Real-Time Communication System Powered by AI for Specially Abled

DOCUMENTATION

TEAM ID : PNT2022TMID06649

TEAM LEADER : BARATHKUMAR S R **(61072011902)**

TEAM MEMBERS : AJAY S **(61072011901)**

ANANTH KRISHNA A **(61071911104)**

ARISH KARTHIK P **(61071911106)**

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language. The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

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. In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

2. LITERATURE SURVEY

2.1 Existing problem

[1] AAWAAZ: A Communication System for Deaf & Dumb by Anchal Sood. Anju Mishra (2016)

The paper proposes a framework for recognizing hand gesture which would serve not only as a way of communication between deaf and dumb and mute people, but also, as an instructor. Deaf and dumb individuals lack in proper communication with normal people and find it difficult to properly express themselves. Thus, they are subjected to face many issues in this regard. The sign language is very popular among them and they use it to express themselves. Thus, there is a need of a proper translator. The deaf and dumb are not idle as past, they are working outside and doing great at it. So, an efficient system must be set up, to interact with them, to know their views and ideas. The framework here, act as a communication system for deaf and dumb individuals. It would take the sign language as an input which would display the result not only in the form of text but also in the form of audio. Similarly, if there is any input in the form of text, it would display the corresponding image.

Advantage:

 Early and accurate recognizing

Disadvantage:

 Lack of proper communication

[2] Full Duplex Communication System for Deaf & Dumb People by Shraddha R.

Ghorpade, Surendra K. Waghamare (2015)

One of the important problems that our society faces is that people with disabilities are finding it hard to cope-up with the fast-growing technology. The access to communication technologies has become

essential for the handicapped people. Generally deaf and dumb people use sign language for communication but they find difficulty in communicating with others who don't understand sign

Advantage:

 Fast Recogination

Disadvantage:

 Difficulty in communicating with others who don't understand sign

[3] Computer Technology Department, RTMNU, Nagpur, Maharashtra, India (2017)

An evolution of Information and Communication Technology has influenced every part of human life. It has modified the way we do the job, occupation, travel, acknowledge and convey. For the Deaf people group, the utilization ICT has enhance their personal satisfaction by creating frameworks that can help them discuss better with whatever remains of the world and among themselves. Gesture based communication is the essential method for correspondence in the almost totally impaired group. The issue emerges when hard of hearing individuals attempt to convey what needs be to other individuals with the assistance of these gesture based communication language structures and had habit a versa. The application gives hard of hearing individuals a method for getting more shut to cutting edge innovation by

utilizing discourse to picture interpretation. This deaf individual to learn new advances by looking toward pictures which are being changed over to pictures by utilizing discourse acknowledgment framework.

Advantage:

 Deaf peoples has enhance to utilize personal satisfaction.

Disadvantage:

 The issue emerges when hard of hearing individuals attempt to convey what needs be to other individuals

2.2 Problem Statement Definition

2.2.1 Problem Statement :

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output

2.2.2 Approach:

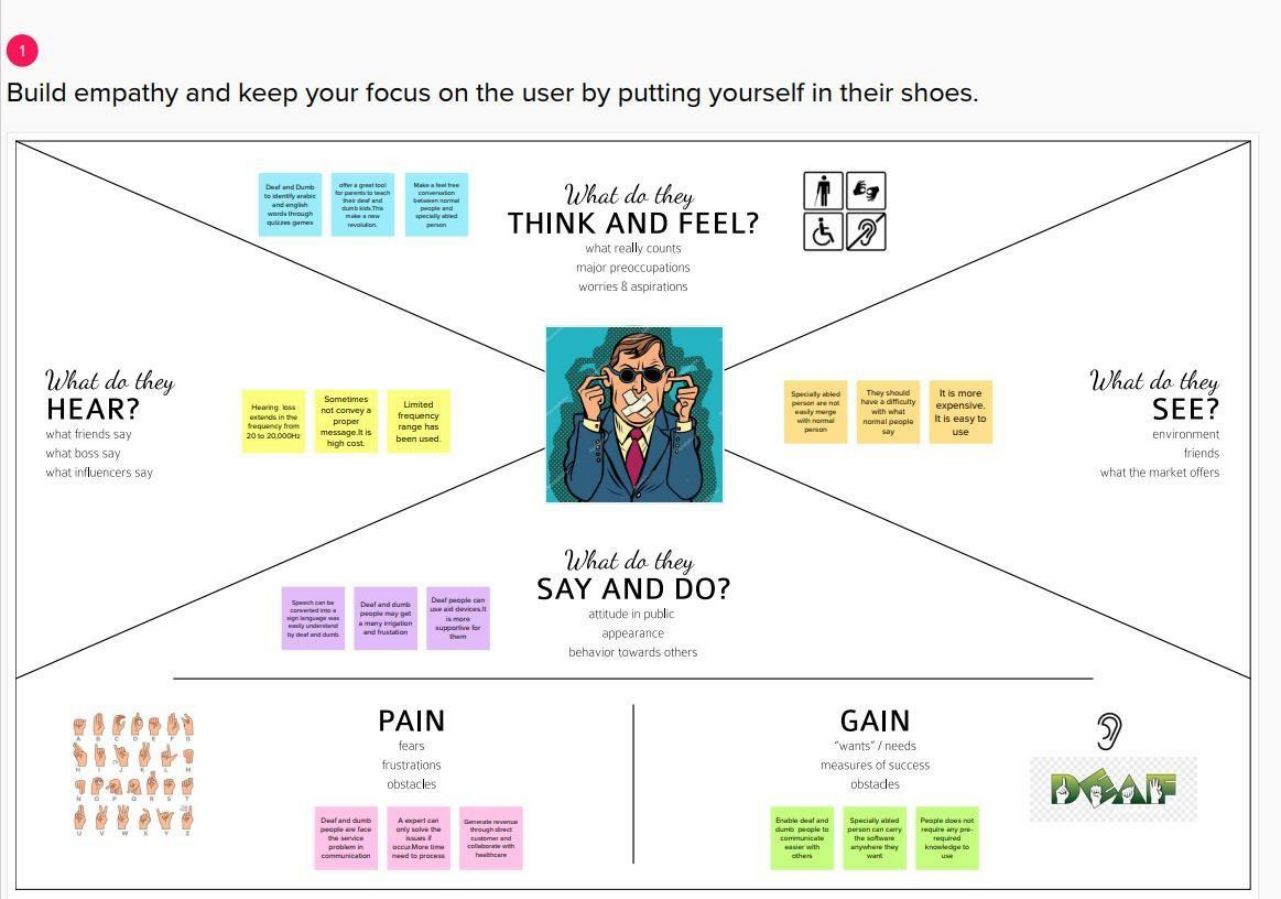
- Communication plays a significant role in making the world a better place. Most people communicate efficiently without any issues, but many cannot due to disability.
- They cannot hear or speak, which makes Earth a problematic place to live for them. Even simple basic tasks become difficult for them.
- Disability is an emotive human condition, Being deaf and dumb pushes the subject to oblivion, highly introverted.
- How artificial intelligence is being used to help people who are unable to do what most people do in their everyday lives.
- Technology should create a platform or a world of equality despite the natural state of humans.

2.2.3 Benefits:

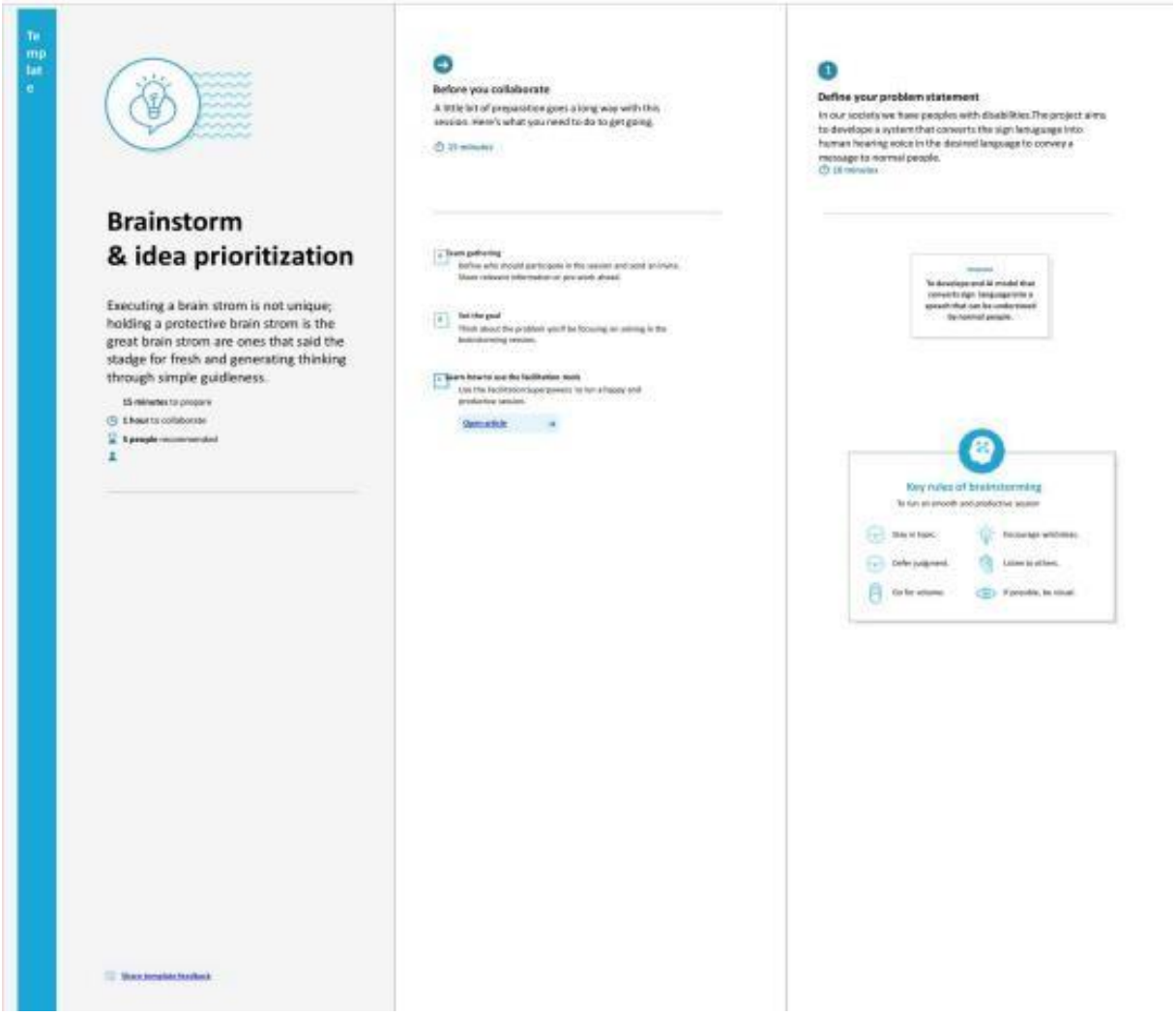
- To help people overcome physical and cognitive challenges .

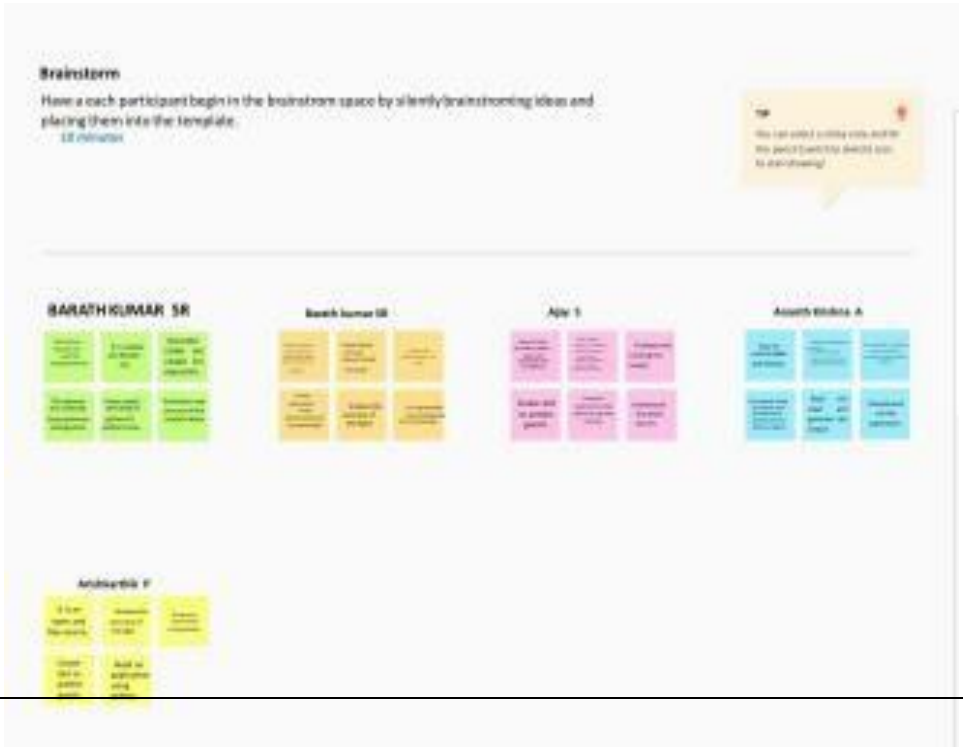
3. IDEATION & PROPOSED SOLUTION

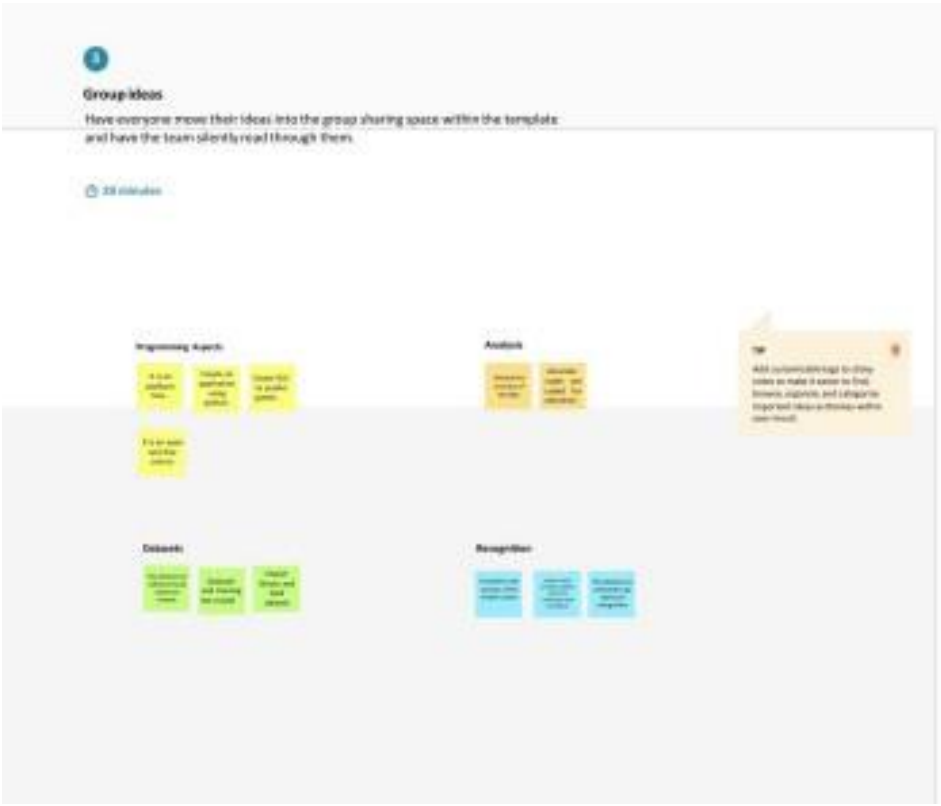
3.1 Empathy Map Canvas:

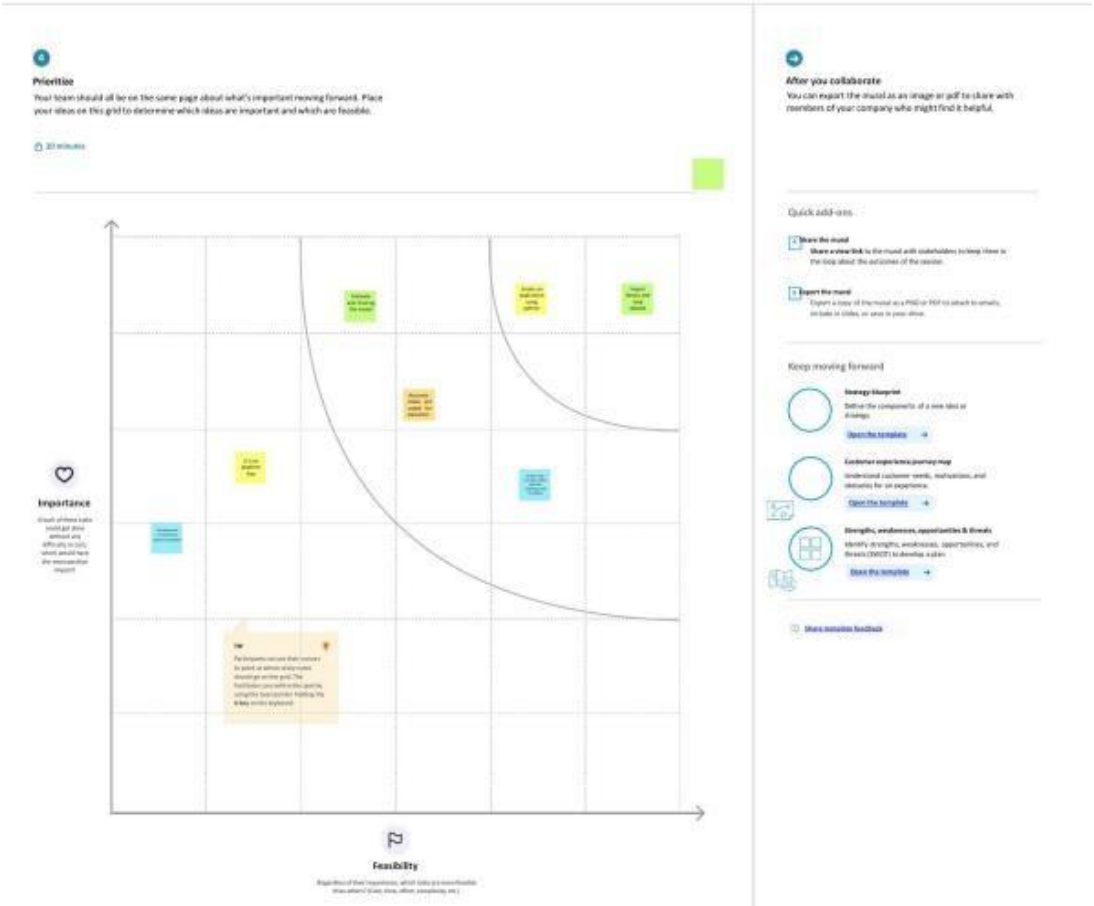


3.2 Ideation & Brainstorming:

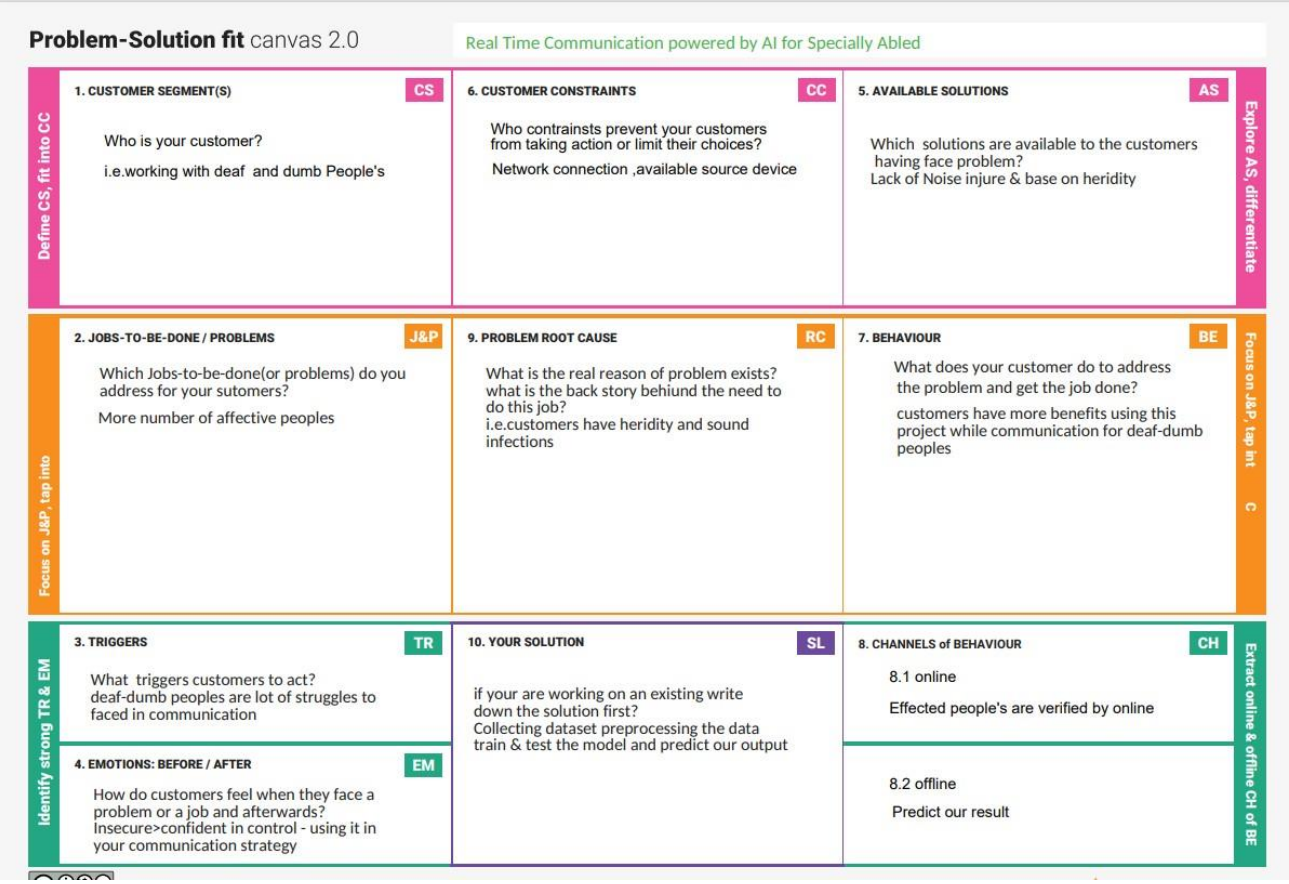





3
Group ideas
Have everyone move their ideas into the group sharing space within the template and have the team silently read through them.
Analysis
Recognition


4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
Importance
Feasibility
5
After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
Keep moving forward

3.3 Problem Solution Fit:



4.REQUIREMENT ANALYSIS:

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement	Sub Requirement (Sub-Task)
FR-1	**User Input**	The system gets the sign language input. The input may be image, video or live feed (depending upon the scope of the project)
FR-2	**Processing**	The system based on the trained model, should output the corresponding normal message.
FR-3	**System Output**	The system should output to the users, the normal message and voice.

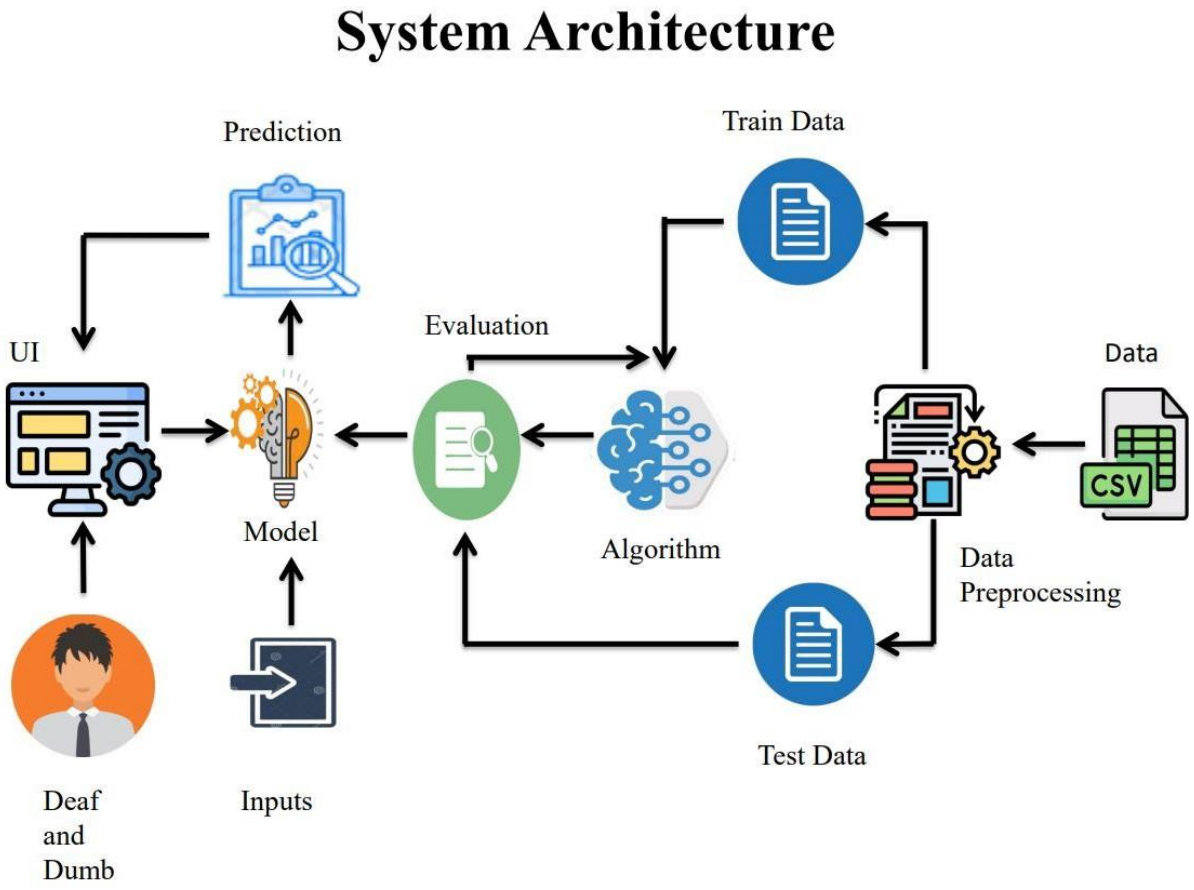
4.2 Non-Functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Deaf-mute people should be able to use the system with ease. The same applies for normal people who get the system's output. The system should have good UI.
NFR-2	**Security**	Even though the use-case of the system doesn't need any security feature, it must be ensured that the privacy of user data be maintained and handled appropriately.
NFR-3	**Reliability**	The translation of sign languages should be reliable. The accuracy of the system should be tested extensively to make sure that it is up to the mark.
NFR-4	**Performance**	The processing should be done in a considerable time so the conversation can go on without waiting for the system's output.
NFR-5	**Availability**	The system should be universally accessible. Since sign language is almost same everywhere, the system can be used across the globe.
NFR-6	**Scalability**	The system should be scalable to accommodate new features and functionalities and to cater wider range of people in future.

5 .PROJECT DESIGN:

5.1 System Architecture:



5.2 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information.

5.3 User Stories :

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Low vision)	Registration	USN-1	As a user, who has trouble reading due to low vision, I want to be able to make the text larger on the screen so that I can read it.	I can access my account / dashboard	High	Sprint-1
Customer (Color blindness)		USN-2	As a user, who is color blind ,I want to have access to information conveyed in color so that, I do not miss anything and I understand the content.	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Impaired user)		USN-3	As a user, who is hearing-mpaired, Iwant a transcript of the spoken audio so that I can have access to all information provided in audio clips	I can register & access the dashboard with Facebook Login	Low	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Project Backlog, Sprint Schedule and Estimation :

Sprint	Functional Requiremen (Epic)	User Story / Number	Userstory / Task	Story Point	Priority	Team Members
Sprint 1	Dataset	USN-1	It is fairly Possible to get the dataset we need on the internet but in this project, we will be creating on our own.	2	High	Barathkumar SR
Sprint 2	CNN on the captured	USN-2	The data using ImageDataGenerator of keras through which we can use the flow_from_directory function to load the train and test data.	2	Medium	Ajay S
Sprint 3	Predicting the Data	USN-3	The load the previously saved model using keras.models.load-model and feed the threshold image of the ROI consisting of the hand as an input to prediction.	2	High	Ananth Krishna A
Sprint 4	Machine Learning	USN-4	This is an interesting machine learning python project to gain expertise.This can be further extended for detecting the English alphabets.	2	High	Arishkarthik P

6.2 Sprint delivery Plan :

Sprint	Total Story Point	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Point Completed	Sprint Release Date (Actual)
Sprint - 1	20	4 Days	24 Oct 2022	27 Oct 2022	20	27 Oct 2022
Sprint - 2	20	6 Days	29 Oct 2022	03 Nov 2022	20	03 Nov 2022
Sprint – 3	20	6 Days	04 Nov 2022	09 Nov 2022	20	09 Nov 2022
Sprint - 4	20	6 Days	10 Nov 2022	16 Nov 2022	20	16 Nov 2022

Velocity:

Imagine we have a 10 day sprint duration and the velocity of the team is 20.

$$AV = \frac{SPRINT\ DURATION}{VELOCITY} = \frac{20}{10} = 2$$

Burndown Chart:

A Burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

BURNDOWN CHART			
Sprint	**Date**	**Estimated Effort**	**Actual Effort**
Sprint - 1	24 – Oct - 2022	20	20
	25 – Oct - 2022	19	20
	26 – Oct - 2022	18	19
	27 – Oct - 2022	17	19
	28 – Oct - 2022	17	18
Sprint - 2	29 – Oct - 2022	16	17
	30 – Oct - 2022	15	15
	31 – Oct - 2022	14	13
	01 – Nov - 2022	13	12
	02 – Nov - 2022	12	11
	03 – Nov - 2022	11	11
Sprint - 3	04 – Nov - 2022	11	11
	05 – Nov - 2022	10	9
	06 – Nov - 2022	9	8
	07 – Nov - 2022	8	7
	08 – Nov - 2022	7	6
	09 – Nov - 2022	6	6
Sprint - 4	10 – Nov - 2022	5	5
	11 – Nov - 2022	5	5
	12 – Nov - 2022	5	4
	13 – Nov - 2022	4	3
	14 – Nov - 2022	3	2
	15 – Nov - 2022	2	2
	16 – Nov - 2022	1	2

7. CODING :

7.1 Data collection:

In Feature – 1 of our project, we collect the dataset, pre-process it, create our CNN model, train the model, and save the trained model. We have used a dataset from Kaggle.com to classify the ISL (Indian Sign Language) alphabets (A-Z).

7.2 Image Processing:

```
##Apply ImageDataGenerator Functionality To Train And Test set.
from google.colab import drive
!unzip '/content/training_set.zip'
from tensorflow.keras.preprocessing.image import ImageDataGenerator
print("This dataset has been created and uploaded by IBM-TeamID-IBM-Project-33773-
1660226649")
train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,
vertical_flip=False)
test_datagen= ImageDataGenerator(rescale=1./255)
x_train =
train_datagen.flow_from_directory('/content/training_set',target_size=(64,64),
batch_size=300,
class_mode='categorical', color_mode = "grayscale")
x_test = test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),
batch_size=300,
class_mode='categorical', color_mode = "grayscale")
x_train.class_indices
x_test.class_indices
##Import ImageDataGenerator Library And Configure It

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Training Datagen
train_datagen =
ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=F
alse)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlib
```

7.3 Model Building :

```
##Model Building
drive.mount('/content/drive')
!unzip '/content/drive/MyDrive/IBM/training_set.zip'
!unzip '/content/drive/MyDrive/IBM/test_set.zip'
##Train And Test The Data
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale = 1./255, shear_range=0.2,
zoom_range=0.2,horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
train_data= train_datagen.flow_from_directory("/content/training_set",
target_size=(64,64),batch_size=300,class_mode='categorical', color_mode
="grayscale")
test_data = test_datagen.flow_from_directory("/content/test_set",
target_size=(64,64),batch_size=300,class_mode='categorical', color_mode
="grayscale")
##Import The Required Model Building Libraries
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
## Initialize The Model
model=Sequential()
## The Convolution Layer
model.add(Convolution2D(32,(3,3), input_shape=(64,64,1), activation = 'relu'))
## The POOLING Layer
model.add(MaxPooling2D(pool_size=(2,2)))
## The FLATTEN Layer
model.add(Flatten())
## The DENSE Layer
model.add(Dense( units=512, activation='relu'))
model.add(Dense(units=9, activation='softmax'))
## Compile The Model
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

7.4 main.py

```
import cv2
video = cv2.VideoCapture(0)
while True:
    ret, frame = video.read()
    cv2.imshow("Frame", frame)
    k = cv2.waitKey(1)
    if k == ord('q'):
        break
```

```
video.release()
cv2.destroyAllWindows()
```

7.5 flask application

```
from flask import Flask, Response, render_template
from camera import Video
app = Flask(__name_)
@app.route('/')
def index():
    return render_template('index.html')
def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')
@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary =
frame')
if_name_== '_main_':
    app.run(host='0.0.0.0')
```

7.5 camera.py

```
import cv2
import numpy as np
from keras.models import load_model
from keras.utils import load_img, img_to_array
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('My_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 = load_img('image.jpg', target_size=(64,64))
        x = 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()
```

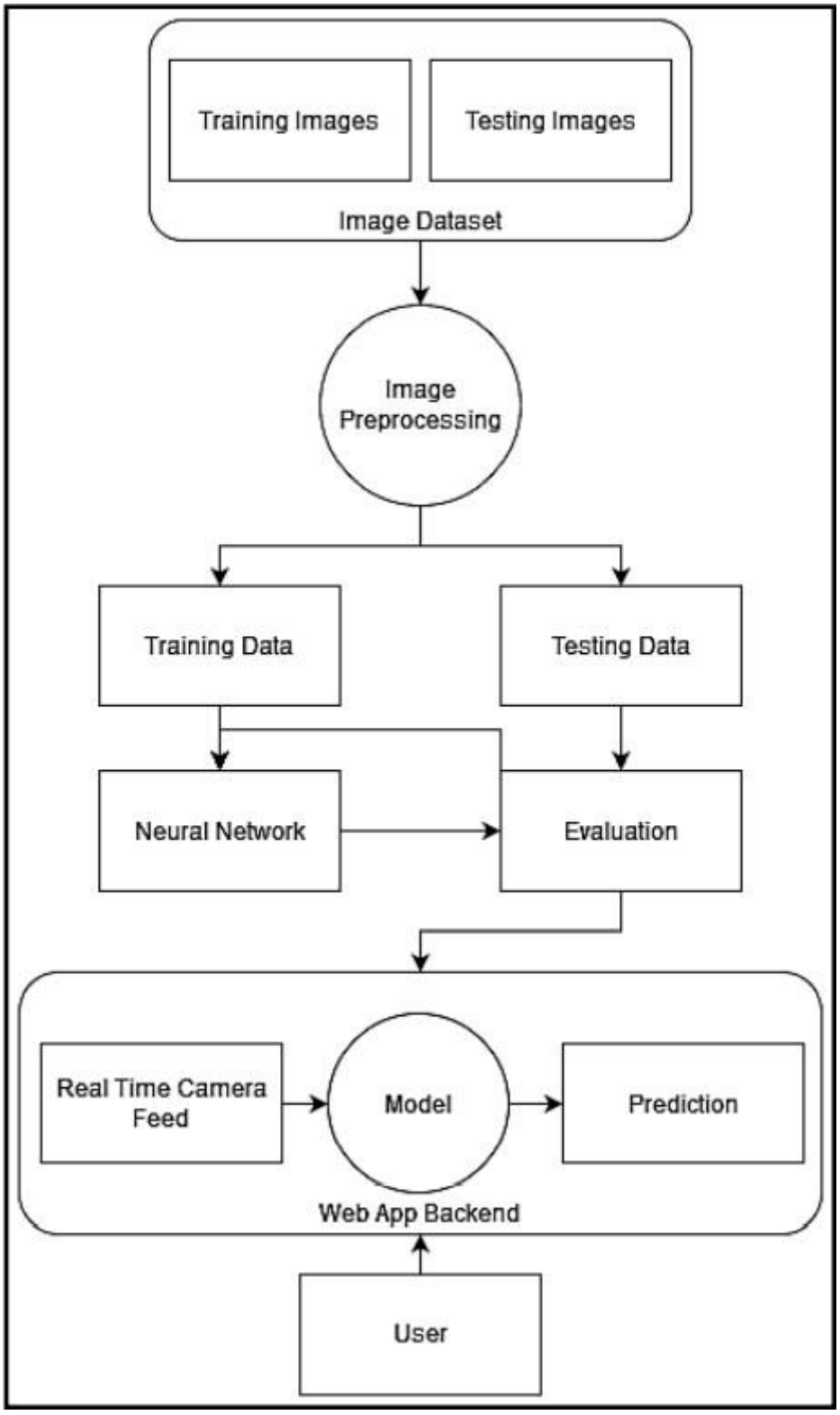
7.6 index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-
fit=no">
    <title>Sign Language Detection</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.12.0/css/all.css">
    <link rel="stylesheet" href="assets/css/Banner-Heading-Image.css">
    <link rel="stylesheet" href="assets/css/Navbar-Centered-Brand.css">
    <link rel="stylesheet" href="assets/css/styles.css">
</head>

<body style=" background-image: url('images.jpg');">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background-
color:rgb(206, 37, 37);">
        <div class="container">
            <div></div><a class="navbar-brand d-flex align-items-center"
href="#"><span
                    class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex
justify-content-center align-items-center me-2 bs-icon"><i
                        class="fas fa-flask"></i></span><span style="color:
rgb(255,255,255);">
                    <h3>Real-Time Communication
                        System Powered By AI For Specially Abled</h3>
                </span></a>
            <div></div>
        </div>
    </nav>
```

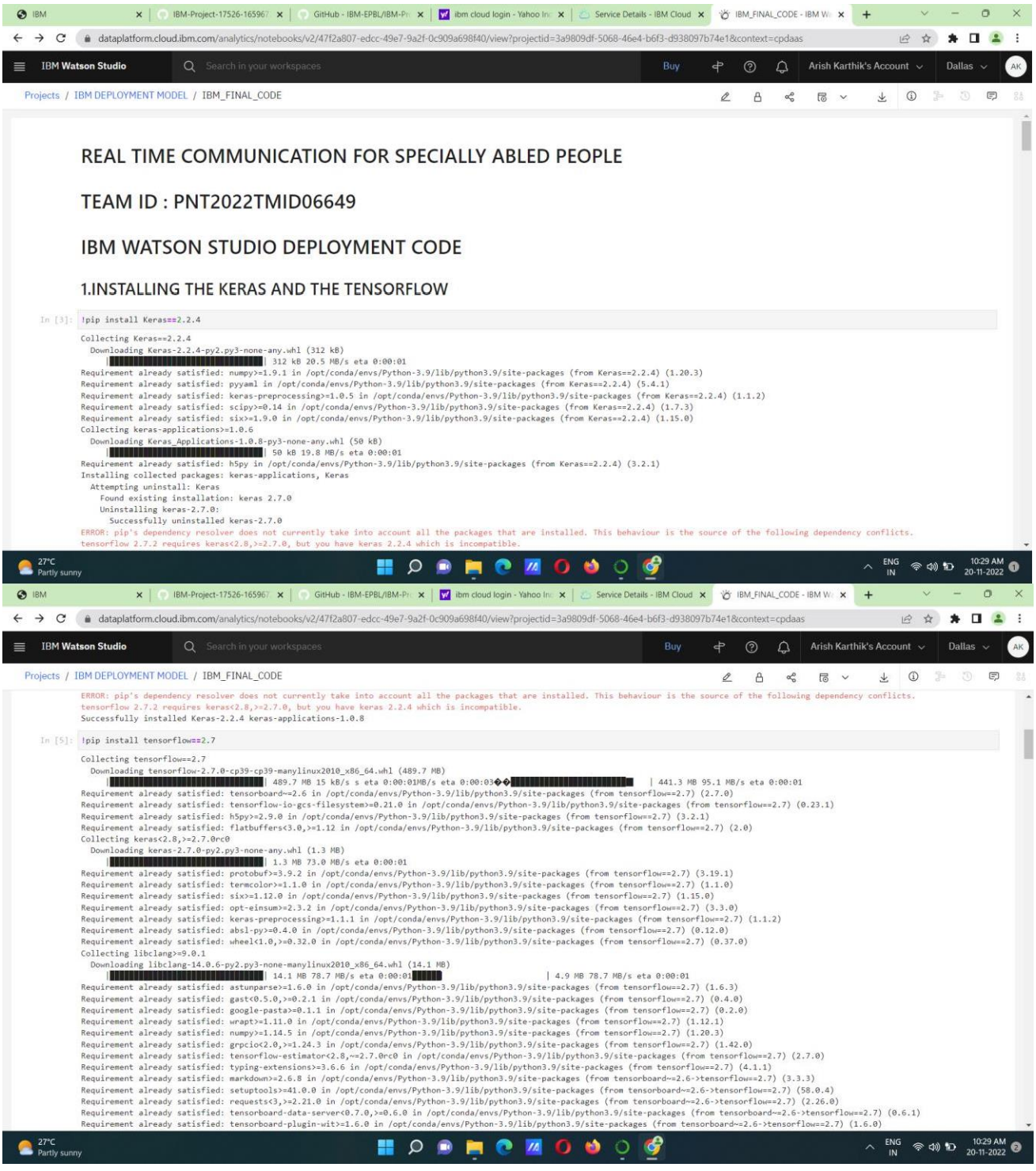
```
    <section>
        <div class="d-flex flex-column justify-content-center align-items-center">
            <div class="d-flex flex-column justify-content-center align-items-
center" id="div-video-feed"
                style="width: 640px;height: 480px;margin: 10px;min-height:
480px;min-width: 640px;border-radius: 10px;border: 4px dashed rgb(0, 0, 0) ;
background-color: antiquewhite;">
                <img src="{{ url_for('video_feed') }}"
                    style="width: 100%;height: 100%;color: rgb(0, 0, 0);text-align:
center;font-size: 20px;"
                    alt="Camera Access Not Provided!">
            </div>
        </div>
        <div class="d-flex flex-column justify-content-center align-items-center"
style="margin-bottom: 10px;"><button
                class="btn btn-info" type="button" data-bs-target="#modal-1" data-
bs-toggle="modal">Quick Reference
                -<strong> ASL Alphabets</strong></button></div>
    </section>
    <section>
        <div class="container">
            <div class="accordion text-white" role="tablist" id="accordion-1">
                <div class="accordion-item" style="background: rgb(112, 66, 104);">
                    <h2 class="accordion-header" role="tab"><button
class="accordion-button" data-bs-toggle="collapse"
                            data-bs-target="#accordion-1 .item-1" aria-
expanded="true"
                            aria-controls="accordion-1 .item-1"
                            style="background: rgb(170, 194, 106);color:
rgb(255,255,255);">About The Project</button>
                    </h2>
                    <div class="accordion-collapse collapse show item-1"
role="tabpanel" data-bs-parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">Artificial Intelligence has made it
possible to handle our daily activities
                                in new and simpler ways. With the ability to
automate tasks that normally require human
                                intelligence, such as speech and voice recognition,
visual perception, predictive text
                                functionality, decision-making, and a variety of
other tasks, AI can assist people with
                                disabilities by significantly improving their
ability to get around and participate in
                                daily activities.<br><br>Currently, Sign Recognition
is available <strong>only for
                                    alphabets A-I</strong> and not for J-Z, since J-
Z alphabets also require Gesture
```

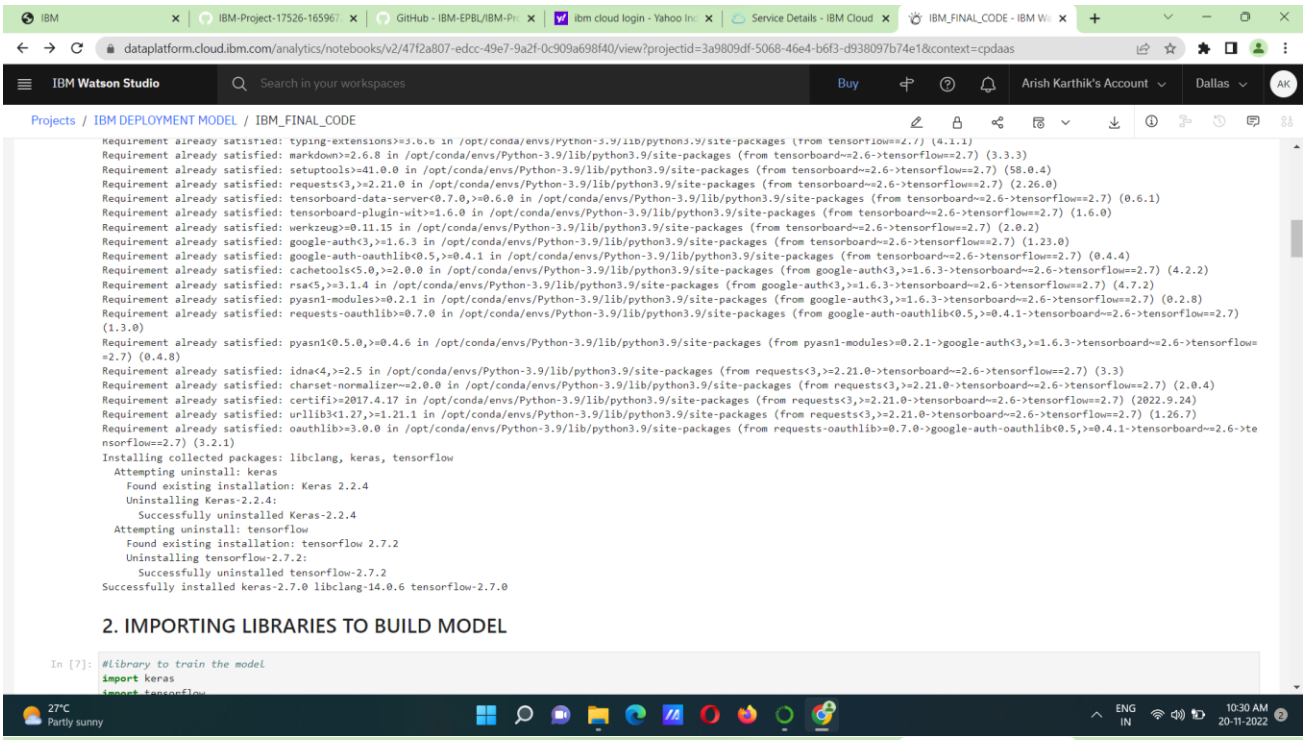
```
                                Recognition for them to be able to be predicted
correctly to a certain degree of
                                accuracy.</p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
    <div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h4 class="modal-title">American Sign Language -
Alphabets</h4><button type="button"
                        class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
                </div>
                <div class="modal-body"><img src="{{ url_for('static',
filename='img/ASL_Alphabets.png') }}"
                        width="100%"></div>
                <div class="modal-footer"><button class="btn btn-secondary"
type="button"
                        data-bs-dismiss="modal">Close</button></div>
            </div>
        </div>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"><
/script>
</body>
</html>
```

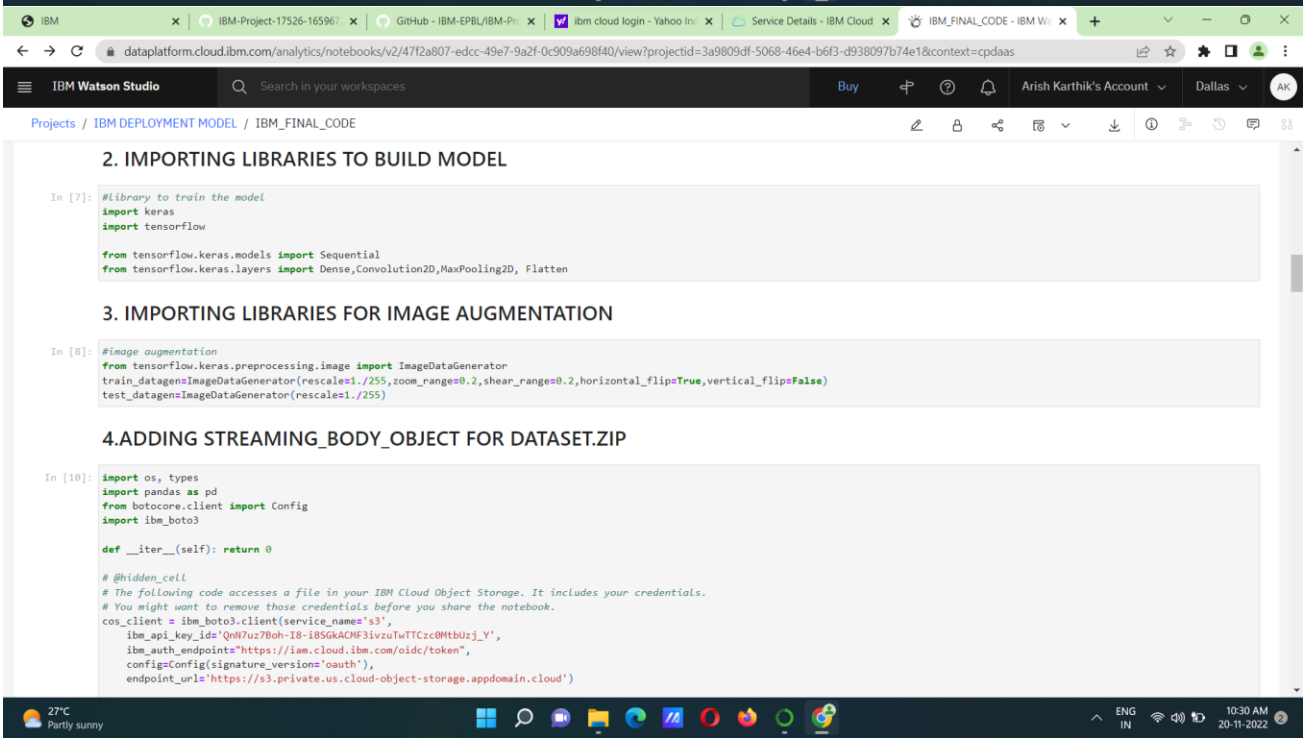
8. FLOWCHART :

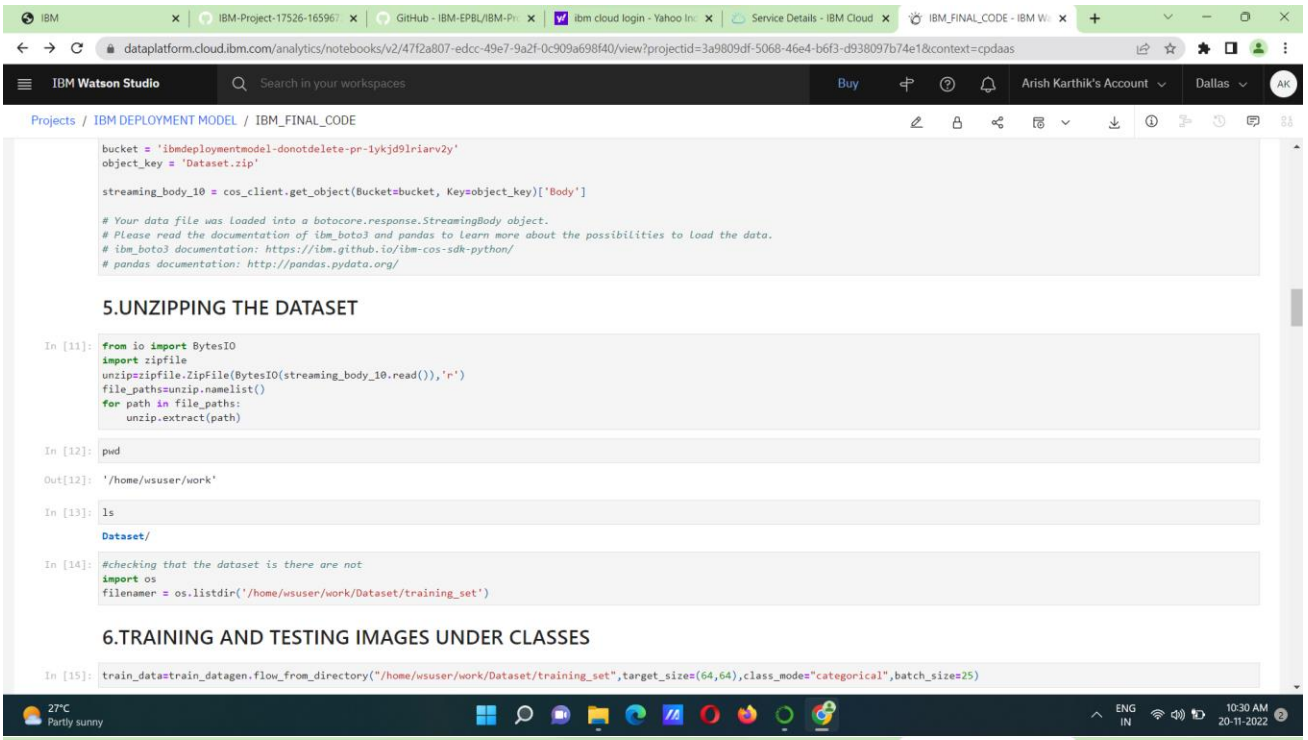

Training Images
Testing Images
Image Dataset
Image Preprocessing
Training Data
Testing Data
Neural Network
Evaluation
Real Time Camera Feed
Model
Prediction
Web App Backend
User

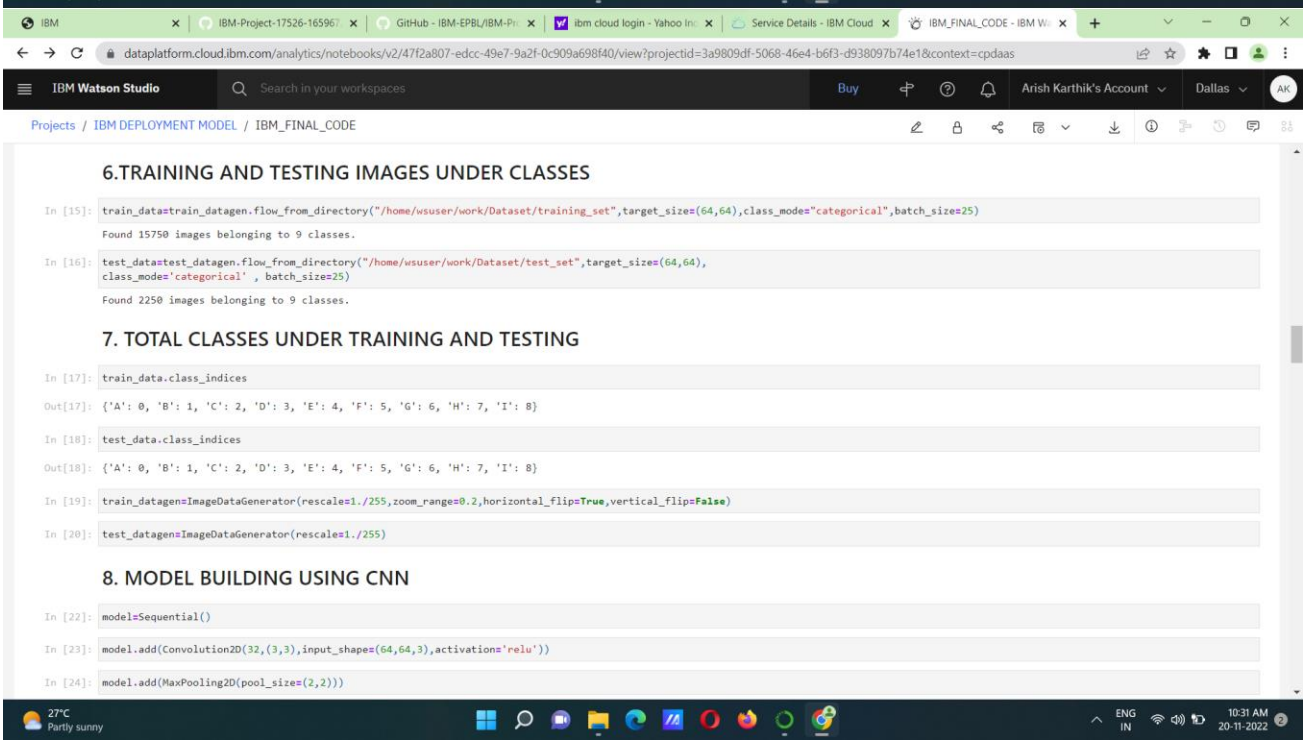
9. IBM DEPLOYMENT :

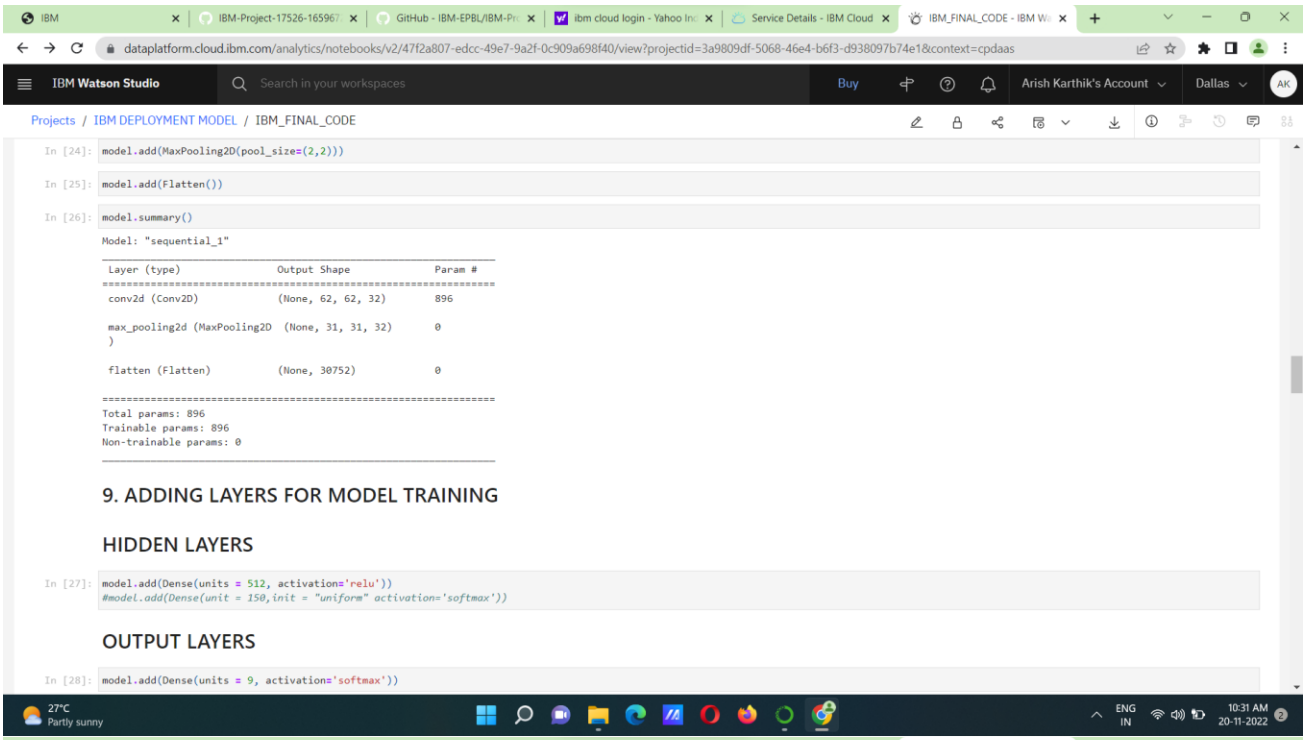



IBM Watson Studio
Search in your workspaces
Buy
Arish Karthik's Account
Dallas
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.7) (4.1.1)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (3.3.3)
Requirement already satisfied: setuptools>=41.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (58.0.4)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (2.26.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (1.6.0)
Requirement already satisfied: werkzeug>=0.11.15 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (2.0.2)
Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (1.23.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.6->tensorflow==2.7) (0.4.4)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow==2.7) (4.2.2)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow==2.7) (4.7.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow==2.7) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow==2.7) (1.3.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow==2.7) (0.4.8)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow==2.7) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow==2.7) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow==2.7) (2022.9.24)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow==2.7) (1.26.7)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow==2.7) (3.2.1)
Installing collected packages: libclang, keras, tensorflow
Attempting uninstall: keras
Found existing installation: Keras 2.2.4
Uninstalling Keras-2.2.4:
Successfully uninstalled Keras-2.2.4
Attempting uninstall: tensorflow
Found existing installation: tensorflow 2.7.2
Uninstalling tensorflow-2.7.2:
Successfully uninstalled tensorflow-2.7.2
Successfully installed keras-2.7.0 libclang-14.0.6 tensorflow-2.7.0
2. IMPORTING LIBRARIES TO BUILD MODEL
In [7]:
#library to train the model
import keras
import tensorflow
27°C
Partly sunny
ENG
IN
10:30 AM
20-11-2022


IBM Watson Studio
Search in your workspaces
Buy
Arish Karthik's Account
Dallas
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
2. IMPORTING LIBRARIES TO BUILD MODEL
In [7]:
#library to train the model
import keras
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D, Flatten
3. IMPORTING LIBRARIES FOR IMAGE AUGMENTATION
In [8]:
#image augmentation
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2,horizontal_flip=True,vertical_flip=False)
test_datagen=ImageDataGenerator(rescale=1./255)
4.ADDING STREAMING_BODY_OBJECT FOR DATASET.ZIP
In [10]:
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='QnN7uz7Boh-I8-i8SGkACMF3ivzuTwTTCzc0MtbUzj_Y',
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')
27°C
Partly sunny
ENG
IN
10:30 AM
20-11-2022


IBM Watson Studio
Search in your workspaces
Buy
Arish Karthik's Account
Dallas
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
bucket = 'ibmdeploymentmodel-donotdelete-pr-1ykjd9lriarv2y'
object_key = 'Dataset.zip'
streaming_body_10 = 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/
5.UNZIPPING THE DATASET
In [11]:
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_10.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
unzip.extract(path)
In [12]:
pwd
Out[12]:
'/home/wsuser/work'
In [13]:
ls
Dataset/
In [14]:
#checking that the dataset is there are not
import os
filenamer = os.listdir('/home/wsuser/work/Dataset/training_set')
6.TRAINING AND TESTING IMAGES UNDER CLASSES
In [15]:
train_data=train_datagen.flow_from_directory("/home/wsuser/work/Dataset/training_set",target_size=(64,64),class_mode="categorical",batch_size=25)
27°C
Partly sunny
ENG
IN
10:30 AM
20-11-2022


IBM Watson Studio
Search in your workspaces
Buy
Arish Karthik's Account
Dallas
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
6.TRAINING AND TESTING IMAGES UNDER CLASSES
In [15]:
train_data=train_datagen.flow_from_directory("/home/wsuser/work/Dataset/training_set",target_size=(64,64),class_mode="categorical",batch_size=25)
Found 15750 images belonging to 9 classes.
In [16]:
test_data=test_datagen.flow_from_directory("/home/wsuser/work/Dataset/test_set",target_size=(64,64),
class_mode='categorical' , batch_size=25)
Found 2250 images belonging to 9 classes.
7. TOTAL CLASSES UNDER TRAINING AND TESTING
In [17]:
train_data.class_indices
Out[17]:
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
In [18]:
test_data.class_indices
Out[18]:
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
In [19]:
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
In [20]:
test_datagen=ImageDataGenerator(rescale=1./255)
8. MODEL BUILDING USING CNN
In [22]:
model=Sequential()
In [23]:
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
In [24]:
model.add(MaxPooling2D(pool_size=(2,2)))
27°C
Partly sunny
ENG
IN
10:31 AM
20-11-2022



```
In [24]: model.add(MaxPooling2D(pool_size=(2,2)))

In [25]: model.add(Flatten())

In [26]: model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 flatten (Flatten)           (None, 30752)             0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

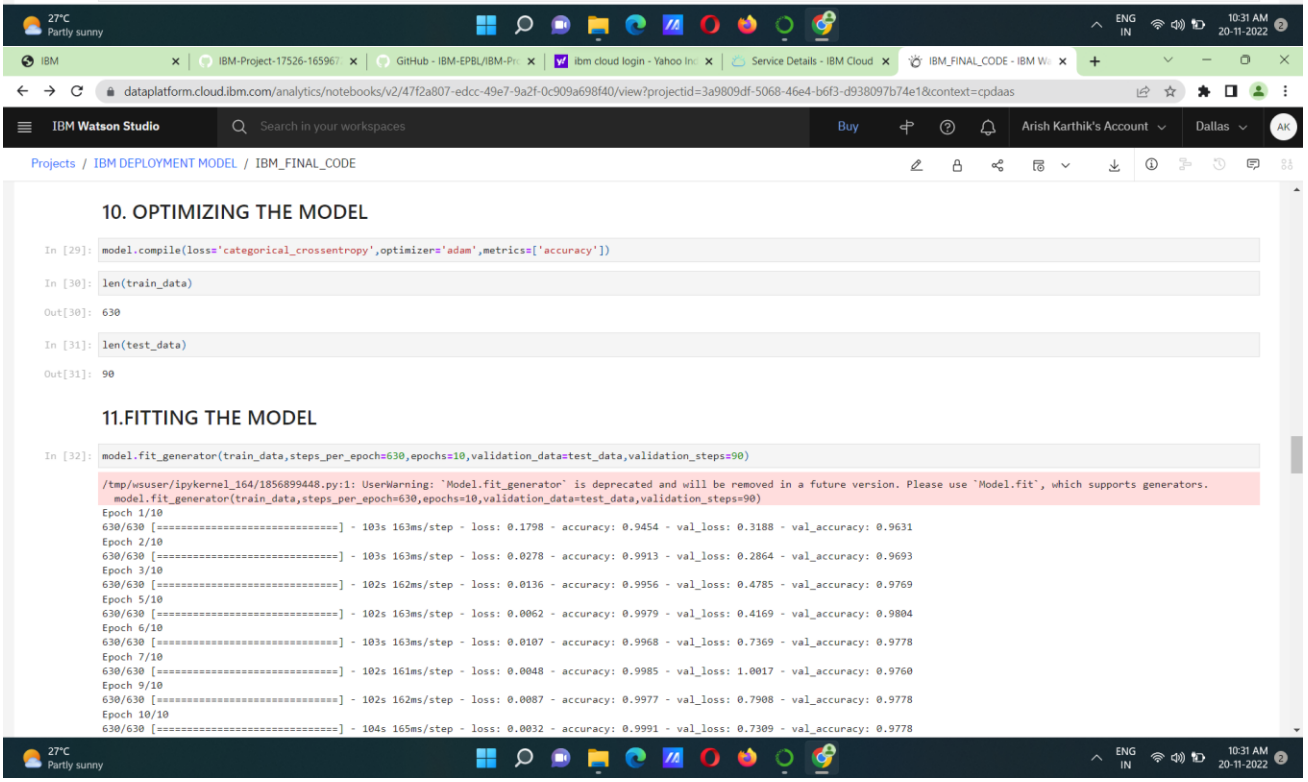
9. ADDING LAYERS FOR MODEL TRAINING

HIDDEN LAYERS

```
In [27]: model.add(Dense(units = 512, activation='relu'))
         #model.add(Dense(unit = 150,init = "uniform" activation='softmax'))
```

OUTPUT LAYERS

```
In [28]: model.add(Dense(units = 9, activation='softmax'))
```



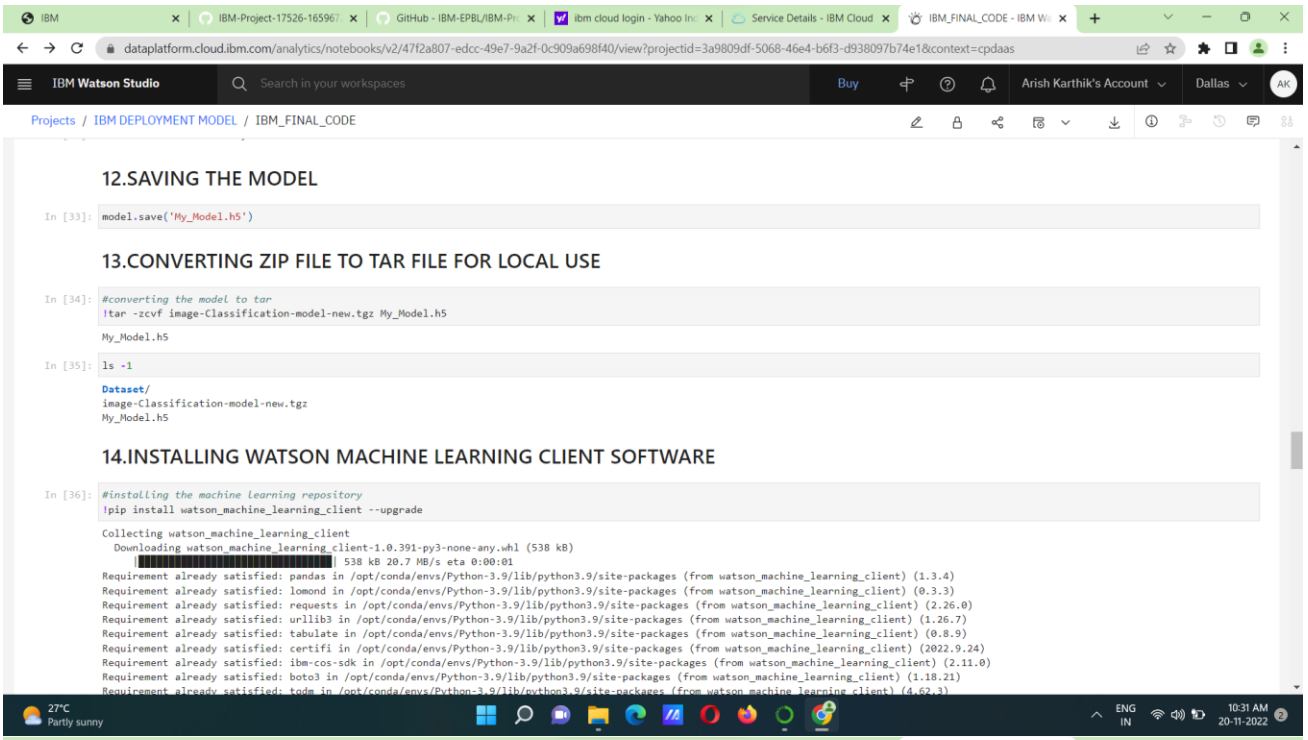
10. OPTIMIZING THE MODEL

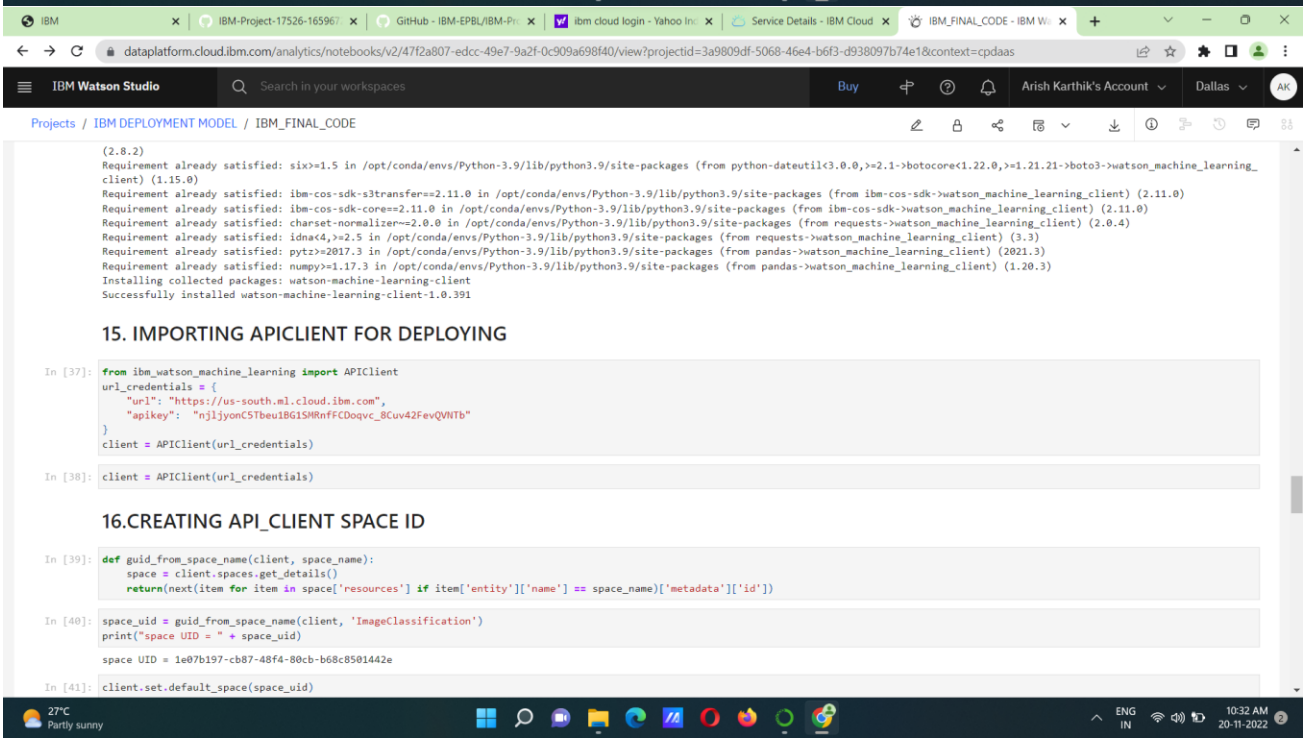
```
In [29]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

In [30]: len(train_data)

Out[30]: 630

In [31]: len(test_data)

Out[31]: 90
```

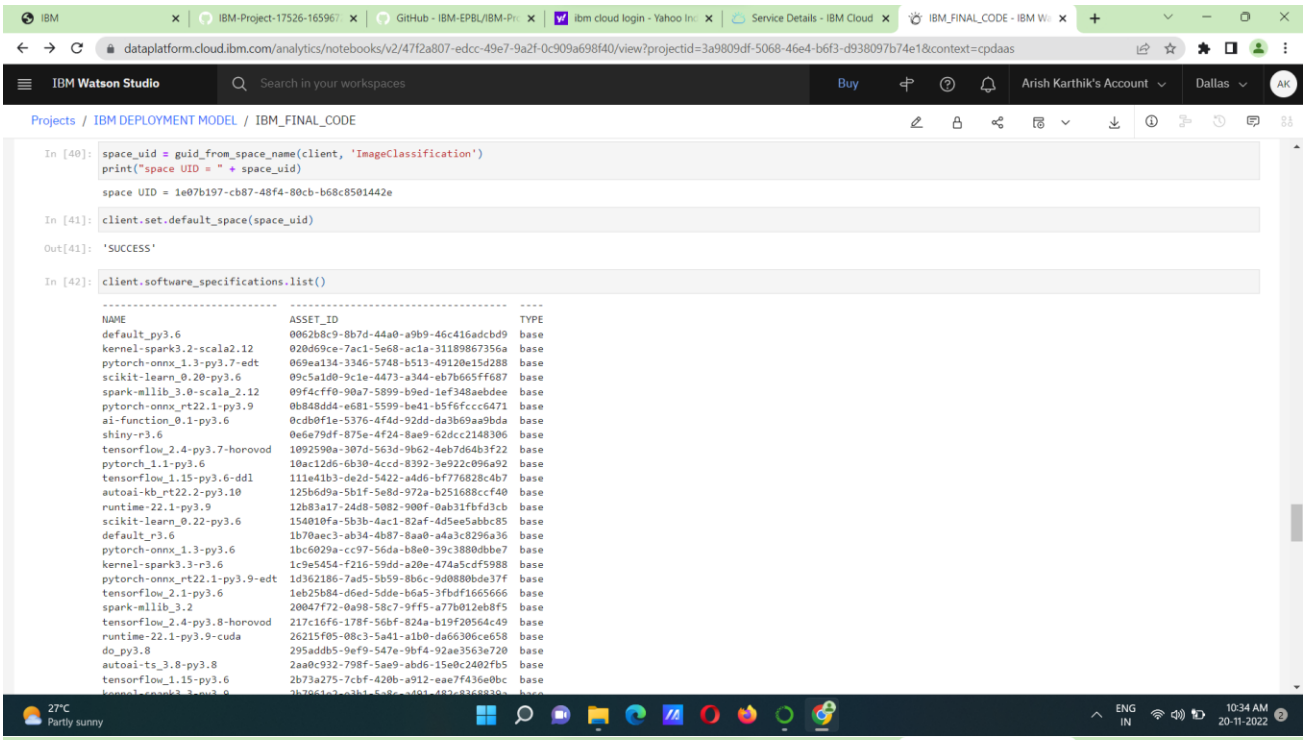
11.FITTING THE MODEL

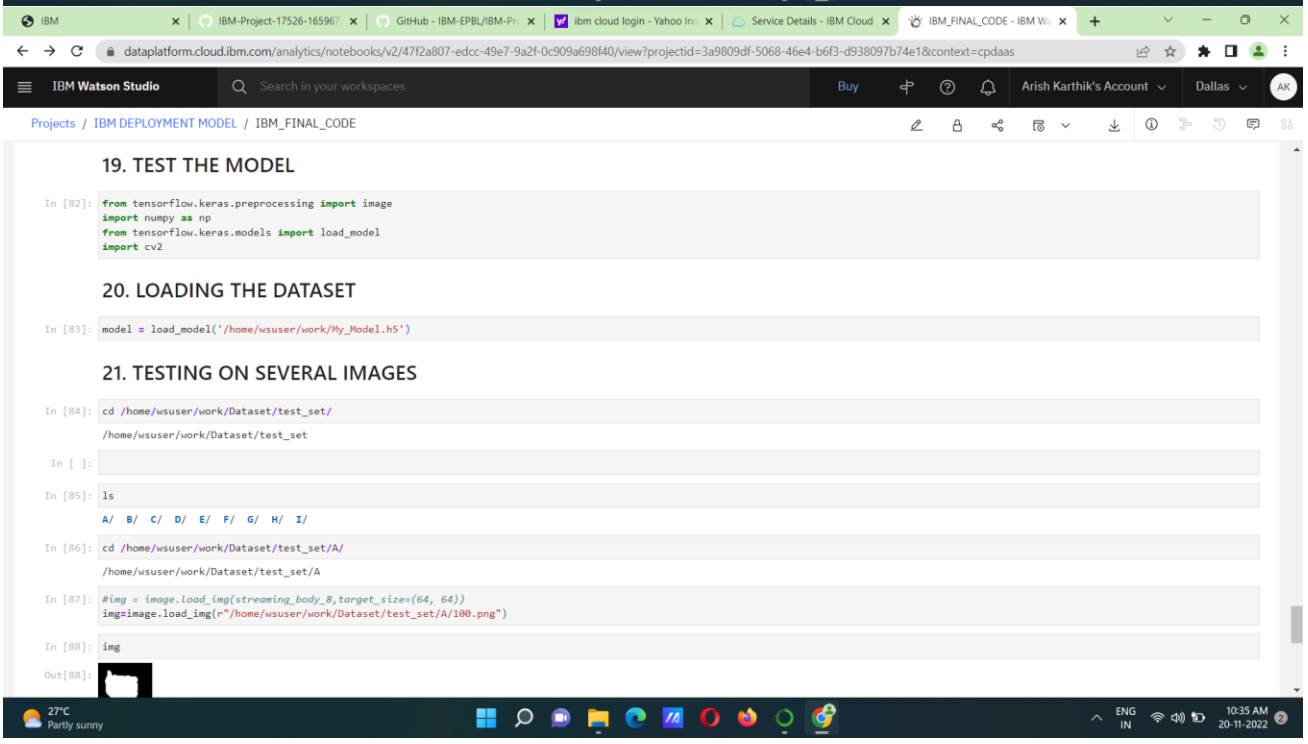
```
In [32]: model.fit_generator(train_data,steps_per_epoch=630,epochs=10,validation_data=test_data,validation_steps=90)
```

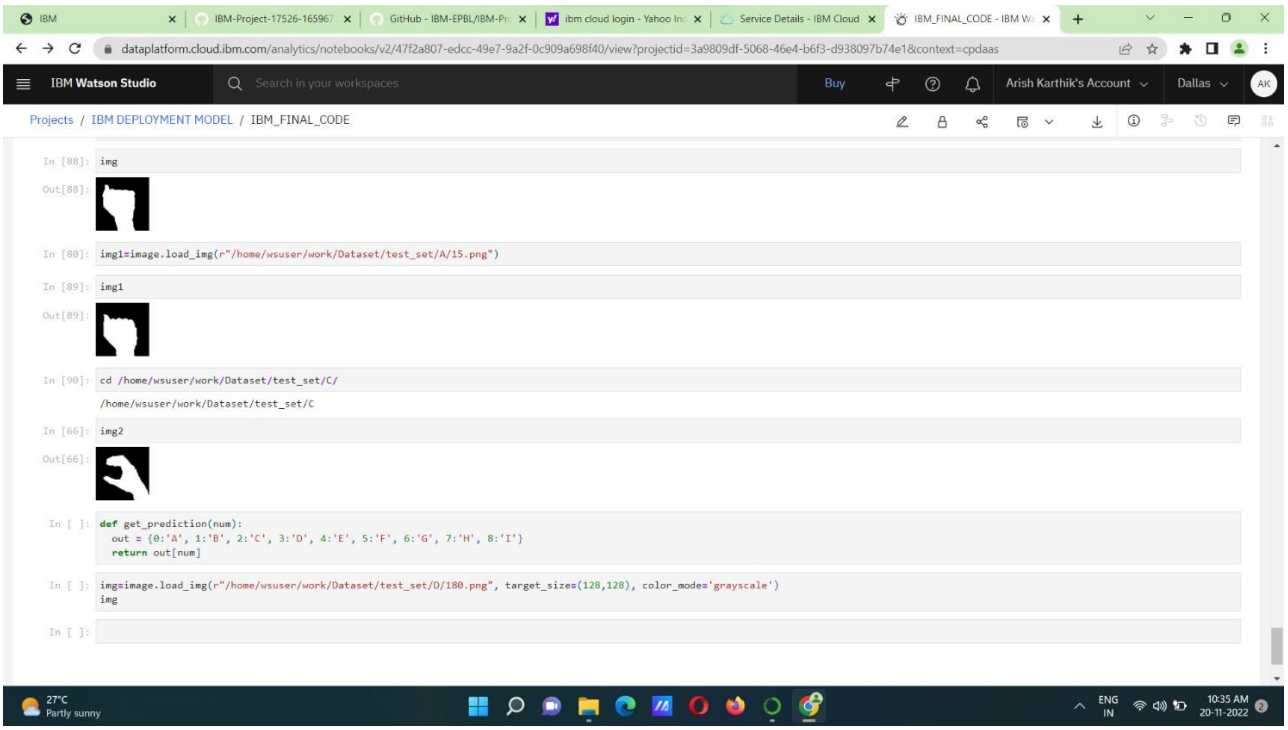
```
/tmp/wsuser/ipykernel_164/1856899448.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(train_data,steps_per_epoch=630,epochs=10,validation_data=test_data,validation_steps=90)
Epoch 1/10
630/630 [==============================] - 103s 163ms/step - loss: 0.1798 - accuracy: 0.9454 - val_loss: 0.3188 - val_accuracy: 0.9631
Epoch 2/10
630/630 [==============================] - 103s 163ms/step - loss: 0.0278 - accuracy: 0.9913 - val_loss: 0.2864 - val_accuracy: 0.9693
Epoch 3/10
630/630 [==============================] - 102s 162ms/step - loss: 0.0136 - accuracy: 0.9956 - val_loss: 0.4785 - val_accuracy: 0.9769
Epoch 5/10
630/630 [==============================] - 102s 163ms/step - loss: 0.0062 - accuracy: 0.9979 - val_loss: 0.4169 - val_accuracy: 0.9804
Epoch 6/10
630/630 [==============================] - 103s 163ms/step - loss: 0.0107 - accuracy: 0.9968 - val_loss: 0.7369 - val_accuracy: 0.9778
Epoch 7/10
630/630 [==============================] - 102s 161ms/step - loss: 0.0048 - accuracy: 0.9985 - val_loss: 1.0017 - val_accuracy: 0.9760
Epoch 9/10
630/630 [==============================] - 102s 162ms/step - loss: 0.0087 - accuracy: 0.9977 - val_loss: 0.7908 - val_accuracy: 0.9778
Epoch 10/10
630/630 [==============================] - 104s 165ms/step - loss: 0.0032 - accuracy: 0.9991 - val_loss: 0.7309 - val_accuracy: 0.9778
```


IBM Watson Studio
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
12.SAVING THE MODEL
In [33]: model.save('My_Model.h5')
13.CONVERTING ZIP FILE TO TAR FILE FOR LOCAL USE
In [34]: #converting the model to tar
!tar -zcvf image-Classification-model-new.tgz My_Model.h5
My_Model.h5
In [35]: ls -1
Dataset/
image-Classification-model-new.tgz
My_Model.h5
14.INSTALLING WATSON MACHINE LEARNING CLIENT SOFTWARE
In [36]: #installing the machine learning repository
!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 20.7 MB/s eta 0:00:01
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: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson_machine_learning_client) (0.3.3)
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: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson_machine_learning_client) (0.8.9)
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: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson_machine_learning_client) (1.18.21)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson_machine_learning_client) (4.62.3)


IBM Watson Studio
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
(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_machine_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: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson_machine_learning_client) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson_machine_learning_client) (3.3)
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
15. IMPORTING APICLIENT FOR DEPLOYING
In [37]: from ibm_watson_machine_learning import APIClient
url_credentials = {
"url": "https://us-south.ml.cloud.ibm.com",
"apikey": "njljyonC5TbeuIBG1SMRnfFCDoqvc_8Cuv42FevQVNTb"
}
client = APIClient(url_credentials)
In [38]: client = APIClient(url_credentials)
16.CREATING API_CLIENT SPACE ID
In [39]: 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 [40]: space_uid = guid_from_space_name(client, 'ImageClassification')
print("space UID = " + space_uid)
space UID = 1e07b197-cb87-48f4-80cb-b68c8501442e
In [41]: client.set.default_space(space_uid)


IBM Watson Studio
Search in your workspaces
Buy
Arish Karthik's Account
Dallas
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
In [40]: space_uid = guid_from_space_name(client, 'ImageClassification')
print("space UID = " + space_uid)
space UID = 1e07b197-cb87-48f4-80cb-b68c8501442e
In [41]: client.set.default_space(space_uid)
Out[41]: 'SUCCESS'
In [42]: 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
27°C
Partly sunny
ENG
IN
10:34 AM
20-11-2022


IBM Watson Studio
Search in your workspaces
Buy
Arish Karthik's Account
Dallas
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
19. TEST THE MODEL
In [82]: from tensorflow.keras.preprocessing import image
import numpy as np
from tensorflow.keras.models import load_model
import cv2
20. LOADING THE DATASET
In [83]: model = load_model('/home/wsuser/work/My_Model.h5')
21. TESTING ON SEVERAL IMAGES
In [84]: cd /home/wsuser/work/Dataset/test_set/
/home/wsuser/work/Dataset/test_set
In [ ]:
In [85]: ls
A/ B/ C/ D/ E/ F/ G/ H/ I/
In [86]: cd /home/wsuser/work/Dataset/test_set/A/
/home/wsuser/work/Dataset/test_set/A
In [87]: #img = image.load_img(streaming_body_8,target_size=(64, 64))
img=image.load_img(r"/home/wsuser/work/Dataset/test_set/A/100.png")
In [88]: img
Out[88]:
27°C
Partly sunny
ENG
IN
10:35 AM
20-11-2022


IBM Watson Studio
Search in your workspaces
Buy
Arish Karthik's Account
Dallas
Projects / IBM DEPLOYMENT MODEL / IBM_FINAL_CODE
In [88]: img
Out[88]:
In [80]: img1=image.load_img(r"/home/wsuser/work/Dataset/test_set/A/15.png")
In [89]: img1
Out[89]:
In [90]: cd /home/wsuser/work/Dataset/test_set/C/
/home/wsuser/work/Dataset/test_set/C
In [66]: img2
Out[66]:
In [ ]: def get_prediction(num):
out = {0:'A', 1:'B', 2:'C', 3:'D', 4:'E', 5:'F', 6:'G', 7:'H', 8:'I'}
return out[num]
In [ ]: img=image.load_img(r"/home/wsuser/work/Dataset/test_set/D/180.png", target_size=(128,128), color_mode='grayscale')
img
In [ ]:
27°C
Partly sunny
ENG
IN
10:35 AM
20-11-2022

10.RESULT

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:

11.ADVANTAGE & DISADVANTAGE

Advantages:

- It lets the specially-abled persons communicate with normal person using their sign language.
- If used properly, the software can bridge the gap between specially-abled persons and normal persons.
- The application is scalable, i.e., its scope can be expanded to recognize digits, words, etc.

Disadvantages:

- Specially-abled persons need to be trained to work with the software.
- Since the trained model's accuracy is not 100%, sometimes there may be cases where the model may produce erroneous results.
- The input image needs to be of good quality for the model to classify correctly.

12. CONCLUSION:

- The project we developed can bridge the gap of communication between deaf-mute people and the normal people.
- The project can be expanded with several functionalities in future.
- The project also has a business potential which can be tapped.

13. FUTURE SCOPE

This project has tremendous scope for future work. As this software includes components integrated into a Flask web application, components can be replaced or updated as per the changing requirements in the future.

Many new functionalities can be added like:

- Making the application predict words by combing multiple signs.
- Using NLP (Natural Language Processing) & ML (Machine Learning) to combine predicted words to create a meaningful sentence.
- Add many User Experience (UX) enhancing changes.
- Making the web application's UI (User Interface) look more appealing and user-friendly.

14.APPENDIX

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-33773-1660226649.git

Project Demo Video Link:

https://youtu.be/Id9Wfskp5MU