PROJECT REPORT

Real-Time Communication System Powered by AI for Specially Abled

Submitted by

TEAM ID: PNT2022TMID18087

Angelene Vidhya J - 9517201903010

Madhumitha K - 9517201903085

Anu Lavanya S - 9517201903012

Kiruthika V - 9517201903079

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

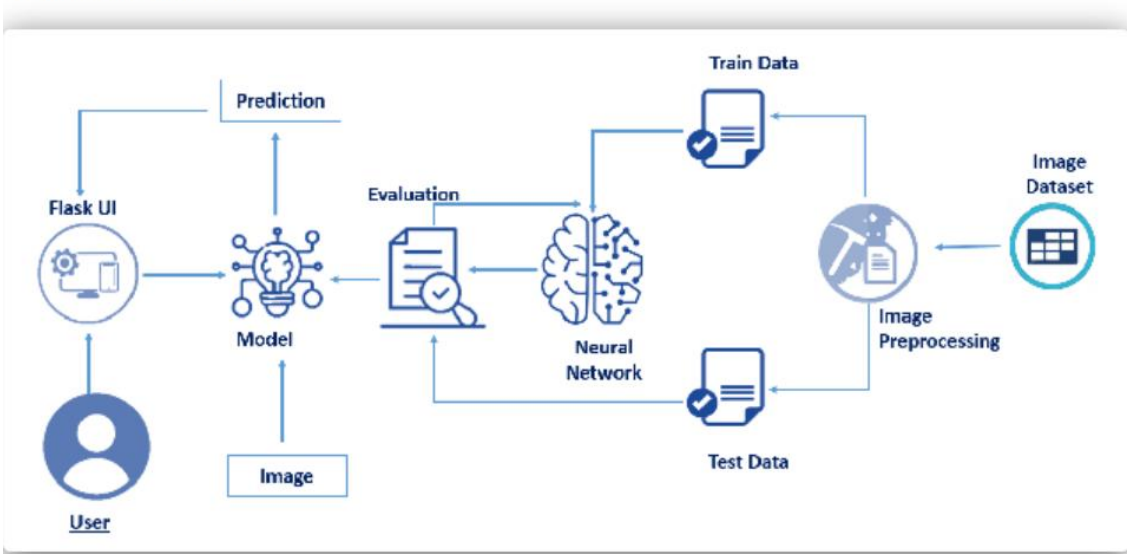
PROJECT OVERVIEW

Machine learning (ML) is **a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so**. Machine learning algorithms use historical data as input to predict new output values.

PURPOSE

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.

Technical Architecture:



CHAPTER 2

LITERATURE SURVEY

REFERENCES

Real-time Sign Language Recognition using Computer Vision

Image processing combined with Machine Learning is used. The image obtained after pre processing is used to forming the data that contains 24 alphabets and then converted the letters into text.

Real Time Two way Communication Approach For Hearing Impaired and Dumb Person Based On Image Processing

The objective of this paper is to develop a real time system for hand gesture recognition which recognize hand gestures, features of hands such as peak calculation and angle calculation and then convert gesture images into voice and vice versa. To implement this system they used a simple night vision web-cam with 20 megapixel intensity. The ideas consisted of designing and implement a system using artificial intelligence, image processing and data mining concepts to take input as hand gestures and generated recognizable outputs in the form of text and voice with 91% accuracy.

Real-Time conversion of sign language to text and speech

Performance of different sign language to text/speech is analysed. Using the best method from analysis, an android application is developed that can convert real time ASL signs to text/speech.

An Interpreter for the Differently able using Haptic Feedback and Machine Learning

The system implemented has two modules – an American Sign Language recognition module which uses a machine-learning algorithm to cater the people with vocal disability and a speech to haptic feedback conversion module which can be used for communicating to any kind of differently-able

Portable Communication Aid for Specially Challenged: Conversion of Hand Gestures into Voice and Vice Versa

This proposed system is to build up a real time embedded product for the disabled persons without handheld gloves. The speech communication by normal person will be converted into gestures for the disabled person for their better understanding

PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Deaf-Muted person	Communicating via sign to speech convertor	Not translated	Gestures not tracked properly	downcasted
PS-2	Normal person	Communicating with deaf-muted person via speech to signconverter	Not matched	Lack of resources	fettered

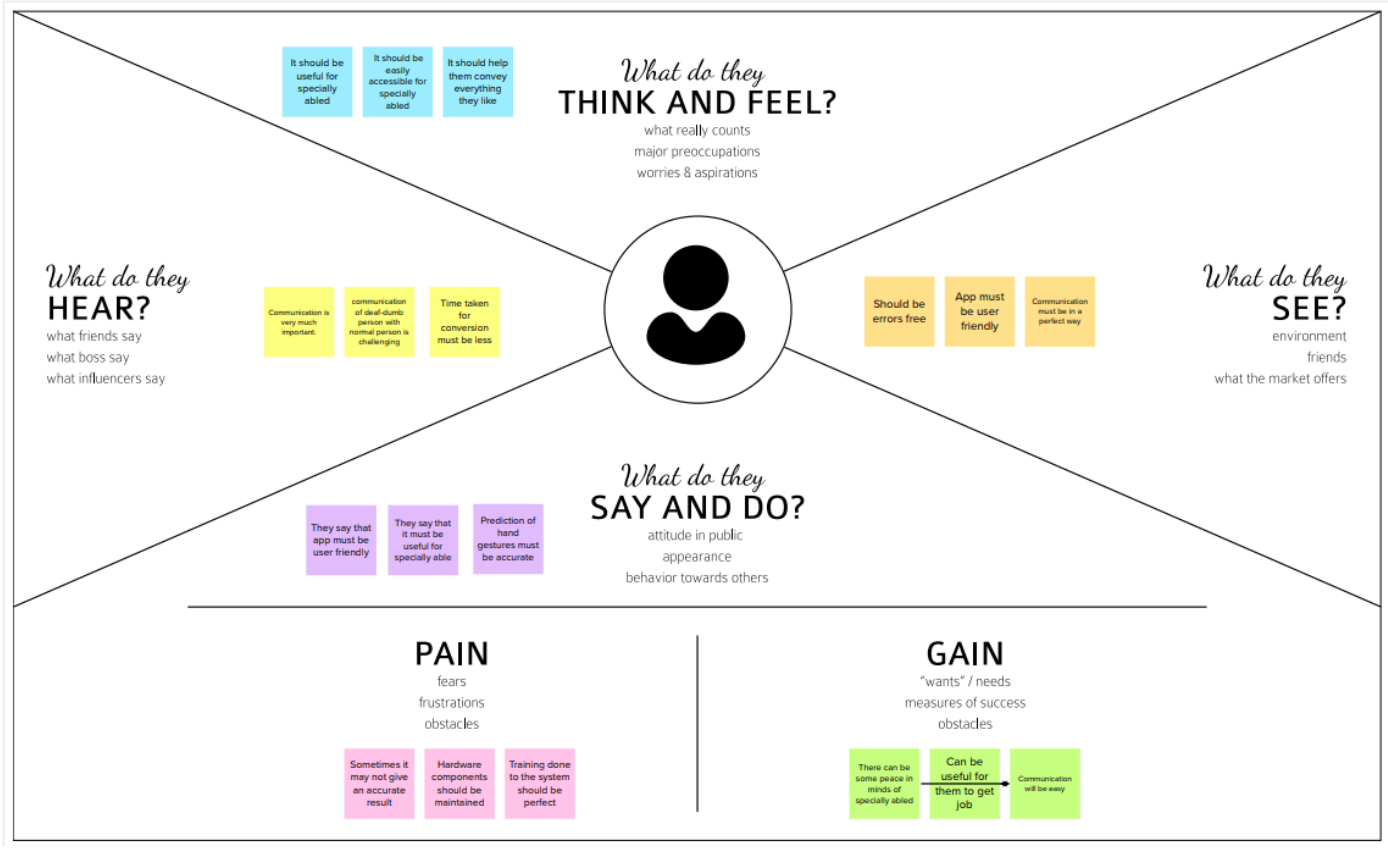
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS

1

Build empathy and keep your focus on the user by putting yourself in their shoes.



Share your feedback

IDEATION & BRAINSTORMING

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

It is difficult for normal person to understand the deaf-dumb person's sign language

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Angelene Vidhya J

- Image is preprocessed
- The pre-processed image forms the data that contains 26 alphabets
- This is done by machine learning algorithm
- At last the data is converted into speech

Madhumitha K

- Image data sets were collected and it was trained
- Input image test features were compared and matched with the trained features
- Statistical classification were done and it was recognized
- Then it been converted into text and then speech

Anu Lavanya S

- The Gesture image is given as the input .Image processing is done by using webcam and converted into grey images.
- Background segmentation is used to separate the object from its background. Noise reduction technique is used to remove the noise.
- Feature Extraction technique is used for the calculatin of "angle and peak calculation".
- Classification of bit generation is used .Then the output is of speech form.

Kiruthika V

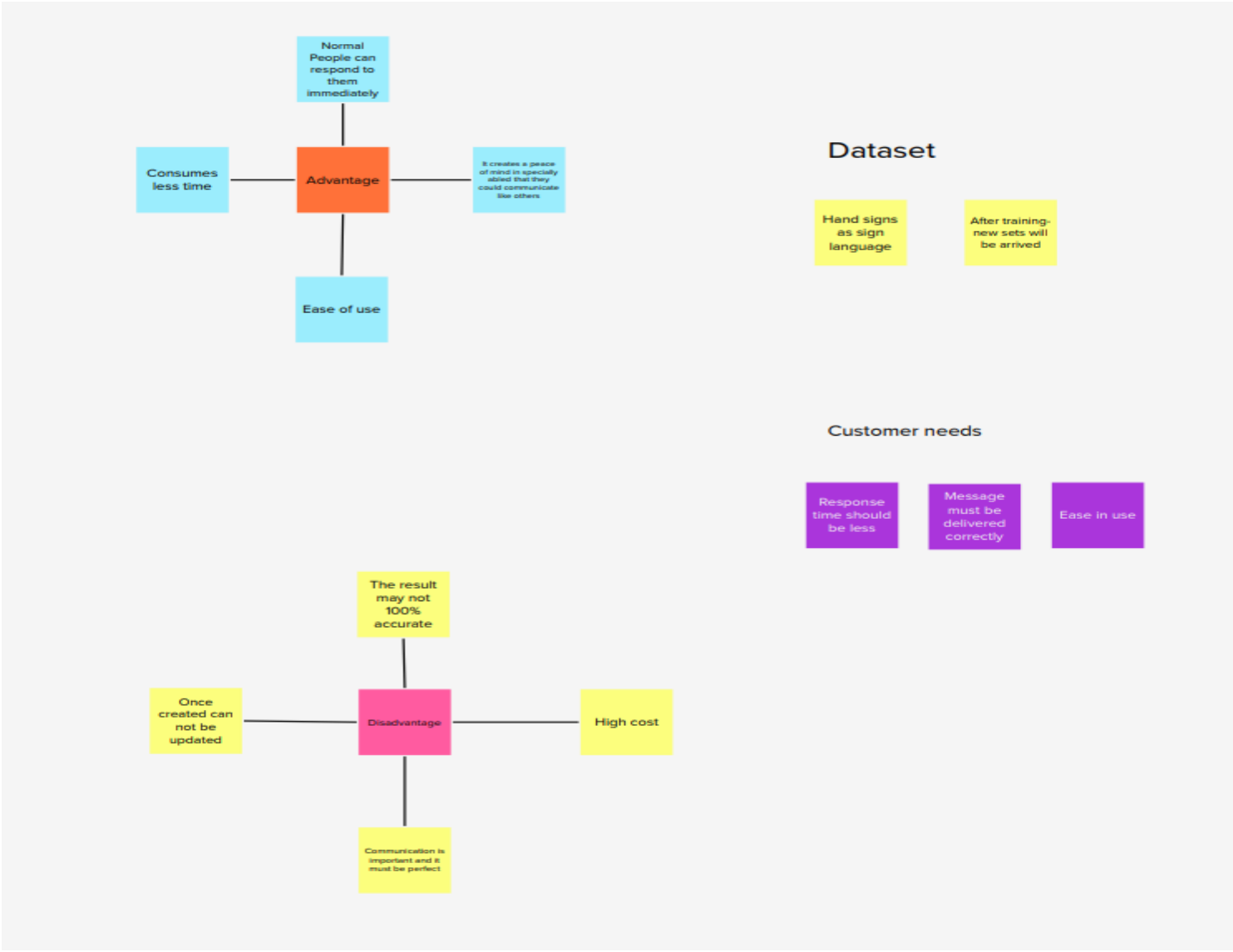
- The data set is done by using handheld gloves
- Every alphabets are given in the form of signs which is the data set
- Using this data set the machine is trained to give the letter as output
- Then the text is converted to speech using the text-to-speech converter

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

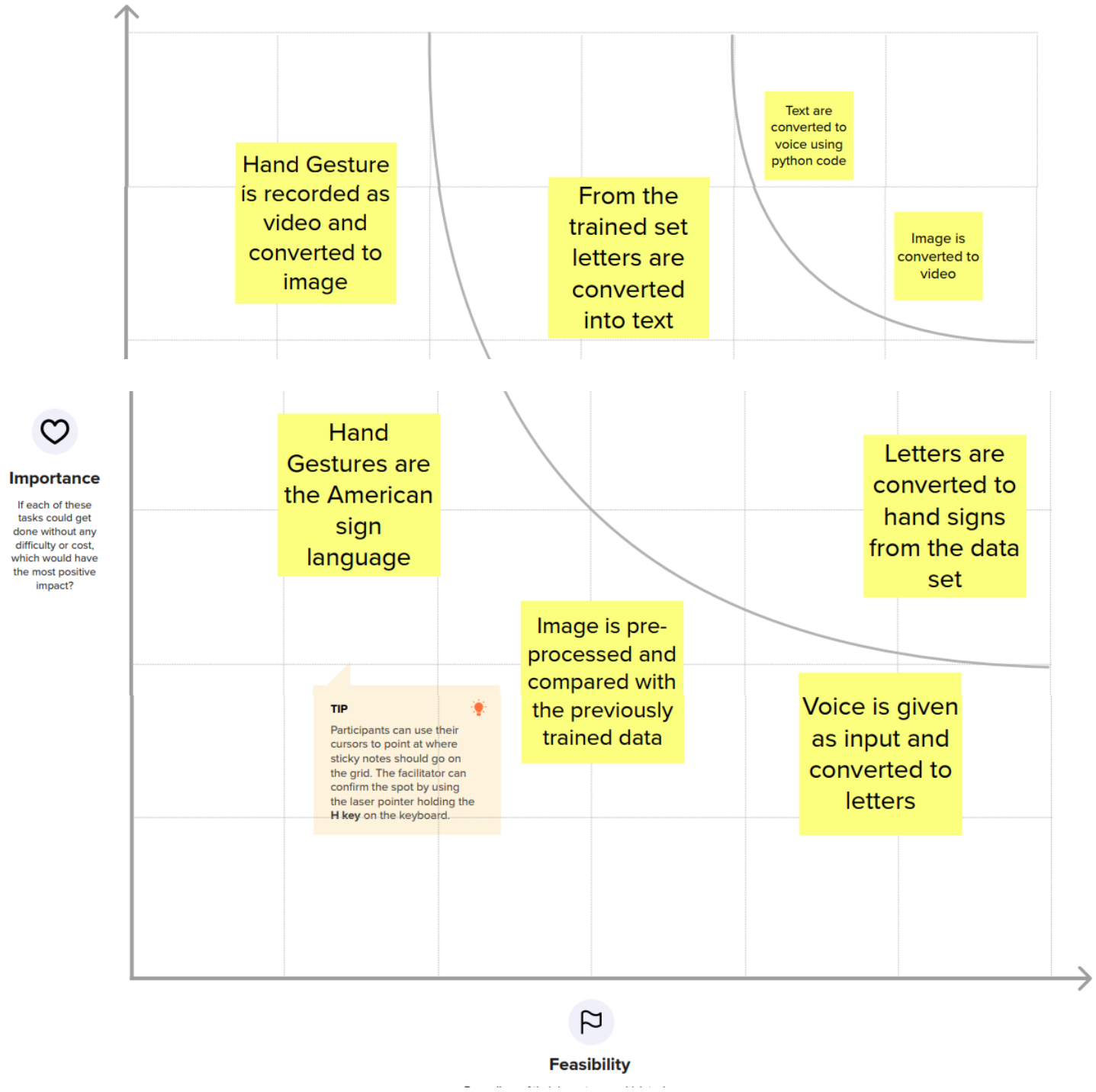


4

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.

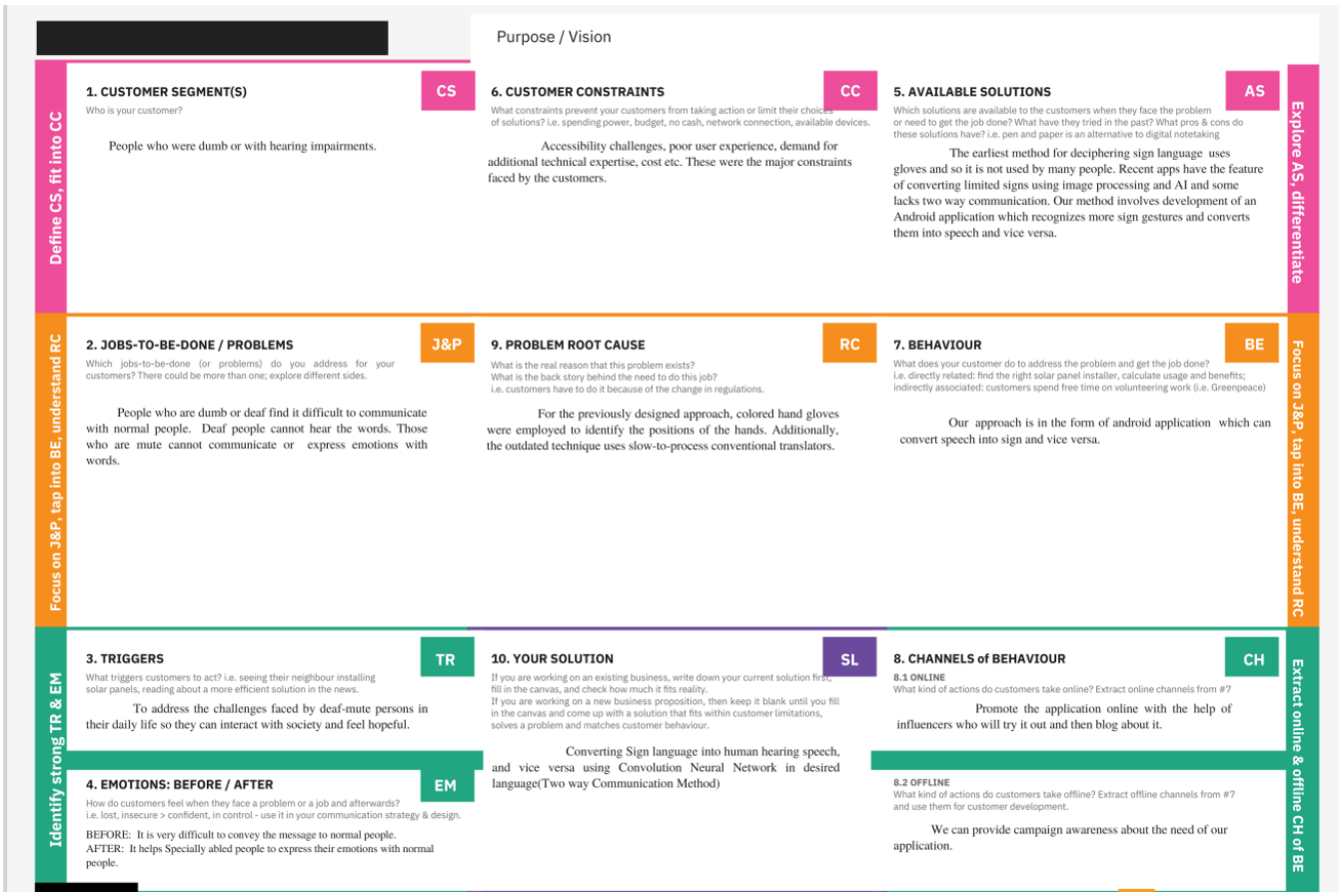
20 minutes


Hand Gesture is recorded as video and converted to image
From the trained set letters are converted into text
Text are converted to voice using python code
Image is converted to video
Hand Gestures are the American sign language
Image is pre-processed and compared with the previously trained data
Letters are converted to hand signs from the data set
Voice is given as input and converted to letters
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility

PROPOSED SOLUTION

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	• The main objective is to build a communication system which enables communication between a speech-hearing impaired and a normal person
2.	Idea / Solution description	• Converting sign language into voice and text in the desired language (two-way communication) using Convolutional Neural Network technology
3.	Novelty / Uniqueness	• Deploying and improving our solution to provide faster response in Desired language.
4.	Social Impact / Customer Satisfaction	• Improves the Communication between Normal People and Deaf-Dumb. • It increases the scope for career development. • It will shatter all the barriers and will help to enhance their skills in a positive manner
5.	Business Model (Revenue Model)	• We will provide campaign awareness about the need of our application. • The application will be made available to more recipients, which will accelerate growth
6.	Scalability of the Solution	• The user will find it very simple to use and update. • Encoding the errors and decoding with better accuracy

PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

FRNo.	Functional Requirement(Epic)	Sub Requirement (Story/Sub-Task)
FR-1	**User Registration**	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	**UserConfirmation**	Confirmation via Email Confirmation via OTP
FR-3	**Dataset**	Collection of dataset of sign images of alphabets

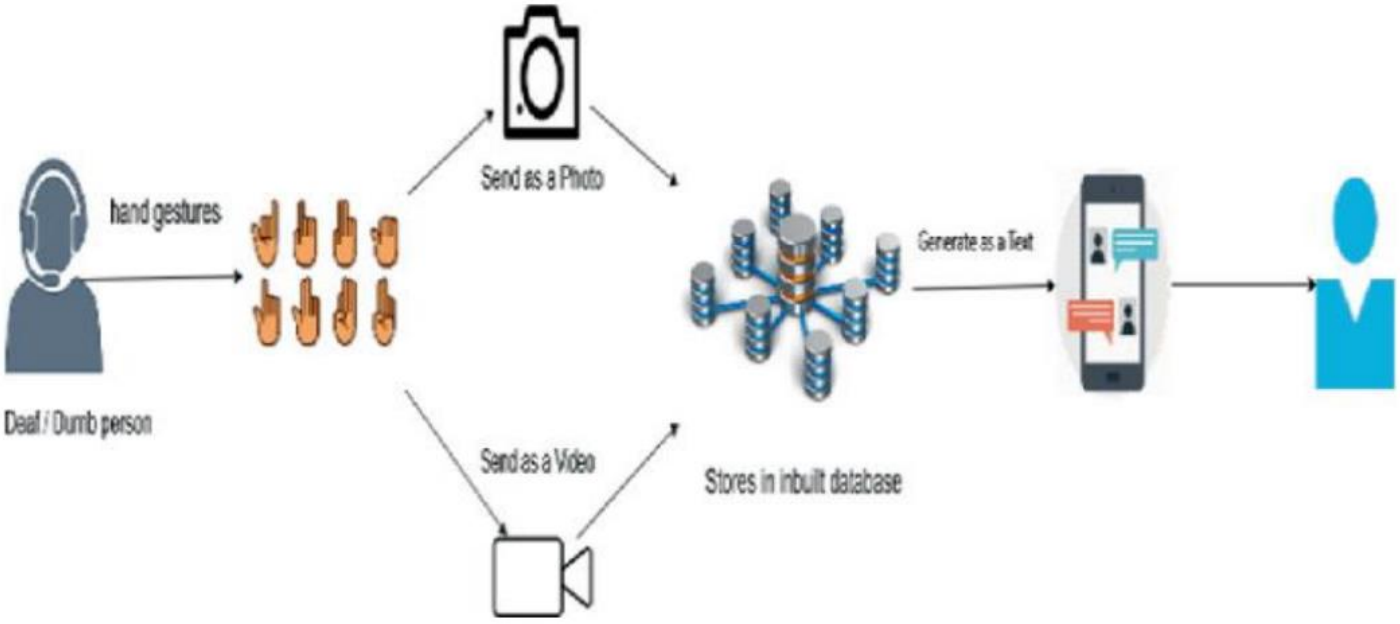
NON FUNCTIONAL REQUIREMENTS

FRNo.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is useful for specially abled people to communicate normally with a normal person
NFR-2	**Security**	The user information will be secured and their conversation will also be secured
NFR-3	**Reliability**	It supports sign to voice conversion and voice to sign conversion from the available dataset
NFR-4	**Performance**	It is an user friendly interface
NFR-5	**Availability**	It is adaptable to any type of device and it will be accessible at any time
NFR-6	**Scalability**	It can also support group conversation

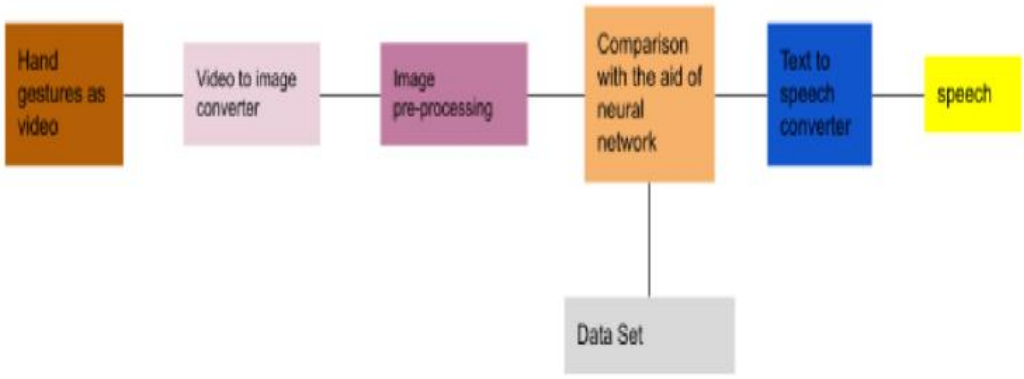
CHAPTER 5

PROJECT DESIGN

DATA FLOW DIAGRAM



SOLUTION & TECHNICAL ARCHITECTURE



CHAPTER 6

PROJECT PLANNING AND SCHEDULING

USER STORIES AND ESTIMATION

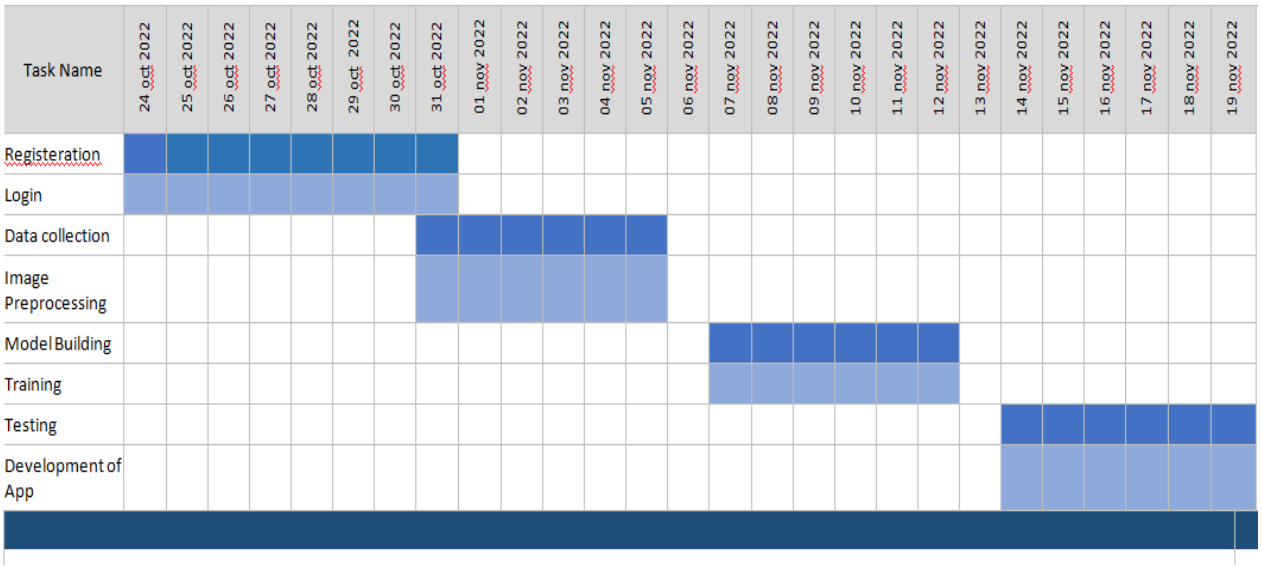
Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, we can register the application by entering my email, password, and confirming my password through mail.	**5**	High	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika
	Login	USN-2	As a user, we can log into the application by entering email & password.	**5**	High	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika
Sprint -2	Data Collection	USN-3	As a user, we can collect Data Sets	5	High	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika
	Image preprocessing	USN-4	As a user , we can do image preprocessing techniques on our Datasets.	5	High	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika
Sprint-3	Model Building	USN-5	As a user we can initialize the model with required layers.	5	High	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika

	Training	USN-6	As a user,we can train the image classification model by using CNN model.	5	Medium	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika
Sprint-4	Testing	USN-7	As a user.we can test the performance of our model.	10	Medium	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika
	Development of App	USN-8	As a user, finally we convert the text tospeech.	10	Medium	Angelene Vidhya Madhumitha Anu Lavanya Kiruthika

SPRINT DELIVERY SCHEDULE

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	7 Days	24 Oct 2022	29 Oct 2022	10	Oct 2022
Sprint-2	10	7 Days	31 Oct 2022	05 Nov 2022	10	Nov 2022
Sprint-3	10	7 Days	07 Nov 2022	12 Nov 2022	10	Nov 2022
Sprint-4	10	7 Days	14 Nov 2022	19 Nov 2022	10	Nov 2022

SPRINT PLANNING :



CHAPTER 7

CODING & SOLUTIONING

Image Processing :

```
In [11]: 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)

In [23]: x_train=train_datagen.flow_from_directory('Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical'

         Found 15750 images belonging to 9 classes.

In [24]: x_test=test_datagen.flow_from_directory('Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color

         Found 2250 images belonging to 9 classes.
```

Test and Train :

```
In [45]: from keras.models import load_model
         from tensorflow.keras.preprocessing import image
         import numpy as np
         import cv2

         model=load_model('aslpng1.h5')

In [46]: model=load_model('aslpng1.h5')

In [47]: from skimage.transform import resize
         def detect(frame):
             img=image.img_to_array(frame)
             img=resize(frame,(64,64,1))
             img=np.expand_dims(img,axis=0)
             pred=np.argmax(model.predict(img))
             op=['A','B','C','D','E','F','G','H','I']
             print("THE PREDICTED LETTER IS ",op[pred])

In [48]: frame=cv2.imread(r"C:\Users\hp\Desktop\Dataset\test_set\A\1.png")
         data=detect(frame)

         1/1 [==============================] - 3s 3s/step
         THE PREDICTED LETTER IS  A
```

Training :

```
In [31]: from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import Convolution2D
         from keras.layers import MaxPooling2D
         from keras.layers import Dropout
         from keras.layers import Flatten
```

```
In [26]: model=Sequential()
```

```
In [27]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
```

```
In [28]: model.add(MaxPooling2D(pool_size=(2,2)))
```

```
In [29]: model.add(Flatten())
```

```
In [30]: model.add(Dense(units=512,activation='relu'))
```

```
In [32]: model.add(Dense(units=9,activation='softmax'))
```


Testing :

```
In [33]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
In [34]: model.fit_generator(x_train,steps_per_epoch=24,epochs=10,validation_data=x_test,validation_steps=40)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_7284\3736481484.py:1: UserWarning: `Model.fit_generator` is deprecated and will b
e removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=24,epochs=10,validation_data=x_test,validation_steps=40)
Epoch 1/10
24/24 [==============================] - ETA: 0s - loss: 0.8844 - accuracy: 0.7103WARNING:tensorflow:Your input ran out of
data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` bat
ches (in this case, 40 batches). You may need to use the repeat() function when building your dataset.
24/24 [==============================] - 67s 2s/step - loss: 0.8844 - accuracy: 0.7103 - val_loss: 0.3296 - val_accuracy:
0.9164
Epoch 2/10
24/24 [==============================] - 46s 2s/step - loss: 0.2181 - accuracy: 0.9423
Epoch 3/10
24/24 [==============================] - 43s 2s/step - loss: 0.1158 - accuracy: 0.9664
Epoch 4/10
24/24 [==============================] - 37s 2s/step - loss: 0.0742 - accuracy: 0.9813
Epoch 5/10
24/24 [==============================] - 38s 2s/step - loss: 0.0517 - accuracy: 0.9872
Epoch 6/10
24/24 [==============================] - 39s 2s/step - loss: 0.0332 - accuracy: 0.9932
Epoch 7/10
24/24 [==============================] - 38s 2s/step - loss: 0.0238 - accuracy: 0.9943
Epoch 8/10
24/24 [==============================] - 37s 2s/step - loss: 0.0202 - accuracy: 0.9959
Epoch 9/10
```


sprint2.ipynb

CHAPTER 8

TESTING

Training on cloud:

In [1]:
```
pwd
```

Out[1]:
```
'/home/wsuser/work'
```

In [6]:
```
!pip install tensorflow==2.5.0
```

```
Requirement already satisfied: tensorflow==2.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.5.0)
Requirement already satisfied: grpcio~=1.34.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (1.34.1)
Requirement already satisfied: tensorflow-estimator<2.6.0,>=2.5.0rc0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow=
=2.5.0) (2.5.0)
Requirement already satisfied: keras-nightly~=2.5.0.dev in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (2.5.
0.dev2021032900)
Requirement already satisfied: astunparse~=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (1.6.3)
Requirement already satisfied: opt-einsum~=3.3.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (3.3.0)
Requirement already satisfied: google-pasta~=0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (0.2.0)
Requirement already satisfied: h5py~=3.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (3.1.0)
Requirement already satisfied: termcolor~=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (1.1.0)
Requirement already satisfied: tensorboard~=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (2.7.0)
Requirement already satisfied: numpy~=1.19.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (1.19.5)
Requirement already satisfied: keras-preprocessing~=1.1.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (1.
1.2)
Requirement already satisfied: absl-py~=0.10 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (0.12.0)
Requirement already satisfied: six~=1.15.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (1.15.0)
Requirement already satisfied: protobuf>=3.9.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0) (3.19.1)
Requirement already satisfied: gast~=0.4.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow~=2.5.0) (0.4.0)
```

In [8]:
```
!pip install keras==2.2.4
```

```
Requirement already satisfied: keras==2.2.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.2.4)
Requirement already satisfied: keras-applications>=1.0.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.0.8)
Requirement already satisfied: h5py in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (3.1.0)
Requirement already satisfied: numpy>=1.9.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.19.5)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.1.2)
Requirement already satisfied: pyyaml in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (5.4.1)
Requirement already satisfied: scipy>=0.14 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.7.3)
Requirement already satisfied: six>=1.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.15.0)
```

In [14]:
```
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='Xe3-hQZnoRL5BwTdvUwRSBwvaklNN6CLCjOkmoHTf4ta',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

bucket = 'projectdeploy-donotdelete-pr-fdqy9pile1ckg1'
object_key = 'conversation engine for deaf and dumb.zip'

streaming_body_1 = 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/
```

In [15]:
```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [16]:
```
ls
```

```
Dataset/
```

In [17]:
```
import tensorflow as tf
import keras_preprocessing
from keras_preprocessing import image
from keras_preprocessing.image import ImageDataGenerator
```

```
2022-11-17 17:01:38.853942: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0';
dlerror: libcudart.so.11.0: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/ibm/dsdriver/lib:/opt/oracle/lib:/o
pt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow
```

In [18]:
```
training_datagen = ImageDataGenerator(
      rescale = 1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')
```

In [19]:
```
train_generator = training_datagen.flow_from_directory(
    r"Dataset/training_set",
    target_size=(64,64),
    class_mode='categorical',
  batch_size=30
)
```

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

In [22]:
```
model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(train_generator, epochs=10,validation_data=test_generator)
```

```
Epoch 1/10
525/525 [==============================] - 170s 324ms/step - loss: 1.1232 - accuracy: 0.6027 - val_loss: 0.7268 - val_accuracy: 0.7520
Epoch 2/10
525/525 [==============================] - 171s 325ms/step - loss: 0.5525 - accuracy: 0.8041 - val_loss: 0.4627 - val_accuracy: 0.8516
Epoch 3/10
525/525 [==============================] - 170s 325ms/step - loss: 0.4251 - accuracy: 0.8524 - val_loss: 0.4056 - val_accuracy: 0.8747
Epoch 4/10
525/525 [==============================] - 170s 324ms/step - loss: 0.3452 - accuracy: 0.8806 - val_loss: 0.4179 - val_accuracy: 0.8596
Epoch 5/10
525/525 [==============================] - 173s 329ms/step - loss: 0.3180 - accuracy: 0.8903 - val_loss: 0.3081 - val_accuracy: 0.9107
Epoch 6/10
525/525 [==============================] - 172s 327ms/step - loss: 0.2838 - accuracy: 0.9014 - val_loss: 0.2897 - val_accuracy: 0.9044
Epoch 7/10
525/525 [==============================] - 170s 324ms/step - loss: 0.2651 - accuracy: 0.9073 - val_loss: 0.2681 - val_accuracy: 0.9111
Epoch 8/10
525/525 [==============================] - 169s 323ms/step - loss: 0.2452 - accuracy: 0.9178 - val_loss: 0.2880 - val_accuracy: 0.9120
Epoch 9/10
525/525 [==============================] - 172s 327ms/step - loss: 0.2345 - accuracy: 0.9195 - val_loss: 0.2445 - val_accuracy: 0.9178
Epoch 10/10
525/525 [==============================] - 173s 330ms/step - loss: 0.2253 - accuracy: 0.9223 - val_loss: 0.2259 - val_accuracy: 0.9369
```

In [23]:
```
model.save("aslpng1.h5")
```

In [25]:
```
!tar -zcvf sign-detection-model.tgz aslpng1.h5
```

```
aslpng1.h5
```

In [26]:
```
!pip install ibm_watson_machine_learning
```

```
Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning)
(2.11.0)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning)
(4.8.2)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learnin
g) (1.3.4)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_w
atson_machine_learning) (0.10.0)
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==2.11.
*->ibm_watson_machine_learning) (2.11.0)
```

In [27]:
```
from ibm_watson_machine_learning import APIClient
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/torch/package/_directory_reader.py:17: UserWarning: Failed to initialize NumPy: module co
mpiled against API version 0xe but this version of numpy is 0xd (Triggered internally at  /opt/conda/conda-bld/pytorch-base_1658814980981/work/t
orch/csrc/utils/tensor_numpy.cpp:68.)
  _dtype_to_storage = {data_type(0).dtype: data_type for data_type in _storages}
```

In [28]:
```
wml_credentials ={
    "url":"https://eu-de.ml.cloud.ibm.com",
    "apikey":"Xe3-hQZnoRL5BwTdvUwRSBwvaklNN6CLCjOkmoHTf4ta"
}
```

In [29]:
```
client=APIClient(wml_credentials)
```

In [30]:
```
client
```

Out[30]:
```
<ibm_watson_machine_learning.client.APIClient at 0x7f0fc029a0a0>
```

In [31]:
```
client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  ------------  -------------------------
ID                                    NAME          CREATED
fa7248cd-8c17-47b0-bdbb-aebd61bece06  Deploy_cloud  2022-11-17T16:03:06.058Z
------------------------------------  ------------  -------------------------
```

In [32]:
```
space_id="fa7248cd-8c17-47b0-bdbb-aebd61bece06"
```

In [33]:
```
client.set.default_space(space_id)
```

Out[33]:
```
'SUCCESS'
```

In [34]:
```
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

------------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

In [35]:
```
software_space_uid = client.software_specifications.get_id_by_name("tensorflow_rt22.1-py3.9")
```

In [36]:
```
software_space_uid
```

Out[36]:
```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

In [37]:
```
model_details = client.repository.store_model(model='sign-detection-model.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"cnn model",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

CHAPTER 9

RESULTS

In [45]:
```
from keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
```

model=load_model('aslpng1.h5')

In [46]:
```
model=load_model('aslpng1.h5')
```

In [47]:
```
from skimage.transform import resize
def detect(frame):
    img=image.img_to_array(frame)
    img=resize(frame,(64,64,1))
    img=np.expand_dims(img,axis=0)
    pred=np.argmax(model.predict(img))
    op=['A','B','C','D','E','F','G','H','I']
    print("THE PREDICTED LETTER IS ",op[pred])
```

In [48]:
```
frame=cv2.imread(r"C:\Users\hp\Desktop\Dataset\test_set\A\1.png")
data=detect(frame)
```

```
1/1 [==============================] - 3s 3s/step
THE PREDICTED LETTER IS  A
```

CHAPTER 10
ADVANTAGES &DISADVANTAGES

ADVANTAGES

- Provides an opportunity for a deaf-dumb people to communicate with non-signing people without the need of an interpreter.

DISADVANTAGES

- Sign language **requires the use of hands to make gestures**. This can be a problem for people who do not have full use of their hands.

CHAPTER 11

CONCLUSION

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.

CHAPTER 12

FUTURE SCOPE

.

There are many more works that can be carried out as an extension of this project. This system predicts the need of the mute person but future systems may be developed that could communicate to the mute person's mobile device, allowing the system to learn the needs of the user, thereby provisioning the development of recommendatory systems as they have the relevant data related to the mute person that can easily be learned thought he neural network model.

APPENDIX

GITHUB

https://github.com/IBM-EPBL/IBM-Project-1619-1658403199

PROJECTDEMO

https://www.kapwing.com/videos/638077d8ef7bb30194762446