REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

Submitted By

TEAM ID : PNT2022TMID45687

TEAM LEADER: KANAGA O (812919104013)

TEAM MEMBER 1: JANAPRIYA N (812919104011)

TEAM MEMBER 2: GAYATHRI V (812919104009)

TEAM MEMBER 3: SHALINI D (812919104028)

CHAPTER NO. TITLE

1. INTRODUCTION

1.1 Overview

People get to know one another by sharing their ideas, thoughts, and experiences with those around them. There are numerous ways to accomplish this, the best of which is the gift of "Speech." Everyone can very convincingly transfer their thoughts and understand each other through speech. It will be unjust if we overlook those who are denied this priceless gift: the deaf and dumb. In such cases, the human hand has remained the preferred method of communication.

1.2 Purpose

The project's purpose is to create a system that translates sign language into a human understandable language so that ordinary people may understand it.

2. LITERATURE SURVEY

2.1 Existing problem

Some of the existing solutions for solving this problem are:

Technology

One of the easiest ways to communicate is through technology such as a smart phone or laptop. A deaf person can type out what they want to say and a person who is blind or has low vision can use a screen reader to read the text out loud. A blind person can also use voice recognition software to convert what they are saying in to text so that a person who is Deaf can then read it.

Interpreter

If a sign language interpreter is available, this facilitates easy communication if the person who is deaf is fluent in sign language. The deaf person and person who is blind can communicate with each other via the interpreter. The deaf person can use sign language and the interpreter can speak what has been said to the person who is blind and then translate anything spoken by the blind person into sign language for the deaf person. Just Speaking

2.2 References

1. TITLE:

Innovative study of an AI voice based smart device to assist deaf people

AUTHOR:

Dhaya Sindhu Battina

YEAR:

2021 Assistive technology consists of a wide range of hardware and software tools that enable a person to receive information in the format that suits their needs best. These Various technology may be available to the deaf.. . Many items, including cochlear implants, loop systems, accessibility, FM technology, and assistive listening devices, visual warning systems, videophones, and much more . Recognizing the worth and boundaries of different assistive devices can be advantageous for both. Artificial intelligence (AI) enables computers to learn from existing experiences, adapt to new information, and perform tasks that are similar to those carried out by humans . The vast majority of artificial intelligence applications that users know of today – ranging from chess playing robots to

selfdriving vehicles – are primarily reliant on deep learning and computational linguistics .Computers may be taught to do particular jobs by Real time communication system powered by AI processing huge quantities of data and detecting trends in the data. This is accomplished via the use of various technologies Machine learning technologies have the potential to substantially reduce the communication obstacles that deaf or hearing-impaired persons have when interacting with other groups, thus promoting social inclusion for these individuals. Recent advancements in both sensing technology and artificial intelligence algorithms have opened the way for the creation of a broad array of applications aimed at meeting the requirements of the deaf and hearing-impaired populations.

2. TITLE:

Communication system for deaf and dumb people

AUTHOR:

Shraddha R. Ghorpade, Prof. Surendra K. Waghmare2

YEAR:

2019 People with disabilities are having a difficult time keeping up with the rapidly evolving technology, which is one of the major issues that our society is dealing with. For those with disabilities, having access to communication tools has become crucial. typically deaf and stupid people use sign language to communicate, but they struggle to do so with non-sign language users language. Information is the main topic of communication between normal and deaf individuals using sign language, which is expressive and natural. So that we can converse with them and comprehend what they're saying, we need a translation. A language translation technology converts common sign language into voice, enabling regular people to communicate with one another. When it comes to communicating with other people, sign language (SL) is the primary method of communication for hearing-impaired individuals and other groups. It is conveyed via both manual (body and hand movements) and non-manual (face expressions) characteristics. All of these characteristics are combined to create utterances that communicate the meaning of words or statements. Understanding and being able to record and comprehend the relationship between utterances and words is critical for the Deaf community's ability to lead us toward a time when automated translating between utterances and words is possible. In recent years, researchers have recognized the need for the development of sign language technology to assist hearing-impaired individuals in communicating and socially integrating into their communities. Even though the advancement of such innovations can be extremely difficult owing to the inclusion of multiple sign languages and a scarcity of large annotated datasets, recent advances in artificial intelligence and machine learning have made significant strides towards automating and improving such systems. Keywords: Image Processing, Human Computer Interface (HCI).

3. TITLE:

Educational Status of Differently Abled Persons and Developed Policies in India

AUTHOR:

Chiranjit Majumder

YEAR:

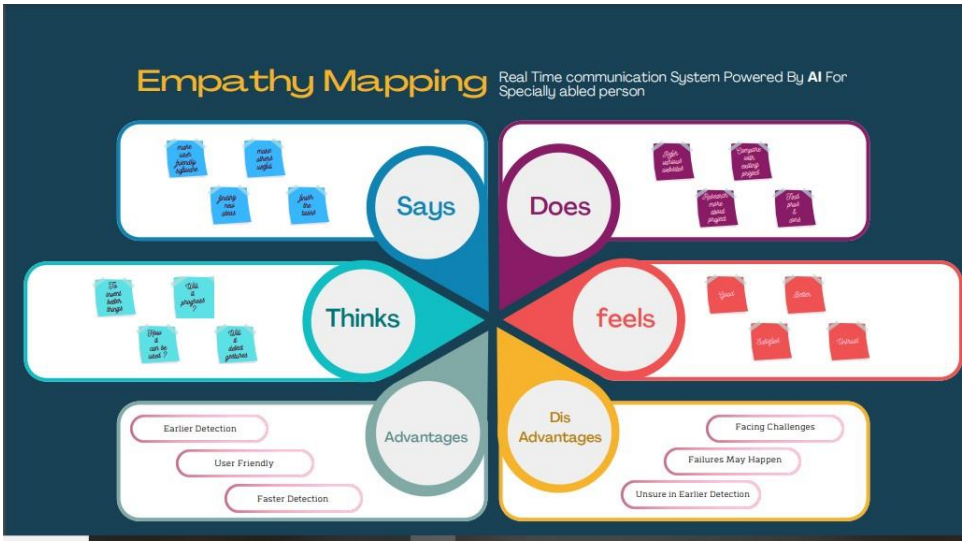
2019 April One of the socially created phenomenon is basically Disability. The fact is that many children and adults suffered from disabilities excluded from mainstream education benefits. Real time communication system powered by AI Disabled persons are segregated from education system because of social negligence and absence of support system in the home and inadequacy of sufficient facilities in schools particularly. However, education is the most important medium for social, economic and political transformation. Socialization of children with disabilities (CWD) through education receives an unremarkably important roles in societies such as India where social exclusion of Physically Challenged Persons (PCPs) is significant. Indisputably, the literacy level of Physically Challenged Persons (PCPs) is very low in India. Very poor educational outcomes for children with disabilities remain in developing countries specially. Most of disabled persons do not get the full benefits of education. However, some policies in India has started to display some concern for Physically Challenged students. Education is utmost significant to lift up the socio-economic status of PCPs. But education of disabled persons has not received adequate intentness and resources that it requires. Physically Challenged Persons (PCPs), few who are enrolled in schools are not given equal opportunity for middle secondary and higher education levels. Many Disabled persons are educated but they do not get any work for earning in our society. However, in India the existing situation began to change. Indian policies has started to understand as for all people that education is essential for children and adults with IJTSRD21762 International Journal of Trend in Scientific Research and Development (IJTSRD) @ www.ijtsrd.com eISSN: 2456-6470 @ IJTSRD | Unique Paper ID - IJTSRD21762 | Volume – 3 | Issue – 3 | Mar-Apr 2019 Page: 342 disabilities in itself and helpful for participating in employment and other sites of social activity. The Ministry of Human Resource Development (MHRD) has initiated various programmes to give educational opportunities to PCPs in an environment that is inclusive (Ghoshal S.K., 2018). Government of India also understands the needs of appropriate vocational training skills to make them self faithful and productive members of society. But, the scheme coverage has stayed limited. Non Governmental Organizations (NGOs) has a biggest role to improve the life of disabled persons in our society. Educational Status of Disabled Persons in India India is the world's largest democracy. India has a countless challenges for ensuring access to education for over all 200 million (20 Cr) children aged 6 to 13 years. As per 2011 National Census, 1.05% of school going children have a problem like disability (2.13 million = 21 Lakhs 30 Thousand); of these 28% (5 Lakhs 88 Thousand) are not accessing school. Particularly 44% of disabled children are not accessing school have complex and multiple forms of activity limitations and functioning difficulties (Bakhshi et al, 2017). Education is an fruitful tool for socioeconomic empowerment which can develop the career of specially disabled ones. disabilities in itself and helpful for participating in employment and other sites of social activity. The Ministry of Human Resource Development (MHRD) has initiated various programmes to give educational opportunities to PCPs in an environment that is inclusive (Ghoshal S.K., 2018). Government of India also understands the needs of appropriate vocational training skills to make them self faithful and productive members of society. But, the scheme coverage has stayed limited. Non Governmental Organizations (NGOs) has a biggest role to improve the life of disabled persons in our society. Educational Status of Disabled Persons in India

2.3 Problem statement Definition

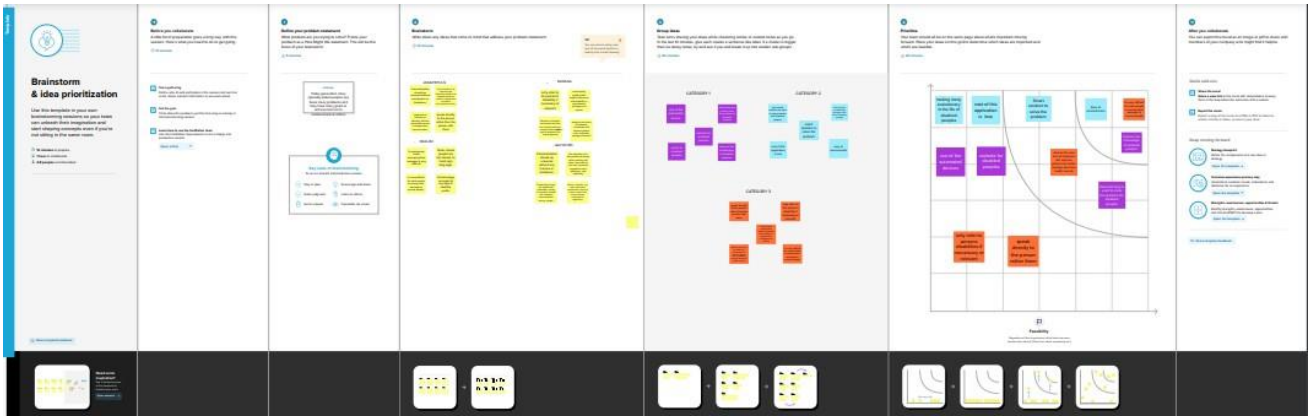
The solution to the issue that the speech and hearing challenged encounter is outlined in this research. The goal of the research is to create a system that narrows the communication gap between speech- and hearing-impaired people and the rest of society.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

1. Problem Statement (Problem to be Solved):

To model a system for aiding deaf and dumb people and help them to communicate in realtime.

2. Idea / Solution description:

The model uses a technique called Background-subtraction which is considered as amajor preprocessing step. There are two methods available in the literature for removing the background to extract the

foreground object. We have planned to employ these methods in the design of the proposed Sign Language Converter for identifying the hand region in the input image captured by the camera.

3. Novelty / Uniqueness:

We will be using the latest and trending wearable technology which makes it possible to carry the device (Mobile Application) easily anywhere and everywhere by the disabled person which makes the communication possible by both specially abled and normal people. We will be using the most recent convolution neural network architecture to improvethe efficiency of the trained model.

4. Social Impact / Customer Satisfaction: Helps to bridge the gaps in communication with hearing and speaking impaired people.

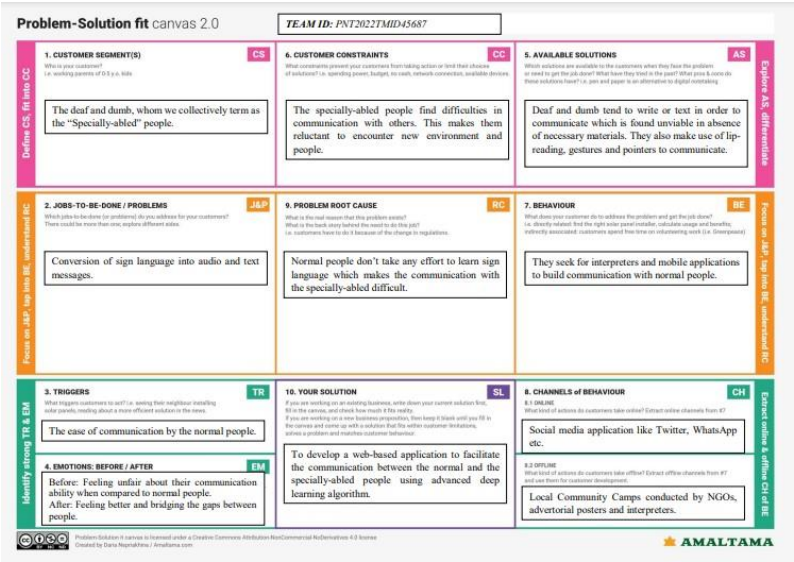
5. Business Model (Revenue Model):

The implemented end product will be marketed as a Retailer model, in which the product will be assigned an initial base price and will be updated once we bring new features to it.

6. Scalability of the Solution:

Bootstrapping the company at first through the founder's funds, but eventually through reinvesting the profit from servicing customers.

3.4 Problem Solution Fit



4.REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirement	Converting sign language into speech that can be understand by normal people using an application.
FR-2	User Registration	Manual Sign up using the application or Gmail.
FR-3	User Confirmation	OTP authentication through phone messages, email, notices, paper and confirmation.
FR-4	Product Implementation	Install the dataset to recognise and translate hand gestures and voice for the real-time communication by using the application.
FR-5	Payment Option	Bank transfer, Debit cards, UPI method, if pro version required.
FR-6	Product Feedback	Through the application, phone conversation and Gmail.

4.2 Non-Functional Requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is used to describe the application and easy to access the application with the guidelines.
NFR-2	**Security**	It ensures the security of the application by building a firewall and two step verification support. Accessed only by authorised person by given user ID and password or OTP verification.
NFR-3	**Reliability**	To maintain the application conditions and update the version of the application. System update and software update are possible to increase various features and durability based on technology.
NFR-4	**Performance**	This application collects the datasets of hand gestures to provide accurate prediction. Using this method, we can communicate easily at anytime. This application is user friendly and can be accessed by both specially abled and normal people.
NFR-5	**Availability**	Depending on the requirements of the user, all required functions will be offered. When the user requests any features, the features are made available in places where users like to know about it.
NFR-6	**Scalability**	As based on application, real-time communication is accessed on a compatible devices. The application is based on voice conversion system, hand gesture recognition and translation.

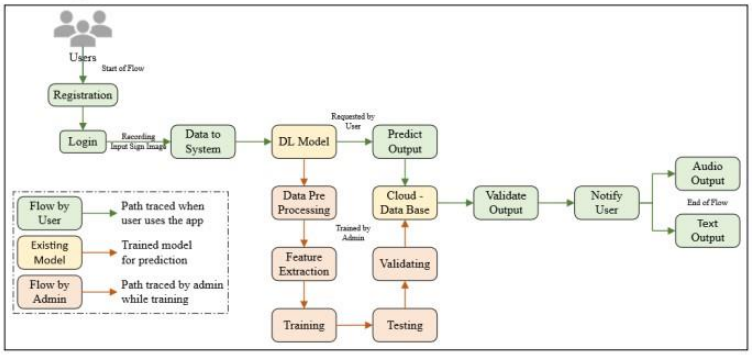
5. PROJECT DESIGN

5.1 Data Flow Diagrams

Project Design Phase-II
Data Flow Diagram & User Stories

Date	30 October 2022
Team ID	PNT2022TMID45687
Project Name	Project: Real-Time Communication System Powered by AI for Specially-Abled
Maximum Marks	4 Marks

Data Flow Diagram:



5.2 Solution & Technical Architecture

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint - 1
	Authentication	USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm	Low	Sprint - 1
	Login	USN-3	As a user, I can log into the application by entering email & password.	I am able to get into the Dashboard	High	Sprint - 2
	Dashboard	USN-4	One place to explore all available features.	I can access my dashboard	High	Sprint - 2
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint - 1
	Authentication	USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm.	Low	Sprint - 1
	Login	USN-3	As a user, I can log into the application by entering email & password	I am able to get into the Dashboard	Low	Sprint - 2
	Dashboard	USN-4	One place to explore all available features.	I can access my dashboard	Low	Sprint - 2
	Upload image	USN-5	As a user, I can upload the sign language image for translating into text format	I can be able to see the appropriate text for the sign language	High	Sprint - 3
Administrator	Manage	USN-6	Do-it-yourself service for delivering Everything.	Set of predefined requirements that must be met to mark a user story complete	High	Sprint - 4

5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Register with the users information.	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard in the application.	High	Sprint-1
Customer (Deaf people)	To communicate with people using signs.	USN-2	As a user, I can see my application and made changes in any browser and register to it.	I can login and see my account in the application anywhere at anytime.	High	Sprint-1
Customer (Dumb people)	To communicate with people easily and efficiently.	USN-3	As a user, I can see my application and made changes in any browsers and register to it.	I can login and see my account in the application anywhere.	High	Sprint-1
Customer (Normal people)	User needs to communicate with specially abled people.	USN-4	As a user, I can register for the application by entering my email, password, and confirmation is made.	I can login and see my account.	Medium	Sprint-2
Customer (Learner of Sign language)	User needs to be aware and learn about sign language.	USN-5	As a user, I can create my account in the application with my email and password, to get knowledge about sign languages.	I can create my account and access the dashboard in the application.	High	Sprint-1
Customer (Web user)	They want the update on the application condition.	USN-6	As a user, I can register for the application by entering my email, password, and confirming my password. To get details about real-time communication.	I can able to use any browser to access the application from anywhere, to know anything about real-time communication.	High	Sprint-1
Customer Care Executive	They want to help people by sending application conditions.	USN-7	As a user, I can receive a message from the administration about conditions of application of real-time communication.	I will analyse and send SMS to the people.	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect theDataset .	10	High	KANAGA O JANAPRIYA N GAYATHRI V SHALINI D
Sprint-1		USN-2	Image preprocessing	10	Medium	KANAGA O JANAPRIYA N GAYATHRI V SHALINI D
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	KANAGA O JANAPRIYA N GAYATHRI V SHALINI D

Sprint-2		USN-4	Training the image classification modelusing CNN	10	Medium	KANAGA O JANAPRIYA N
Sprint-3	Training and Testing	USN-5	Training the model and testing the model's performance	10	High	GAYATHRI V SHALINI D
Sprint-4	Implementation of the application	USN-6	Converting the input sign language images into English alphabets	10	Medium	KANAGA O JANAPRIYA N GAYATHRI V SHALINI D

Use the below template to create product backlog and sprint schedule

6.2 Sprint Delivery Schedule

ProjectTracker, Velocity& BurndownChart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	28 Oct 2022
Sprint-2	10	6 Days	27Oct 2022	04 Nov 2022	10	30Oct 2022
Sprint-3	10	6 Days	07 Nov 2022	31Oct 2022	10	7 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	8 Nov 2022	10	18 Nov 2022

Velocity:

$$AV = \frac{sprint\ duration}{velocity}$$

AV = 6/10 = 0.6

7.CODING & SOLUTIONING

7.1 Feature 1

1. It will contribute to the development of improved communication for the deafened. The majority of people are unable to communicate via sign language, which creates a barrier to communication.

7.2 Feature 2

1. As a result, others will be able to learn and comprehend sign language and communicate with the deaf and dumb via the web app.

2. According to scientific research, learning sign language improves cognitive abilities, attention span, and creativity

8.TESTING

Importing the packages.

In [2]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [3]:
```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

linking the streaming_body

In [4]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='Izw27RyYAXpOXEjMhc04K038iQoczlOKd5DbxFulpmhc',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'realtimecommunication-donotdelete-pr-e7yebdl9hvsfug'
object_key = 'Dataset.zip'

streaming_body_4 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

unzipping the dataset

In [6]:
```
# Unzip the Dataset Zip File
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_4.read()), 'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [7]:
```
%%bash
ls Dataset
```

```
test_set
training_set
```

In [8]:
```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/home/wsuser/work/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/home/wsuser/work/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

In [9]:
```
print("Length of x-train : ", len(x_train))
print("Length of x-test : ", len(x_test))
```

```
Length of x-train :  18
Length of x-test :  3
```

In [10]:
```
# The Class Indices in Training Dataset
x_train.class_indices
```

Out[10]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

Model Creation

In [11]:
```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

In [12]:
```
# Creating Model
model=Sequential()
```

In [13]:
```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

# Adding Hidden Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))

# Adding Output Layer
model.add(Dense(9,activation='softmax'))
```

In [14]:
```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=["accuracy"])
```

In [15]:
```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
/tmp/wsuser/ipykernel_164/1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Mod
el.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
Epoch 1/10
18/18 [==============================] - 71s 4s/step - loss: 1.1343 - accuracy: 0.6152 - val_loss: 0.4147 - val_accuracy: 0.8958
Epoch 2/10
18/18 [==============================] - 69s 4s/step - loss: 0.2602 - accuracy: 0.9239 - val_loss: 0.2582 - val_accuracy: 0.9028
Epoch 3/10
18/18 [==============================] - 71s 4s/step - loss: 0.1165 - accuracy: 0.9669 - val_loss: 0.2227 - val_accuracy: 0.9687
Epoch 4/10
18/18 [==============================] - 72s 4s/step - loss: 0.0618 - accuracy: 0.9846 - val_loss: 0.2374 - val_accuracy: 0.9698
Epoch 5/10
18/18 [==============================] - 73s 4s/step - loss: 0.0357 - accuracy: 0.9902 - val_loss: 0.2313 - val_accuracy: 0.9707
Epoch 6/10
18/18 [==============================] - 72s 4s/step - loss: 0.0205 - accuracy: 0.9939 - val_loss: 0.2498 - val_accuracy: 0.9756
Epoch 7/10
18/18 [==============================] - 71s 4s/step - loss: 0.0170 - accuracy: 0.9965 - val_loss: 0.2795 - val_accuracy: 0.9756
Epoch 8/10
18/18 [==============================] - 72s 4s/step - loss: 0.0119 - accuracy: 0.9977 - val_loss: 0.2573 - val_accuracy: 0.9769
Epoch 9/10
18/18 [==============================] - 72s 4s/step - loss: 0.0095 - accuracy: 0.9981 - val_loss: 0.2782 - val_accuracy: 0.9782
Epoch 10/10
18/18 [==============================] - 72s 4s/step - loss: 0.0075 - accuracy: 0.9987 - val_loss: 0.3134 - val_accuracy: 0.9764
```

Out[15]:

Saving the Model

In [16]:
```
model.save("SANJAI.h5")
# Current accuracy is 0.825
```

In [17]:
```
# Convert the Saved Model to a Tar Compressed Format
!tar -zcvf trainedModel.tgz SANJAI.h5
```

```
SANJAI.h5
```

```
In [18]:
%%bash
ls -ll
```

```
total 210000
drwxrwx--- 4 wsuser wscommon      4096 Nov 16 19:02 Dataset
-rw-rw---- 1 wsuser wscommon 111324760 Nov 16 19:15 SANJAI.h5
-rw-rw---- 1 wsuser wscommon 103709312 Nov 16 19:15 trainedModel.tgz
```

Watson Machine Learning

```
In [19]:
!pip install watson-machine-learning-client --upgrade
```

```
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
     |████████████████████████████████| 538 kB 8.4 MB/s eta 0:00:01
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learni
ng-client) (0.5.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learn
ing-client) (1.21.41)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning
-client) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21
->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.
0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-ma
chine-learning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-
learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-clien
t) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-le
arning-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
(2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
(1.20.3)

Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

```
In [20]:
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "_c04HIUdEc7WeOodIn0_1IgBFNO413OAAIocQOhaWCI"
}

client = APIClient(wml_credentials)
```

```
In [21]:
client
```

```
Out[21]:
```

Save to Deployment Space

```
In [22]:
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return (next(item for item in space['resources'] if item['entity']["name"] == space_name)['metadata']['id'])
```

```
In [23]:
space_uid = guid_from_space_name(client, 'Real_Time')
print("Space UID : ", space_uid)
```

```
Space UID :  48cfd02c-38d1-4f55-b4a6-c1b77eab8cf5
```

```
In [24]:
client.set.default_space(space_uid)
```

```
Out[24]: 'SUCCESS'
```

In [25]:
```
client.software_specifications.list()
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6              4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
pytorch-onnx_1.7-py3.8          5d4d3cdc-05b2-50f9-a204-ea99a4784560  base
------------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

In [26]:
```
software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
```

Out[26]: `'acd9c798-6974-5d2f-a657-ce06e986df4d'`

In [27]:
```
model_details = client.repository.store_model(model='trainedModel.tgz', meta_props={
    client.repository.ModelMetaNames.NAME: "CNN",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid,
    client.repository.ModelMetaNames.TYPE: "tensorflow_2.7"})
model_id = client.repository.get_model_id(model_details)
```

In [28]:
```
model_id
```

Out[28]: `'4154aedd-4fff-46f6-b056-ea4e566d3643'`

In [29]:
```
client.repository.download(model_id,'SANDAI1.tar.gz')
```

```
Successfully saved model content to file: 'SANDAI1.tar.gz'
```

Out[29]: `'/home/wsuser/work/SANDAI1.tar.gz'`

TESTING PART OF MODEL

In [30]:
```
#Testing the model.
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

In [31]:
```
model=load_model('SANDAI.h5')
img=image.load_img(r'/home/wsuser/work/Dataset/test_set/A/1.png',
                   target_size=(64,64))
```

In [24]:
```
img
```

Out[24]:

In [33]:
```
img=image.load_img(r"/home/wsuser/work/Dataset/test_set/A/90.png",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['A','B','C','D','E','F','G','H','I']
index[y[0]]
```

Out[33]: 'A'

In [34]:
```
img=image.load_img(r"/home/wsuser/work/Dataset/test_set/C/90.png",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['A','B','C','D','E','F','G','H','I']
index[y[0]]
```

Out[34]: 'C'

In [35]:
```
img=image.load_img(r"/home/wsuser/work/Dataset/test_set/I/90.png",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['A','B','C','D','E','F','G','H','I']
index[y[0]]
```

Out[35]: 'I'

In [36]:
```
img=image.load_img(r"/home/wsuser/work/Dataset/test_set/E/90.png",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['A','B','C','D','E','F','G','H','I']
index[y[0]]
```

Out[36]: 'E'

In [37]:
```
img=image.load_img(r"/home/wsuser/work/Dataset/test_set/F/90.png",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['A','B','C','D','E','F','G','H','I']
index[y[0]]
```

Out[37]: 'F'

Accuracy is over 90+ percentage because of the overfitting phenomenon,when we test our model with live data then the accuracy will decrease.

Spyder Deployment Code

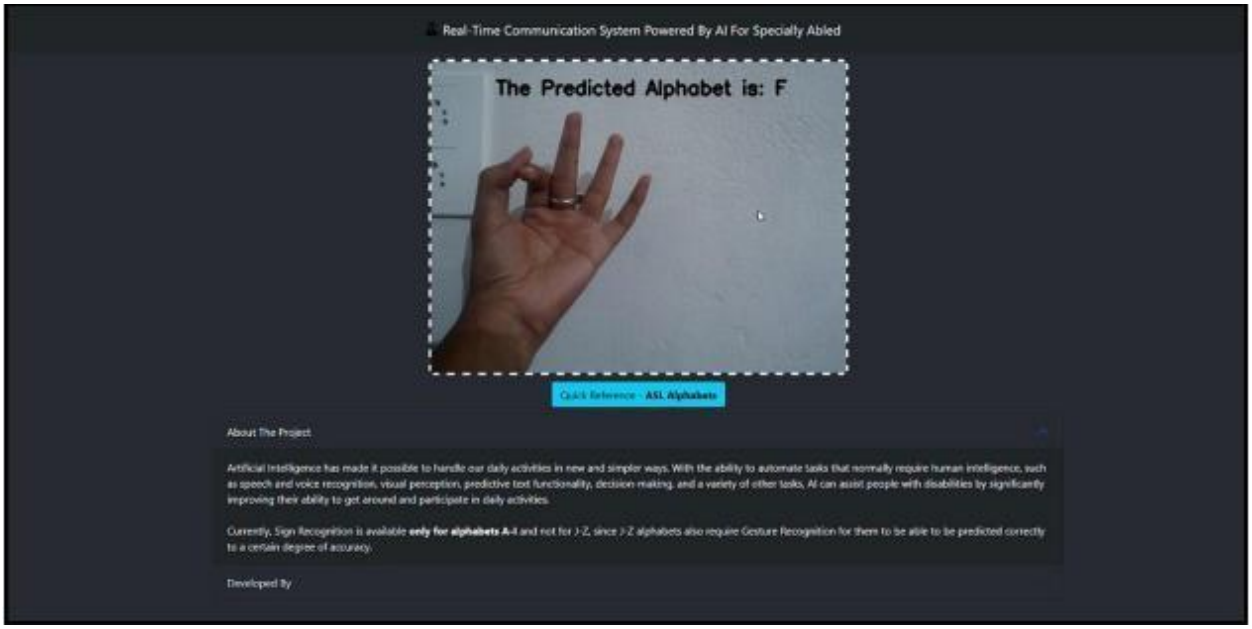
In [ ]:
```
import cv2 #importing opencv library this I to open camera and take the video
import numpy as np # to convert image to array and expand dimensions
from tensorflow.keras.models import load_model # to load the saved model
from tensorflow.keras.preprocessing import image # to preprocess the image
model = load_model("dataset.h5") # we are loading the saved moodel
video = cv2.VideoCapture(0) # two parameters 1, bool 0 or 1, frame
index = ["A","B","C","D","E","F","G","H","I"]
index=['A','B','C','D','E','F','G','H','I']
#from playsound import playsound
while(1):
    success,frame = video.read()
    cv2.imwrite("image.jpg",frame)
    img = image.load_img("image.jpg",target_size = (64,64))
    x = image.img_to_array(img)
    X = np.expand_dims (x,axis = 0)
    pred = np.argmax(model.predict(x),axis=1)
    p = index [pred[0]]
    print("predicted letter is: "+ str(p))
    #playsound("letter"+str(str(index [p])+"is detected"))
    cv2.putText (frame, "predicted letter is "+str(p), (100, 100), cv2. FONT_HERSHEY_SIMPLEX, 1,(0,0,0), 4)
    cv2.imshow("showcasewindow", frame)

    if cv2.waitKey(1) & 0xFF == ord('a'):
        break
video.release()
cv2.destroyAllWindows()
```

9.RESULTS

9.1 Performance metrics

The proposed procedure was implemented and tested with set of images. The set of 15750 images of Alphabets from “A” to “I” are used for training database and a set of 2250 images of Alphabets from “A” to “I” are used for testing database. Once the gesture is recognise the equivalent Alphabet is shown on the screen. Some sample images of the output are provided below:



10.ADVANTAGES & DISADVANTAGES

Advantages:

1. It is possible to create a mobile application to bridge the communication gap between deaf and dumb persons and the general public.
2. 2. As different sign language standards exist, their dataset can be added, and the user can choose which sign language to read.

Disadvantages:

1. The current model only works from alphabets A to I.

2. In absence of gesture recognition, alphabets from J cannot be identified as they require some kind of gesture input from the user.

3. As the quantity/quality of images in the dataset is low, the accuracy is not great, but that can easily be improved by change in dataset.

11.CONCLUSION

Sign language is a useful tool for facilitating communication between deaf and hearing people. Because it allows for two-way communication, the system aims to bridge the communication gap between deaf people and the rest of society. The proposed methodology translates language into English alphabets that are understandable to humans.

This system sends hand gestures to the model, who recognises them and displays the equivalent Alphabet on the screen. Deaf-mute people can use their hands to perform sign language, which will then be converted into alphabets, thanks to this project.

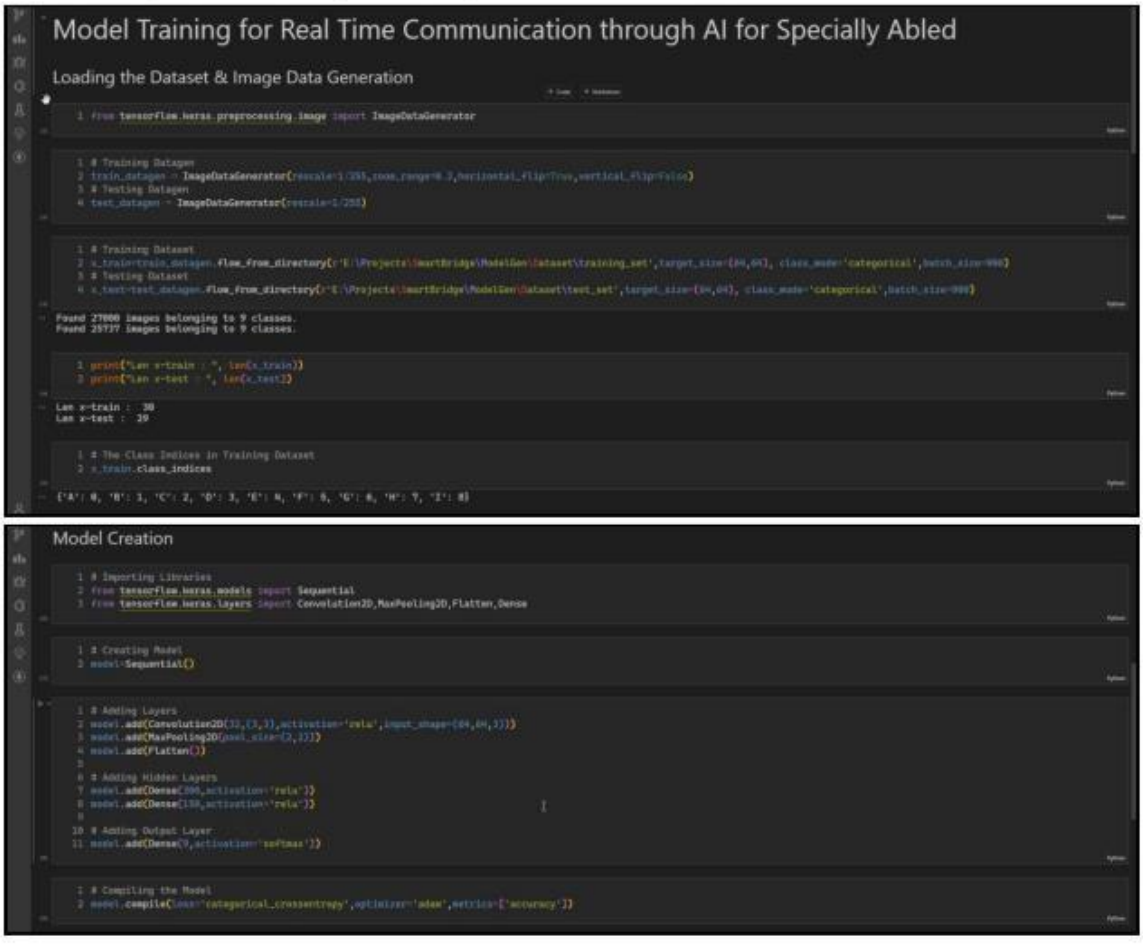
12.FUTURE SCOPE

Having a technology that can translate hand sign language to its corresponding alphabet is a game changer in the field of communication and Ai for the specially abled people such as deaf and dumb. With introduction of gesture recognition, the web app can easily be expanded to recognize letters beyond 'I', digits and other symbols plus gesture recognition can also allow controlling of software/hardware interfaces.

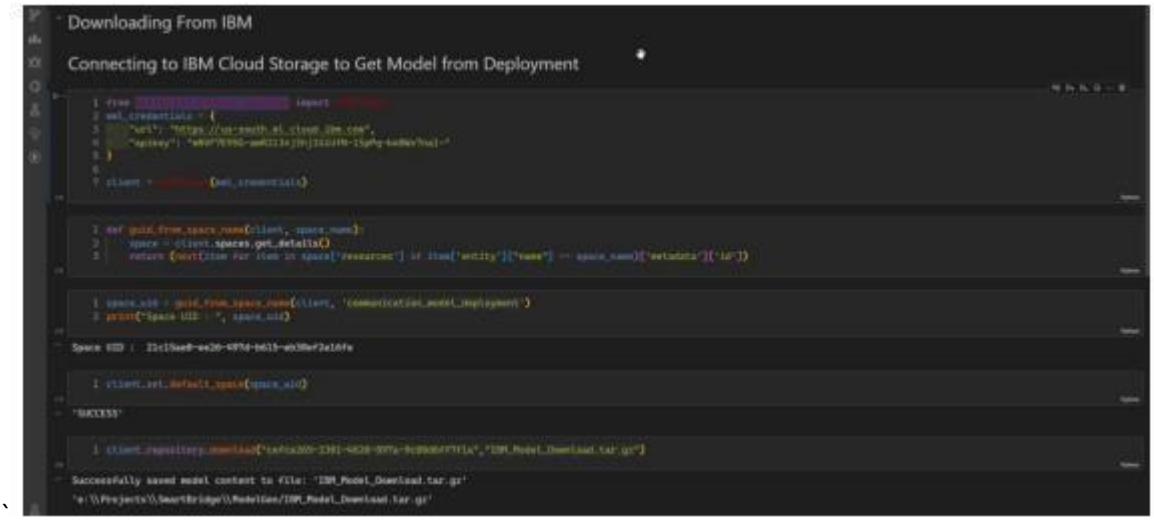
13.APPENDIX

Source Code

Source Code for Model Training and Saving:

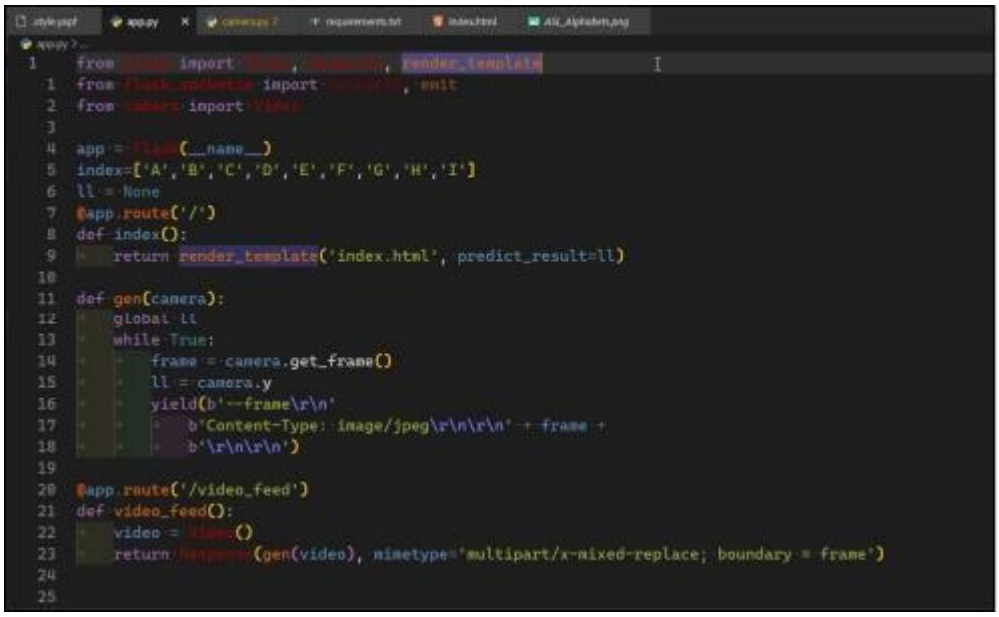


IBM Model Training & Download Code:



`

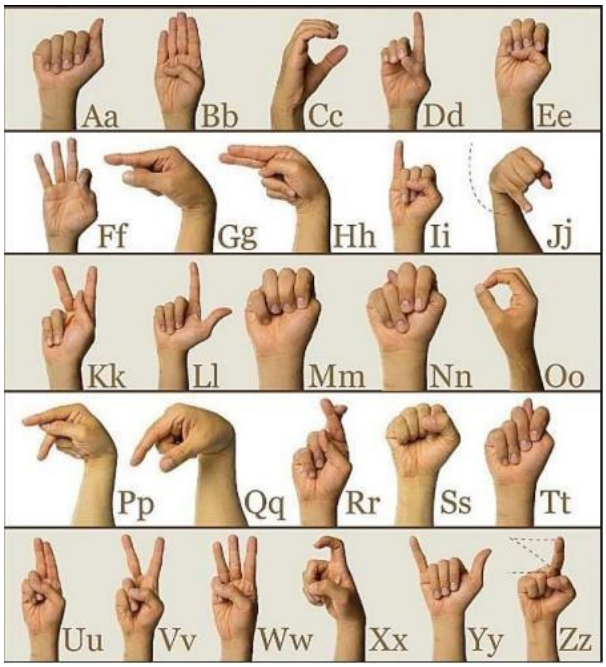
Web app code



```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        # self.model = load_model('asl_model.h5') # Execute Local Trained Model
        self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        # copy_img = image.load_img('image.jpg', target_size=(28,28))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),
        cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

American sign language standard references



GitHub link : https://github.com/IBM-EPBL/IBM-Project-12663-1659457225

Project demo link :

https://drive.google.com/file/d/1Lld5r3qL7mkkzwvVWblUzfWBT89X9NXu/view?usp=share_link