: Use Case Diagram for Sign Language Recognition Using Hand Gestures

3.4 ACTIVITY DIAGRAM

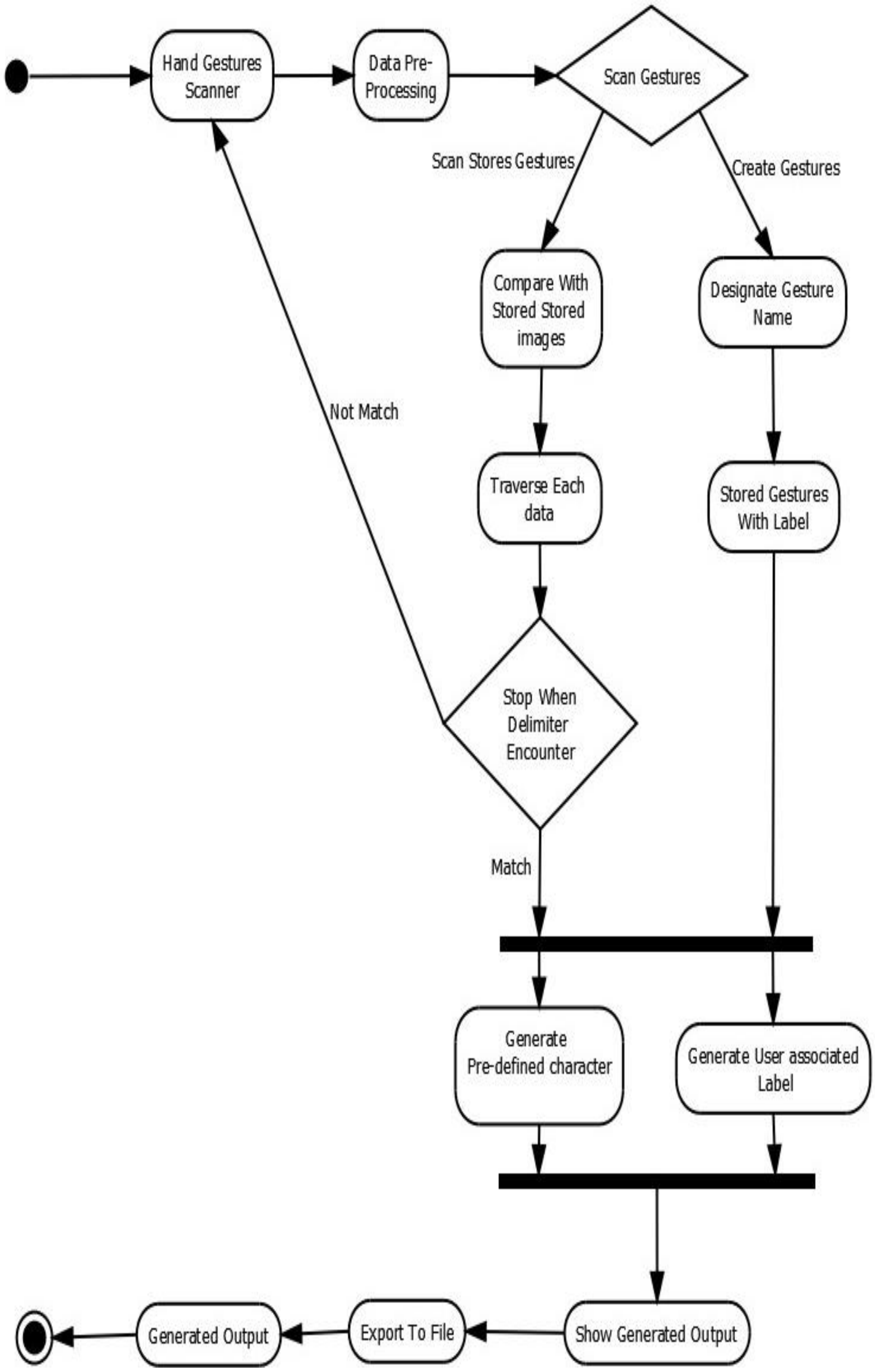


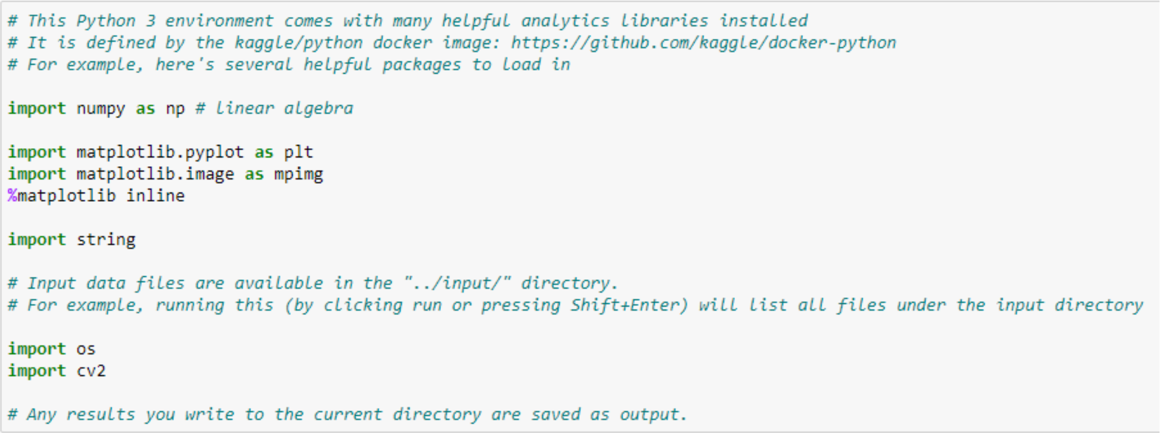
Fig 3: Activity Diagram for Sign Language Recognition Using Hand Gestures

Chapter 4

Implementation

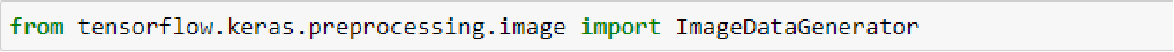
4.1 CODE SNIPPETS

Loading the libraries



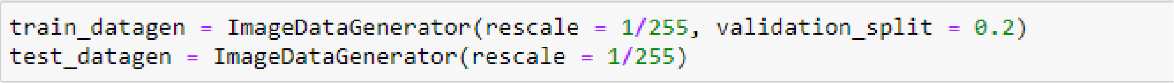
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline

import string

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
import cv2

# Any results you write to the current directory are saved as output.
```

- **Preparing the data generator**



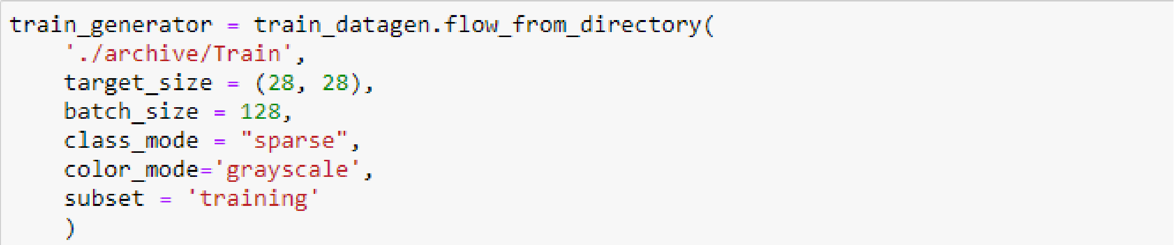
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Normalizing the data before feeding to model

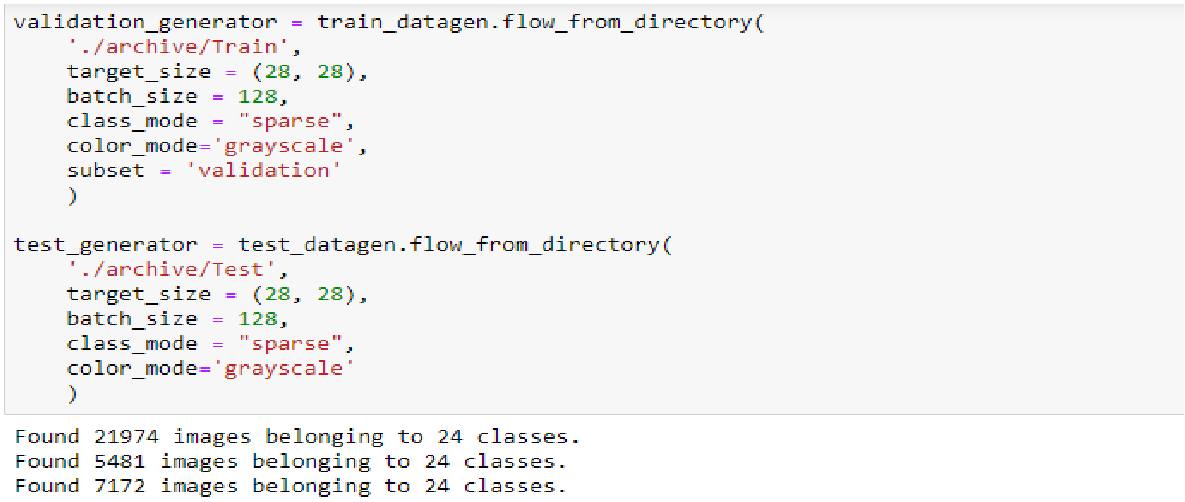


```
train_datagen = ImageDataGenerator(rescale = 1/255, validation_split = 0.2)
test_datagen = ImageDataGenerator(rescale = 1/255)
```

Loading data as 28 * 28 grayscale images

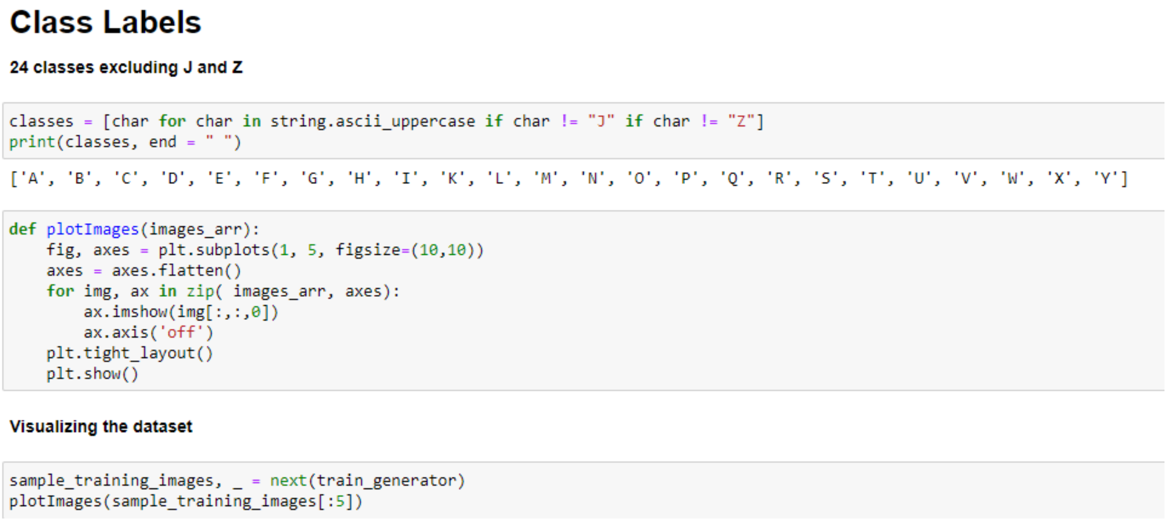


```
train_generator = train_datagen.flow_from_directory(
    './archive/Train',
    target_size = (28, 28),
    batch_size = 128,
    class_mode = "sparse",
    color_mode='grayscale',
    subset = 'training'
    )
```



```
validation_generator = train_datagen.flow_from_directory(
    './archive/Train',
    target_size = (28, 28),
    batch_size = 128,
    class_mode = "sparse",
    color_mode='grayscale',
    subset = 'validation'
    )

test_generator = test_datagen.flow_from_directory(
    './archive/Test',
    target_size = (28, 28),
    batch_size = 128,
    class_mode = "sparse",
    color_mode='grayscale'
    )
```

```
Found 21974 images belonging to 24 classes.
Found 5481 images belonging to 24 classes.
Found 7172 images belonging to 24 classes.
```



Class Labels

24 classes excluding J and Z

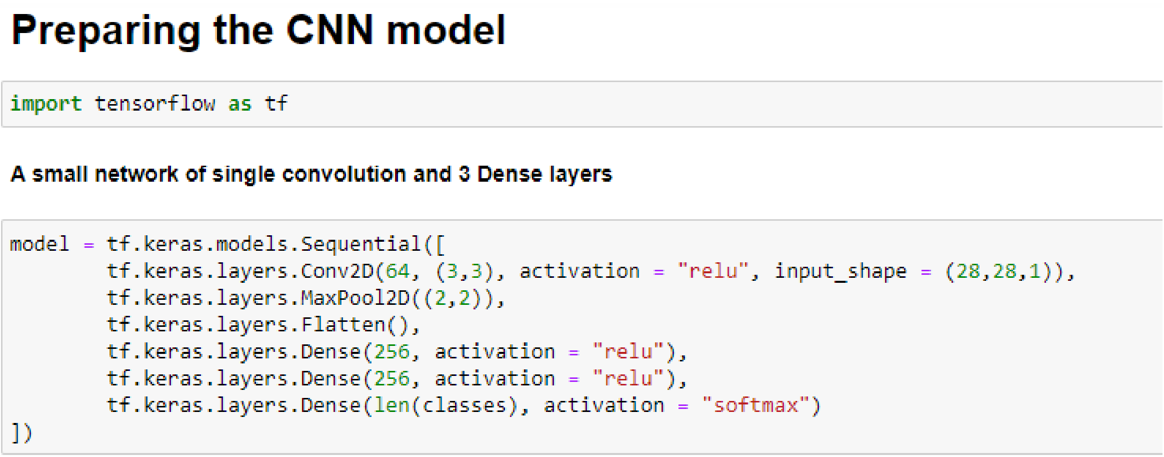
```
classes = [char for char in string.ascii_uppercase if char != "J" if char != "Z"]
print(classes, end = " ")
```

```
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y']
```

```
def plotImages(images_arr):
    fig, axes = plt.subplots(1, 5, figsize=(10,10))
    axes = axes.flatten()
    for img, ax in zip( images_arr, axes):
        ax.imshow(img[:,:,0])
        ax.axis('off')
    plt.tight_layout()
    plt.show()
```

Visualizing the dataset

```
sample_training_images, _ = next(train_generator)
plotImages(sample_training_images[:5])
```



Preparing the CNN model

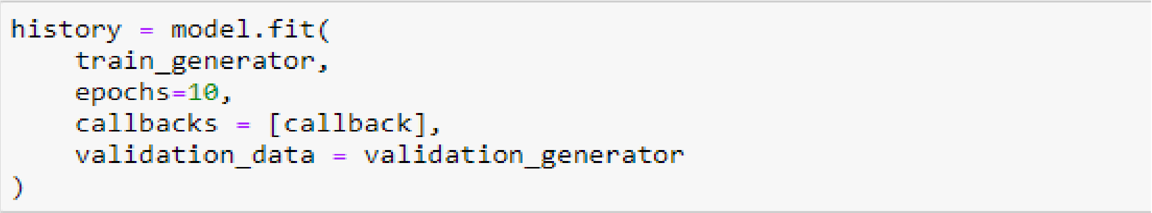
```
import tensorflow as tf
```

A small network of single convolution and 3 Dense layers

```
model = tf.keras.models.Sequential([
        tf.keras.layers.Conv2D(64, (3,3), activation = "relu", input_shape = (28,28,1)),
        tf.keras.layers.MaxPool2D((2,2)),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(256, activation = "relu"),
        tf.keras.layers.Dense(256, activation = "relu"),
        tf.keras.layers.Dense(len(classes), activation = "softmax")
])
```

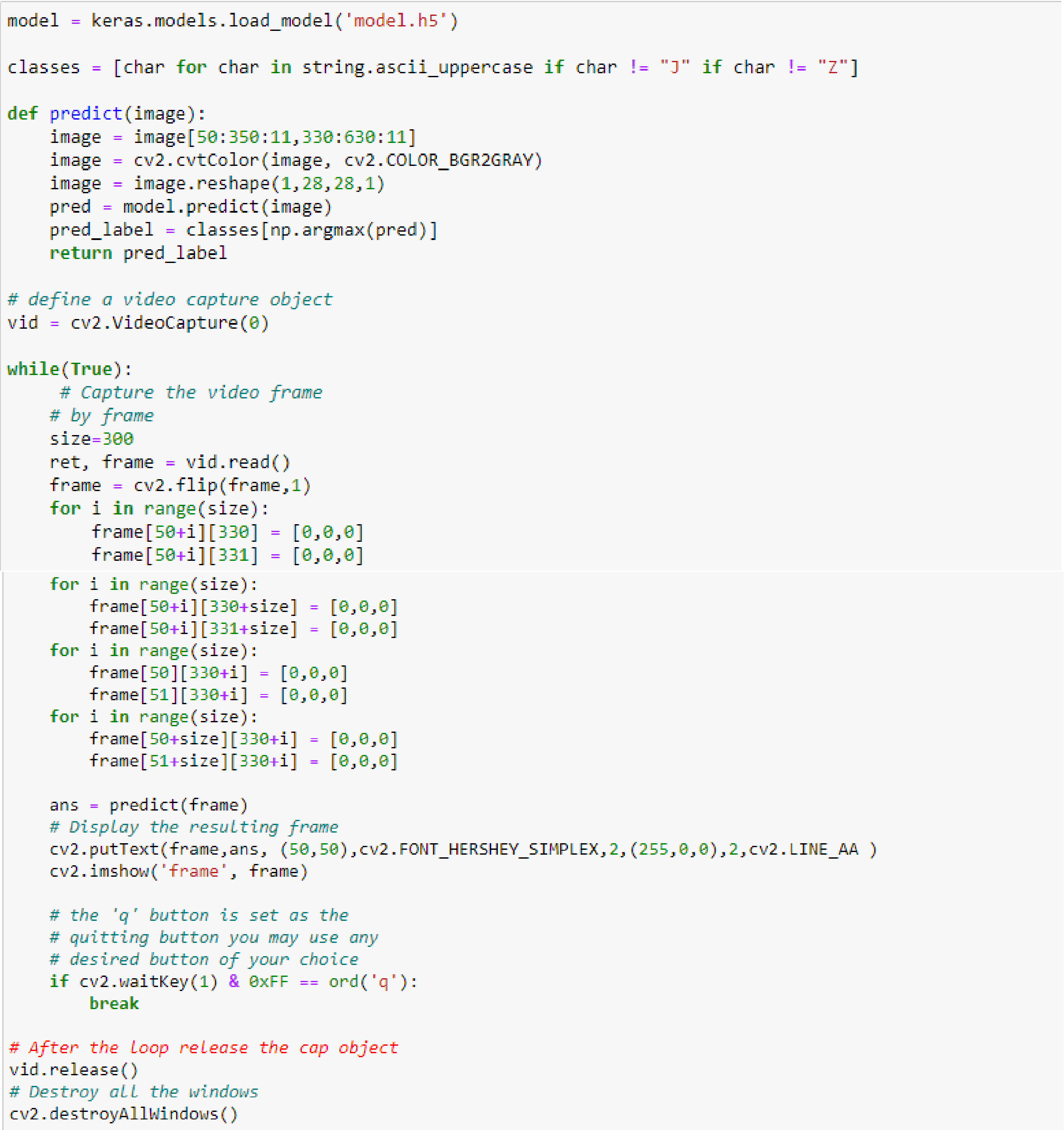
Training

Training for 10 epochs



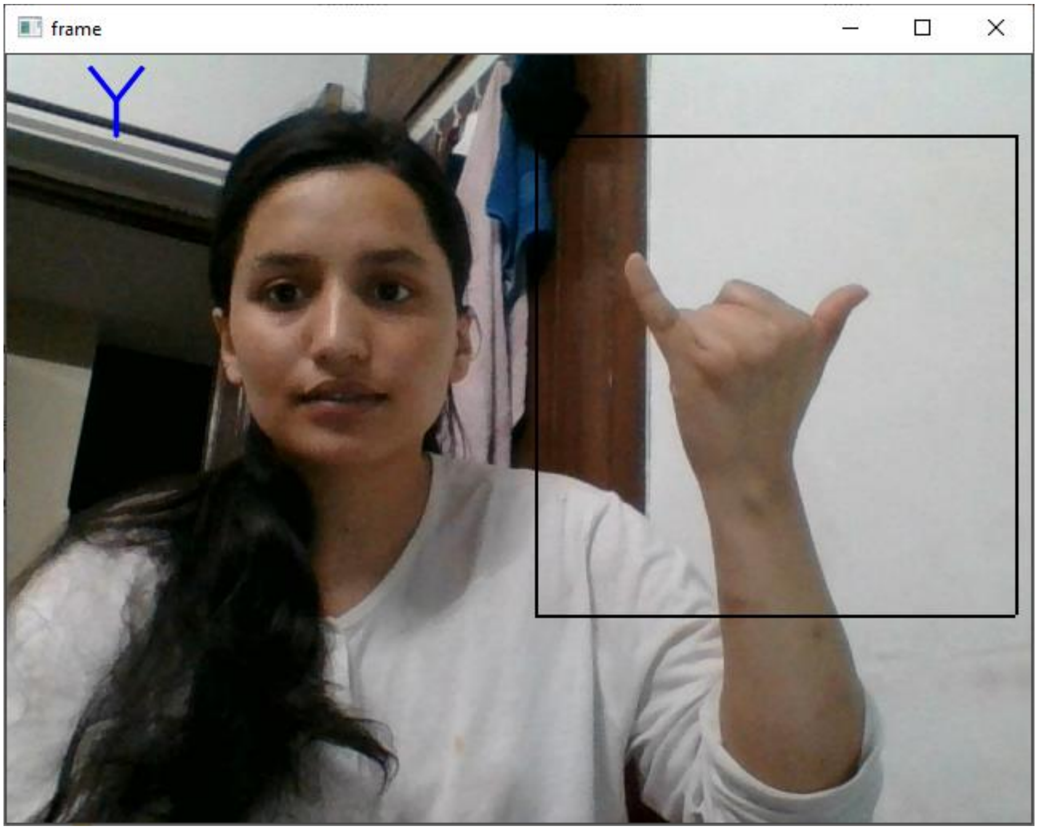
```
history = model.fit(
     train_generator,
     epochs=10,
     callbacks = [callback],
     validation_data = validation_generator
)
```

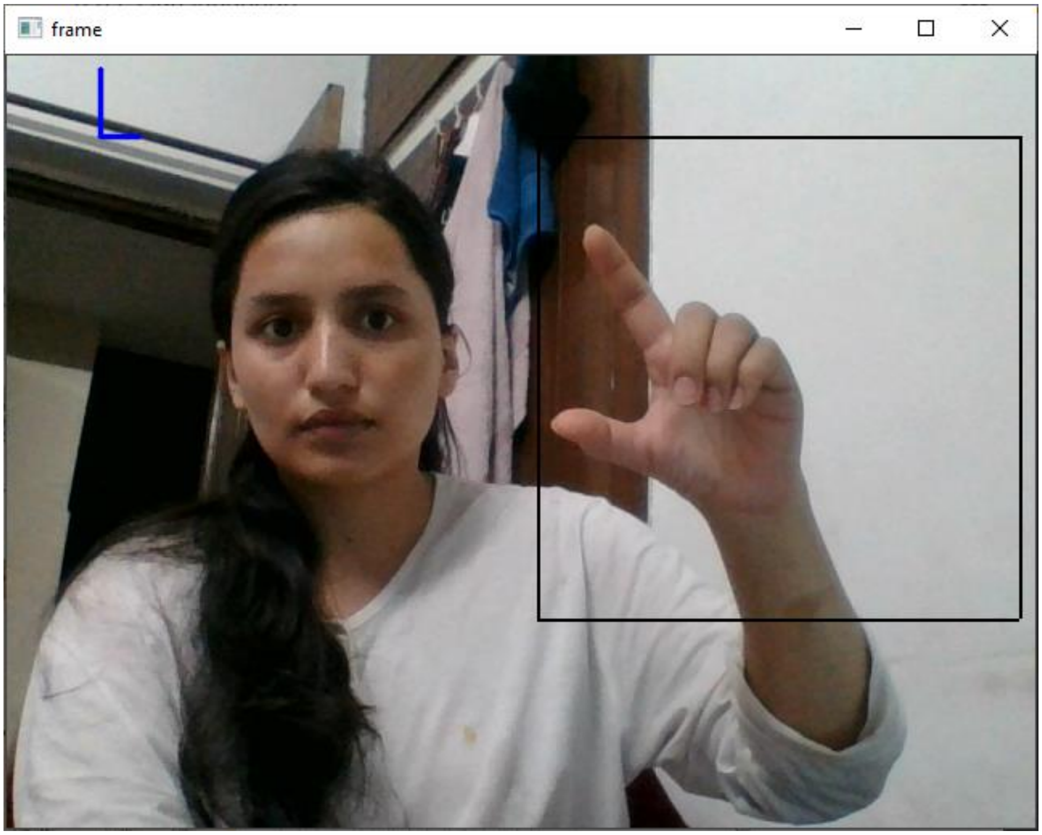
main program

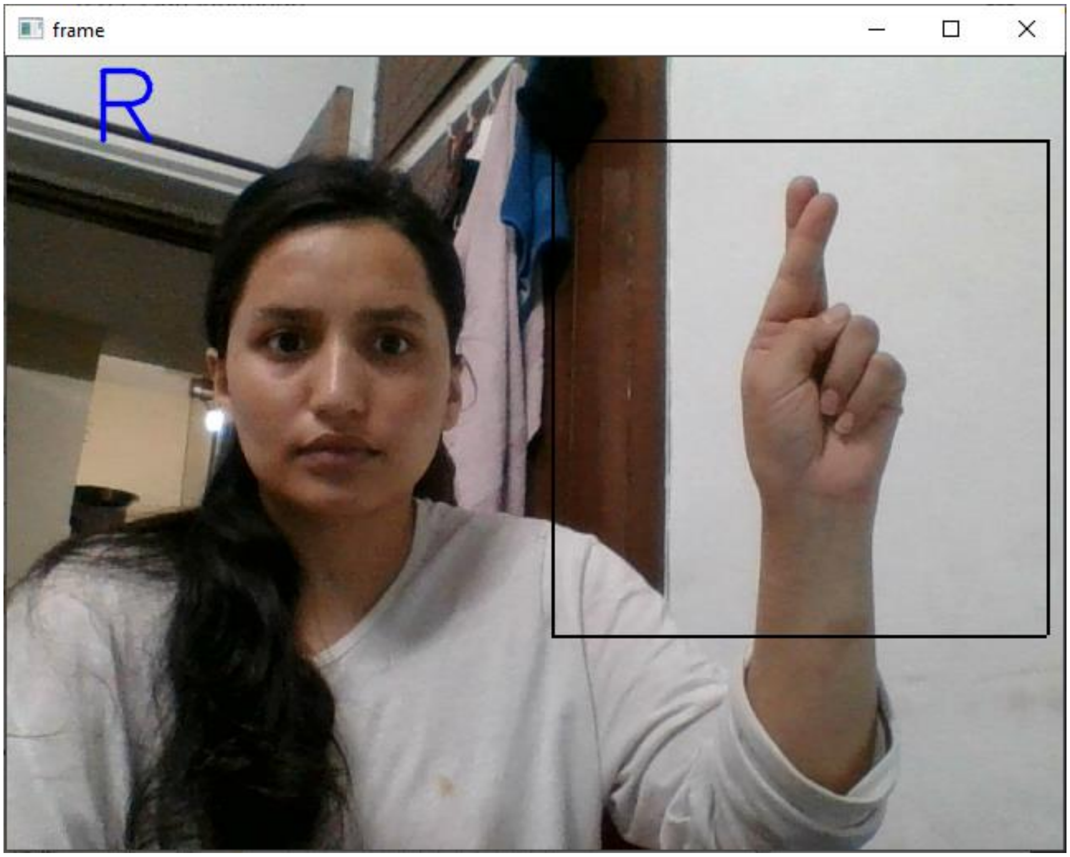


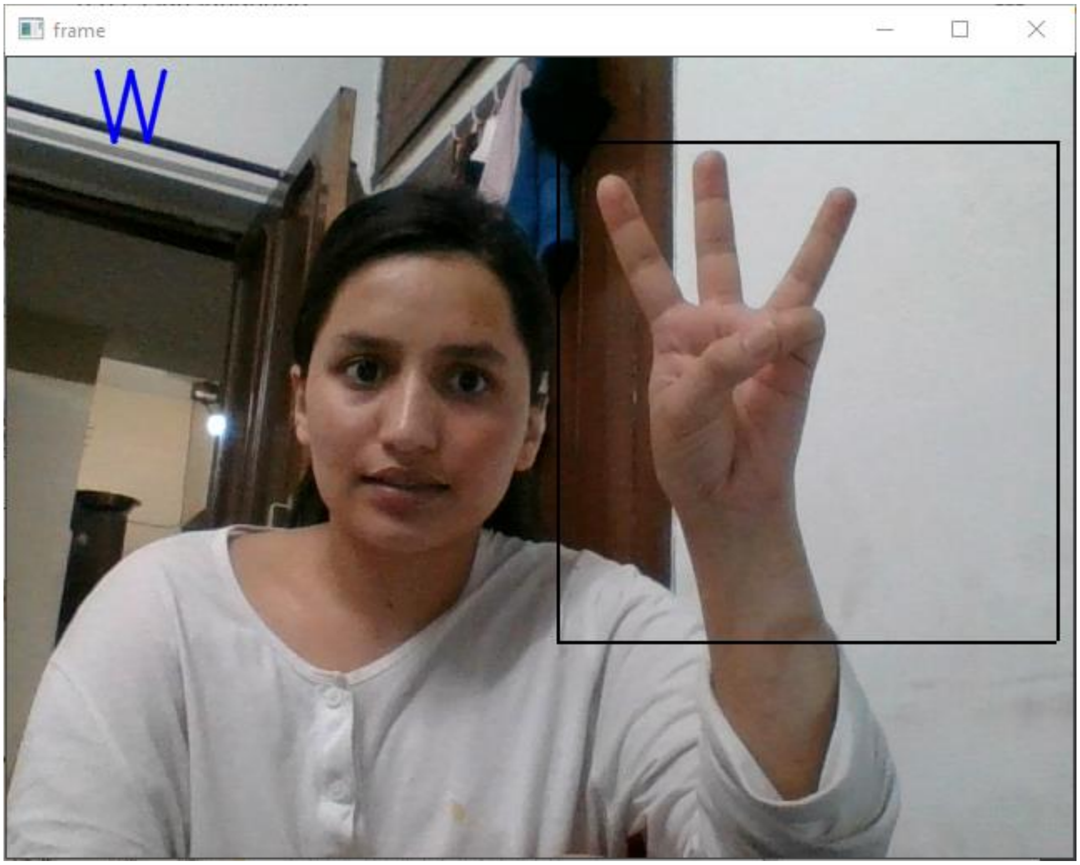
```
model = keras.models.load_model('model.h5')

classes = [char for char in string.ascii_uppercase if char != "J" if char != "Z"]

def predict(image):
    image = image[50:350:11,330:630:11]
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = image.reshape(1,28,28,1)
    pred = model.predict(image)
    pred_label = classes[np.argmax(pred)]
    return pred_label

# define a video capture object
vid = cv2.VideoCapture(0)

while(True):
     # Capture the video frame
    # by frame
    size=300
    ret, frame = vid.read()
    frame = cv2.flip(frame,1)
    for i in range(size):
        frame[50+i][330] = [0,0,0]
        frame[50+i][331] = [0,0,0]
    for i in range(size):
        frame[50+i][330+size] = [0,0,0]
        frame[50+i][331+size] = [0,0,0]
    for i in range(size):
        frame[50][330+i] = [0,0,0]
        frame[51][330+i] = [0,0,0]
    for i in range(size):
        frame[50+size][330+i] = [0,0,0]
        frame[51+size][330+i] = [0,0,0]

    ans = predict(frame)
    # Display the resulting frame
    cv2.putText(frame,ans, (50,50),cv2.FONT_HERSHEY_SIMPLEX,2,(255,0,0),2,cv2.LINE_AA )
    cv2.imshow('frame', frame)

    # the 'q' button is set as the
    # quitting button you may use any
    # desired button of your choice
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# After the loop release the cap object
vid.release()
# Destroy all the windows
cv2.destroyAllWindows()
```

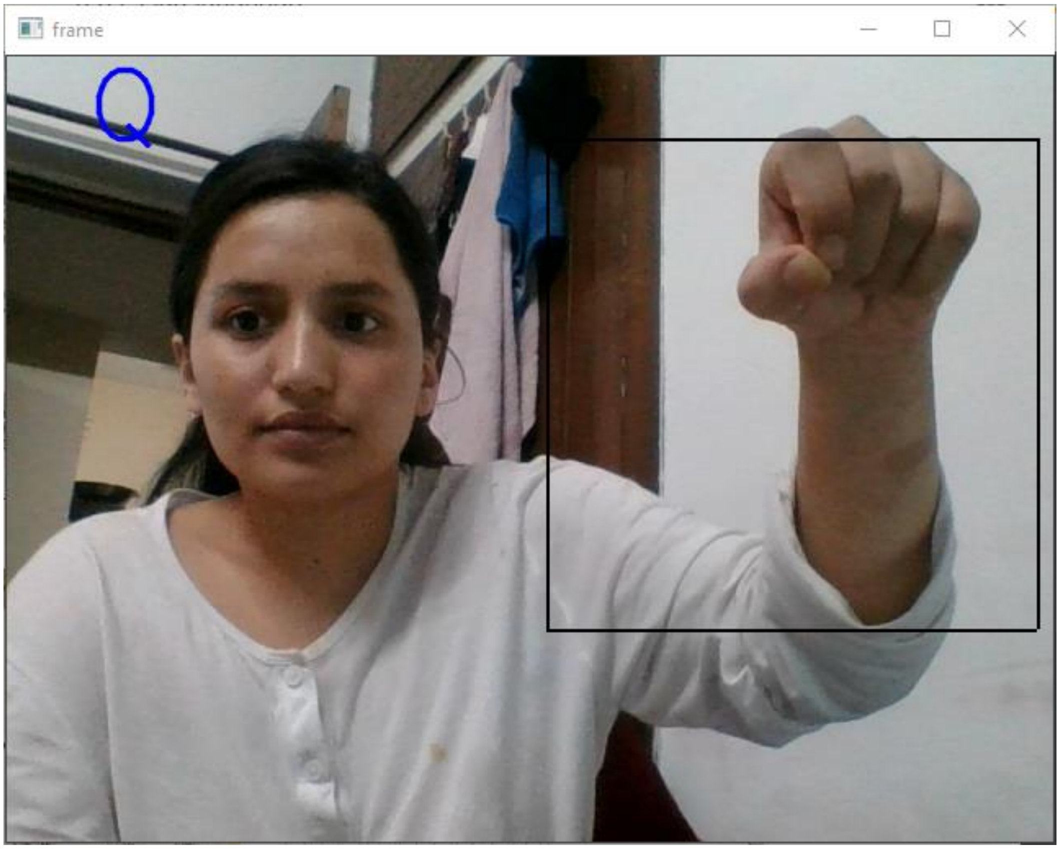
4.2 Output Snapshots:

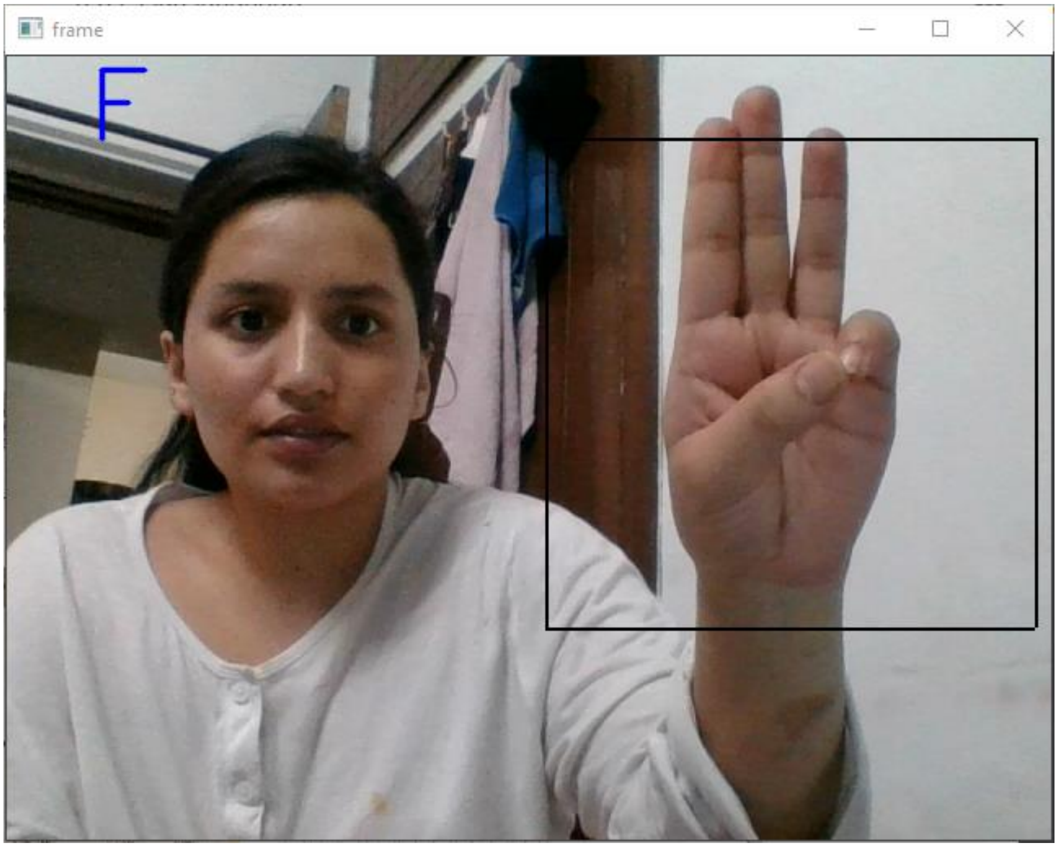


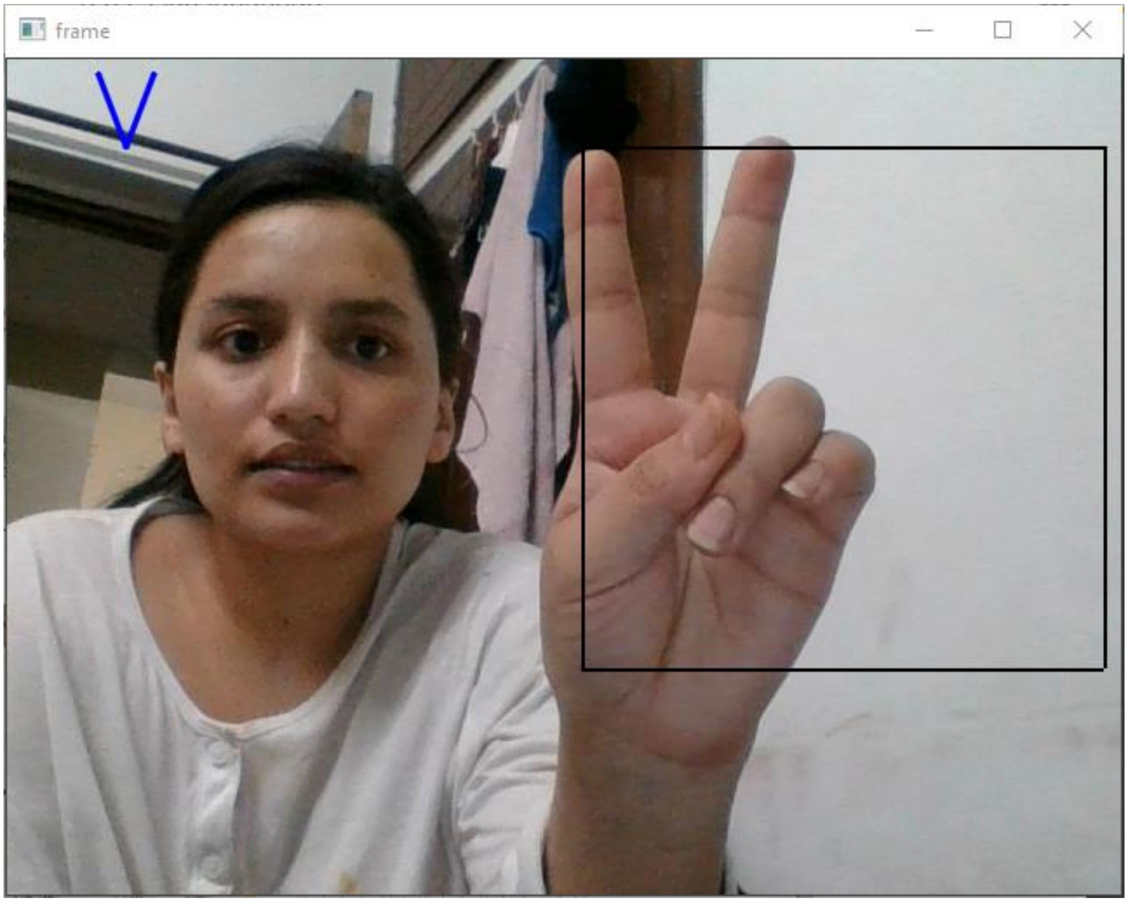



frame
R


frame
W


frame
Q


frame
F


frame
V

CONCLUSION

From this project/application I have tried to overshadow some of the major problems faced by the disabled persons in terms of talking. I found out the root cause of why they can’t express more freely. The result that I got was the other side of the audienceare not able to interpret what these persons are trying to say or what is the message that they want to convey.

Thereby this application serves the person who wants to learn and talk in sign languages.With this application a person will quickly adapt various gestures and their meaning as per ASL standards. They can quickly learn what alphabet is assigned to which gesture. Add-on to this custom gesture facility is also provided along with sentence formation. Auser need not be a literate person if they know the action of the gesture, they can quicklyform the gesture and appropriate assigned character will be shown onto the screen.

Concerning to the implementation, I have used TensorFlow framework, with keras API. Appropriate user-friendly messages are prompted as per the user actions along with whatgesture means which character window.

Additionally, an export to file module is also provided with TTS(Text-To-Speech) assistance meaning whatever the sentence was formed a user will be able to listen to it and then quickly export along with observing what gesture he/she made during the sentence formation.

REFERENCE

[1] M. Ibrahim, “Sign Language Translation via Image Processing”, https://www.kics.edu.pk/project/startup/203

[2] NAD, “American sign language-community and culture frequently asked questions”, 2017 https://www.nad.org/resources/american-sign-language/community-and-culture

[3] Sanil Jain and K.V.Sameer Raja, “Indian Sign Language Character Recognition” , https://cse.iitk.ac.in/users/cs365/2015/_submissions/vinsam/report.pdf