REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

NALAIYA THIRAN PROJECT REPORT

IBM-Project-35171-1660282130

Team ID : PNT2022TMID09339

Submitted by

ROSHNI S	**(310619104110)**
PRIYA DHARSHINI S	**(310619104097)**
RIYA P	**(310619104107)**
SANJU SHREE K	**(310619104116)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINNERING

IN

COMPUTER SCIENCE AND ENGINEERING

EASWARI ENGINEERING COLLEGE

(AUTONOMOUS)

CHENNAI-600089

PROJECT REPORT

ABSTRACT:

Artificial Intelligence(AI) has proved its power and efficiency into different domains(industry, medicine, economic, etc), hoping for giving a chance for people with disabilities to change their life for the best. Basically, AI offers many advanced tools and machine learning algorithms that could support human capabilities, needs and preferences and improve User Experience (UX). In this paper, we focus on the impact of AI in Human Machine Interaction. Likewise, we aim to demonstrate what is the impact of AI on disability Human life. Finally, we proceed by introducing a revolutionary approach of adaptive user interfaces to users with special needs.

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Sign language recognition is the process of translating the user's gestures and signs into text. It aids those who are unable to interact with the general population in communication Using image processing methods and neural networks, the motion is mapped to pertinent text in the training data, transforming unprocessed photos and videos into text that can be read and understood. People who are dumb are typically prohibited from having regular conversations with other people in society. They sometimes struggle to communicate with regular people through gestures because the majority of people only recognise a small number of them. People who are deaf or have hearing loss are unable to communicate vocally, so they must frequently use some type of visual communication. The primary form of communication for the deaf and dumb community is sign language. Similar to other languages, it contains grammar and vocabulary, but it communicates primarily through images.

1.2 PURPOSE

The recognition of sign language gestures from real time video and successfully classifying it into either one from a list of categories have been a popular and challenging field of research. Many researchers have been working on this field for a long time, so we have also thought of contributing to this field as by working on it in our final year major project. Liang et al. have also put their research on this concept which has guided us throughout the implementation. The process of recognizing a sign language gesture and classifying it is the single line definition of the task performed by this proposed system. Along with this, a text to ASL finger spelling feature is also available that makes the two-way communication from sign to text and text to sign

possible. The following steps were taken while working on this project. Many vision-based and sensor-based techniques have been used for sign language recognition. Pavlovic et al. The paper published on 1997 emphasizes on the advantages and shortcomings and important differences in the gesture interpretation approaches depending on whether a 3D model of thehuman hand or an image appearance model of the human hand is used. As of the time, this survey was done 3D hand models offered a way of more elaborate modeling of hand gestures but lead to computational hurdles that had not been overcome given the real-time requirements of HCI.

2. LITERATURE REVIEW

S.NO	PAPER TITLE	AUTHORS	DESCRIPTION
1	Portable Communication Aid for Specially Challenged: Conversion of Hand Gestures into Voice and Vice Versa	T Meera Devi, K M Shravan Raju	The goal of the project is to provide a portable communication tool for people with disabilities who have trouble adequately communicating with healthy people. The process of identifying the characteristic hand gesture that sets it apart involves several steps. Using a neural network, the gathered gesticulation is taught. A continuous recording of gestures is isolated from the hand movement pattern. The gestural section represents low-level comprehension of the feature pattern. This will let normal people and those with disabilities communicate with one another.
2	Real-Time Two-Way Communication Approach for Hearing Impaired and Dumb Person Based on Image Processing.	Shweta. S. Shinde, Rajesh M. Autee, Vitthal K. Bhosal	The proposed system employs a method for hand recognition based on vision. The hand movements are recognised in various lighting situations. The suggested method segments the hand's background using the data that has been collected, and then each letter is given a specific gesture. It uses feature extraction techniques to

			determine hand motions' peak and angle values. By translating the motions into speech and vice versa, the gestures are finally recognised. Mel-frequency cepstrum coefficients and dynamic temporal warping are utilised to extract the voice signal's characteristics. The suggested system is MATLAB-based.
3	Full Duplex Communication System for Deaf & Dumb People	Shraddha R. Ghorpade, Surendra K. Waghamare (2015)	People with disabilities are having a difficult time keeping up with the rapidly advancing 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. Deaf and stupid individuals typically use sign language for however, they struggle to communicate with people who don't comprehend sign language.
4	Smart communication for differently abled people	R. Bhavani , B. Poornima, M. Surya Bharathi , M. Saraswathi	The majority of the tasks we perform on a daily basis involve speaking and hearing. People who are deaf or dumb find it challenging to communicate with those who do not comprehend sign language or misinterpreters. In this study, we developed a straightforward embedded system-based solution to address this issue. To collect data from the deaf and dumb using sign language, we used a flex sensor. The user of the android-based voice software will speak into it when he or she is deaf and needs to communicate. The programme will then convert this specific speech to text, which will be displayed on LCD. Two Flex sensors are utilised to play whatever messages that Dumb People want to play for the user.

5	Real-Time Recognition of Indian Sign Language	Muthu Mariappan H, Dr Gomathi V	The real-time sign language recognition system is designed to identify Indian Sign Language motions (ISL). Sign languages often only use hand gestures and facial emotions. The skin segmentation function of OpenCV is used to identify and track the Regions of Interest (ROI) for the purpose of recognising the indications. By using the fuzzy c-means clustering machine learning algorithm, hand motions are trained and predicted. Applications for gesture recognition include game control, Human-Computer Interaction (HCI), sign language interpretation, and gesture-controlled robots and automated houses. Real-time signs are recognised using the suggested approach. Therefore, it is highly helpful for those with hearing and speech impairments to communicate with other individuals.

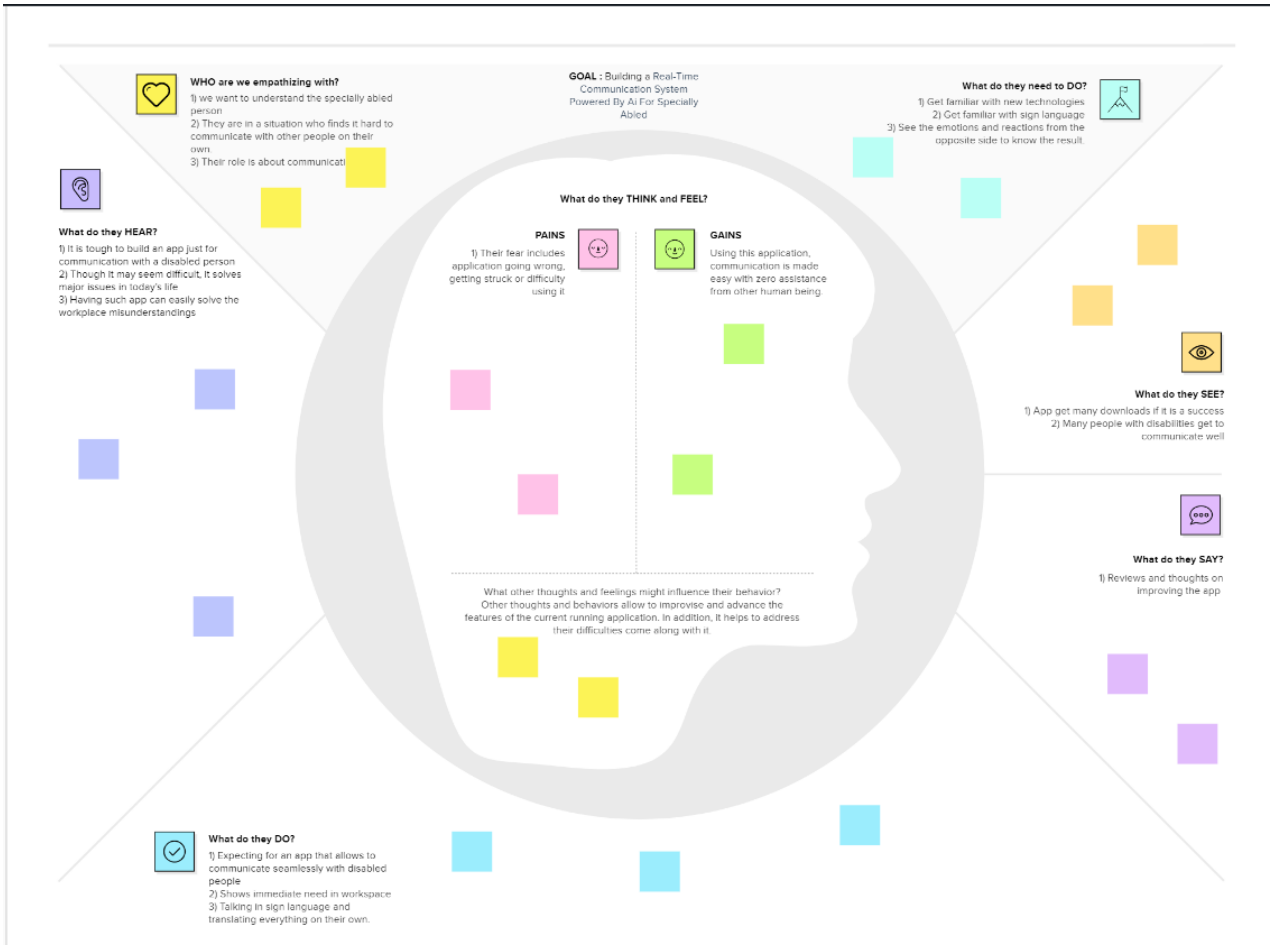
2.1 EXISTING

PROBLEM STATEMENT DEFINITION

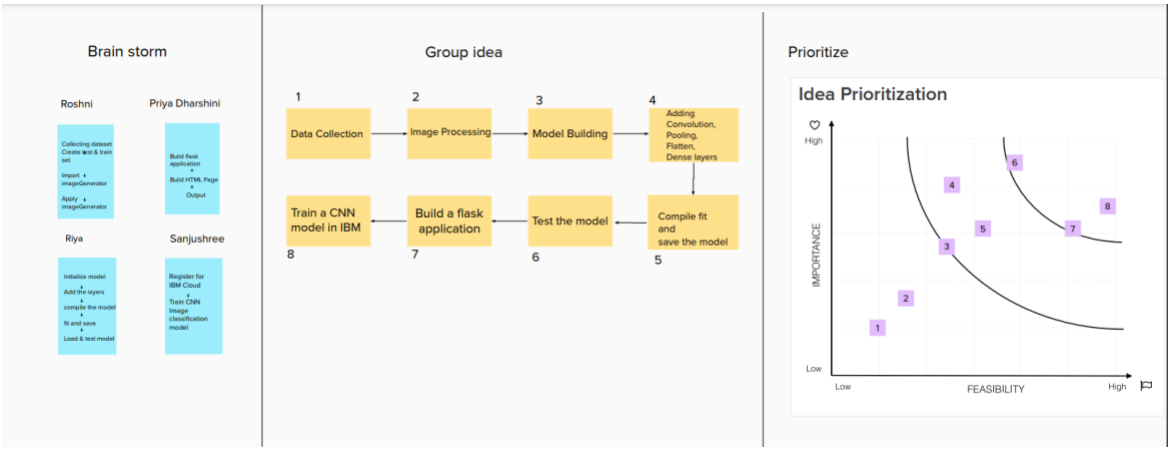
The sign language is used widely by people who are deaf-dumb these are used as a medium for communication. A sign language is nothing but composed of various gestures formed by different shapes of hand, its movements, orientations as well as the facial expressions. There are around 466 million people worldwide with hearing loss and 34 million of these are children. `Deaf' people have very little or no hearing ability. They use sign language for communication. People use different sign languages in different parts of the world. Compared to spoken languages they are very less in number. In existing system, lack of datasets along with variance in sign language with locality has resulted in restrained efforts in finger gesture detection. Existing project aims at taking the basic step in bridging the communication gap between normal people and deaf and dumb people using Indian sign language. Effective extension of this project to words and common expressions may not only make the deaf and dumb people communicate faster and easier with outer world, but also provide a boost in Developing autonomous systems for understanding and aiding them. The Indian Sign Language lags behind its American Counterpart as the research in this field is hampered by the lack of standard datasets

1. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	An application for deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech in Artificial Intelligence.
2	Idea / Solution description	• By using voice conversation system with hand gesture recognition and translation will be very useful to havea proper conversation. • This makes two people to communicate in a easier and efficient way
3	Novelty / Uniqueness	We are using Convolution neural network to create a model that is trained on different hand gestures andan app is built for the use this mode. • AI has been deployed/used in an efficient manner in the developmentof this idea.
4	Social Impact / Customer Satisfaction	Communicating with others and being connected in the society andremove accessibility barriers. • With clear results, two people can communicate with each other.
5	Business Model (Revenue Model)	• First, we offer free usage for everyone. Once our product is familiarized among people, we will turn the users to get the premium subscriptions by offering them withmore premium features. • As long as our product is beneficial to the users, subscriptions will increase which is a great business

		model, as this worked in case of manytop MNC's around the world.
5	Problem Statement (Problem to be solved)	• Useful in organization where communication between co-workersis much important. • Can make collaborations with government. • Example: Tata Group made a collaboration with Indian governmentin educating people through Cutting- Edge, which is a great success.

3.4 Problem Fit Solution

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer? Specially-abled people and normal people who want to make communication

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? Always wait by depending on someone to get the work done.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done?
Make use of other people's assistance
What have they tried in the past?
Learning sign-language
What pros & cons do these solutions have?
Cons: always dependent on someone
Pros: Reliable results

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? Get familiar with technology and electronic devices.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
This problem exists because there was no system that can help specially abled people communicate easily without others' help. What is the back story behind the need to do this job?
To interact with more independency on others.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act?
seeing other people communicate well.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
lost, insecure > confident, in control
less communicative > easily communicative

10. YOUR SOLUTION SL

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.

8. CHANNELS of BEHAVIOUR CH

Check the correctness of the output.
Can seamlessly try any set of phrases.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Image Upload	Upload image through camera Upload image from Gallery
FR-4	System	Desktop/mobile with good quality camera
FR-5	Sign Language Conversion	Conversion of sign language into text using Convolutional Neural Network (CNN)
FR-6	System allowance/access	Provide system access to capture images/videos

Non-functional Requirements:

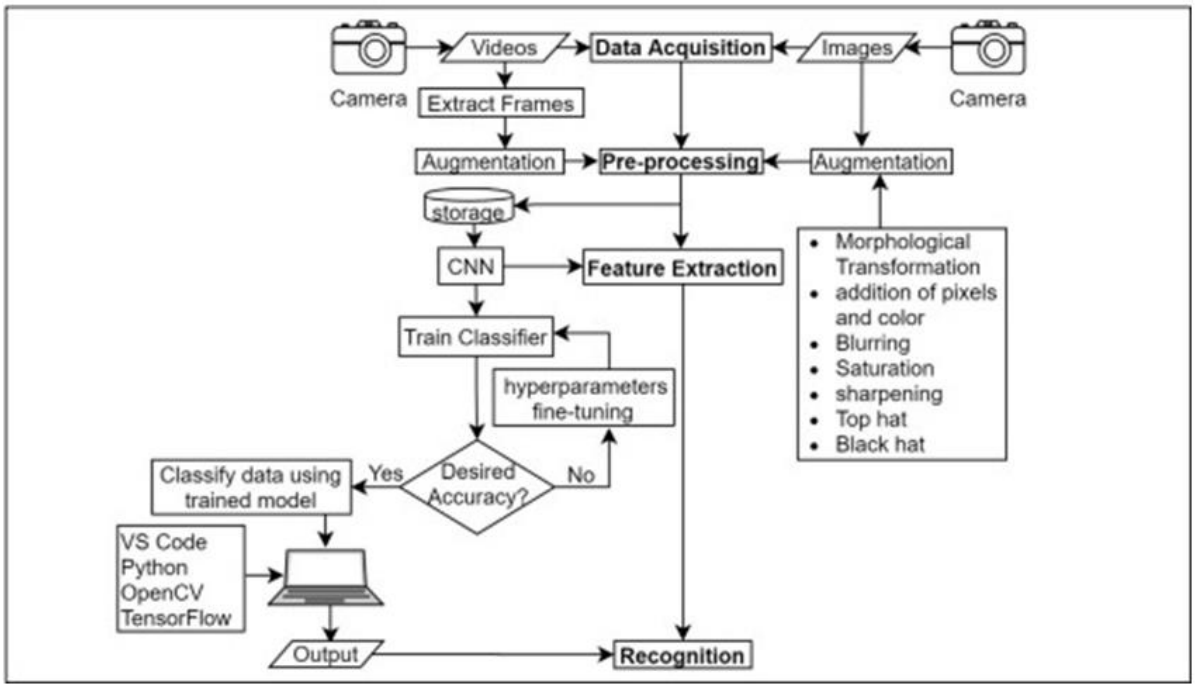
Following are the non-functional requirements of the proposed solution.

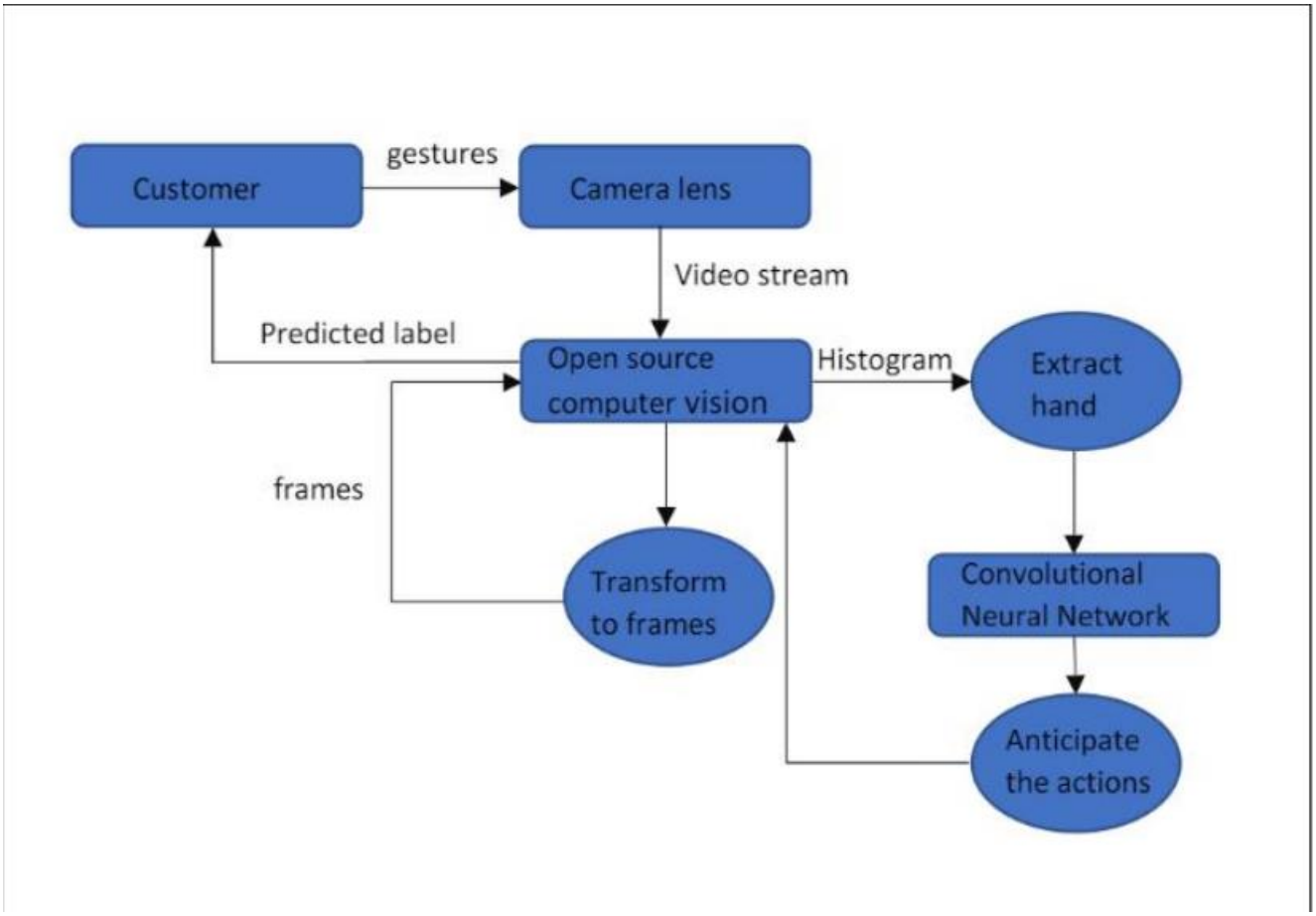
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This system is easy to use. It helps in translating sign language into our desired language.
NFR-2	**Security**	Only the user can sign in to the system and unauthorised users will be avoided at max.
NFR-3	**Reliability**	Accuracy, Predictability to a greater level
NFR-4	**Performance**	After receiving the sign language input from the user, translation is done faster that normal people can understand.
NFR-5	**Availability**	The system is accessible for every user at given point of time.
NFR-6	**Scalability**	Servers helps in achieving scalability whenever needed especially when the number of users increase.

5. PROJECT DESIGN

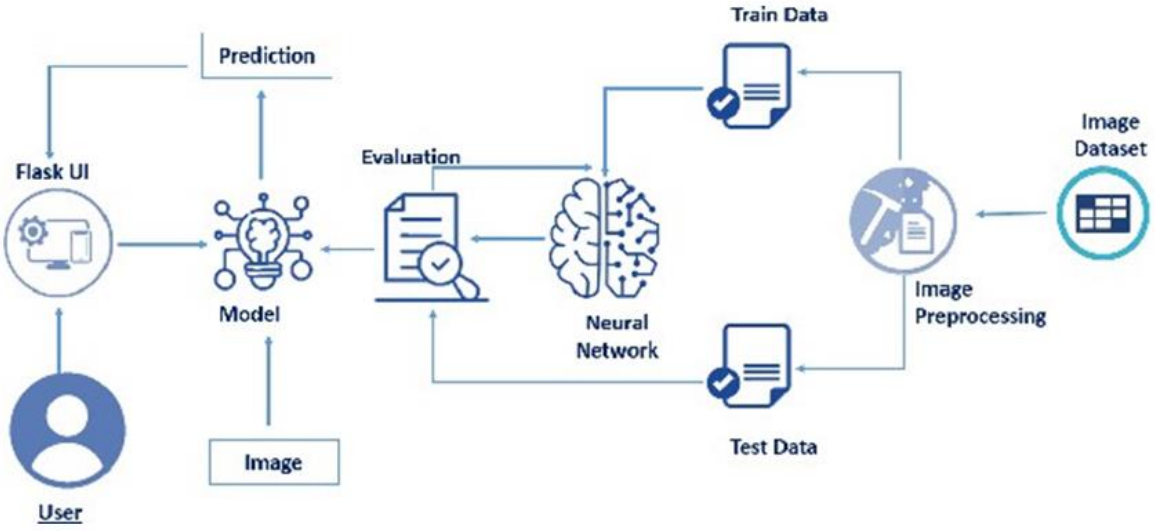
5.1 DATA FLOW DIAGRAMS

A data flow diagram is a two-dimensional diagram that explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in.





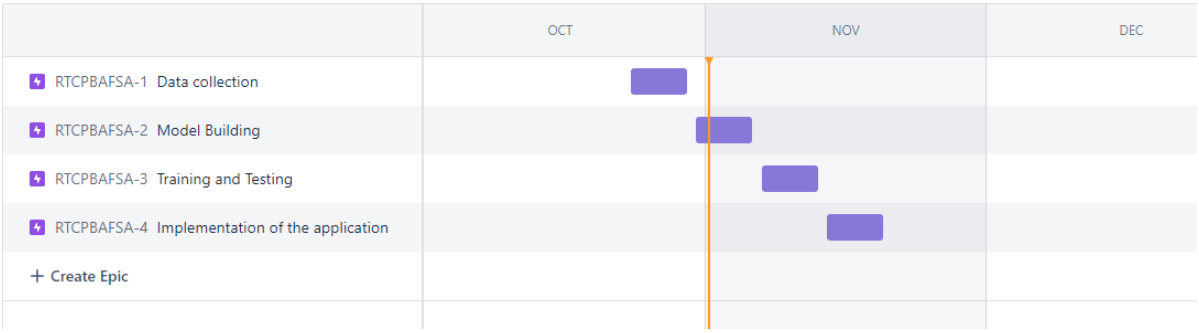
5.2 SOLUTION & TECHNICAL ARCHITECTURE



Software architecture involves the high level structure of software system abstraction, by using decomposition and composition, with architectural style and quality attributes. A software architecture design must conform to the major functionality and performance requirements of the system, as well as satisfy the non-functional requirements such as reliability, scalability, portability, and availability. Software architecture must describe its group of components, their connections, interactions among them and deployment configuration of all components.

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION



6.2 SPRINT DELIVERY SCHEDULE

Project Tracker, Velocity & Burndown Chart: (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	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	04 Nov 2022	5	04 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	11 Nov 2022	7	11 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	18 Nov 2022	5	18 Nov 2022

6.3 REPORTS FROM JIRA

JIRA has categorized reports in four levels, which are

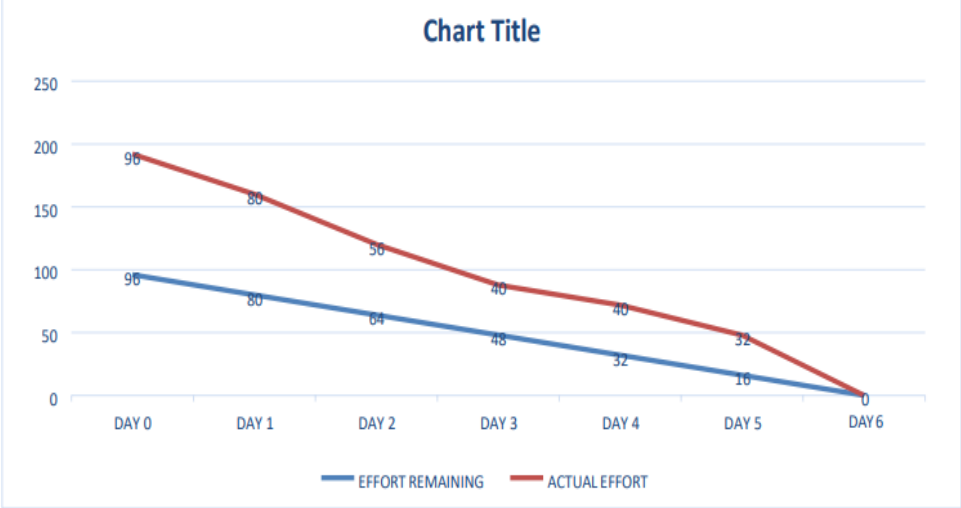
- Agile
- Issue Analysis
- Forecast & Management
- Others

Velocity:

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

$$AV = 6/10 = 0.6$$

Burndown chart:



SPRINT BURNDOWN

7. CODING & SOLUTION

7.1 FEATURE 1

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator as ig
train = ig(rescale = 1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test = ig(rescale=1./255)
xtrain = train.flow_from_directory(r'E:\ibm\IBM-Project-31230-1660197974\Data Collection\Dataset\training_set', target_size=(64,64),
batch_size=300, class_mode='categorical', color_mode= 'grayscale')
ytrain = train.flow_from_directory(r'E:\ibm\IBM-Project-31230-1660197974\Data Collection\Dataset\test_set', target_size=(64,64),
batch_size=300, class_mode='categorical', color_mode= 'grayscale')
```

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

In [5]:

```
from tens File display s.preprocessing.image import ImageDataGenerator as ig
train = ig 1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test = ig(rescale=1./255)
```

In [7]:

```
xtrain = train.flow_from_directory(r'E:\ibm\IBM-Project-31230-1660197974\Data Collection\Dataset\training_set', target_size=(64,64), batch_size=300, c
ytrain = train.flow_from_directory(r'E:\ibm\IBM-Project-31230-1660197974\Data Collection\Dataset\test_set', target_size=(64,64), batch_size=300, class
```

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

MODEL BUILDING

```
model = Sequential()
model.add(Convolution2D(32, (3,3), input_shape=(64,64,1), activation='relu')) #convolution layer
model.add(MaxPooling2D(pool_size=(2,2))) # pooling layer
model.add(Flatten()) #flatten layer
model.add(Dense(units=512, activation='relu')) #dense layer
model.add(Dense(units=9, activation='softmax'))  #dense layer
```

In [17]:

```
#compile the model
```

In [20]:

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

model.fit_generator(xtrain, steps_per_epoch=24, epochs=10, validation_data=xtest, validation_steps=40)  #fit the model
model.save('aslpng1.h5')  #saving the model
```

```
C:\Users\riyap\anaconda3\envs\tf-gpu\lib\site-packages\tensorflow\python\keras\engine\training.py:1940: UserWarning: `Model.fit_generator` is d
d and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/10
24/24 [==============================] - ETA: 0s - loss: 1.0995 - accuracy: 0.6942WARNING:tensorflow:Your input ran out of data; interrupting t
Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches (in this case, 40 batches). You may need to u
epeat() function when building your dataset.
24/24 [==============================] - 59s 2s/step - loss: 1.0995 - accuracy: 0.6942 - val_loss: 0.4567 - val_accuracy: 0.8680
Epoch 2/10
24/24 [==============================] - 22s 916ms/step - loss: 0.2447 - accuracy: 0.9319
Epoch 3/10
24/24 [==============================] - 15s 605ms/step - loss: 0.1207 - accuracy: 0.9684
Epoch 4/10
24/24 [==============================] - 12s 498ms/step - loss: 0.0734 - accuracy: 0.9800
Epoch 5/10
24/24 [==============================] - 9s 358ms/step - loss: 0.0536 - accuracy: 0.9861
Epoch 6/10
24/24 [==============================] - 8s 311ms/step - loss: 0.0422 - accuracy: 0.9893
Epoch 7/10
24/24 [==============================] - 7s 280ms/step - loss: 0.0286 - accuracy: 0.9930
Epoch 8/10
24/24 [==============================] - 7s 295ms/step - loss: 0.0241 - accuracy: 0.9939
Epoch 9/10
24/24 [==============================] - 6s 256ms/step - loss: 0.0215 - accuracy: 0.9949
Epoch 10/10
24/24 [==============================] - 7s 286ms/step - loss: 0.0155 - accuracy: 0.9966
```

TESTING MODEL:

```
from tensorflow.keras.models import load_model
import numpy as np
import cv2
import skimage
model = load_model('aslpng1.h5')
#preprocess the images
from skimage.transform import resize
def detect(frame):
    img = resize(frame, (64,64,1))
    img =np.expand_dims(img, axis=0)
    if(np.max(img)>1):
        img = img/255.0
    pred = model.predict(img)
    print(pred)
    pred = model.predict_classes(img)
    print(pred)
#Load and predict
frame = cv2.imread(r'E:\ibm\IBM-Project-31230-1660197974\Data Collection\Dataset\test_set\G\1.png')
data = detect(frame)
```

```
[[4.5008630e-08 5.9901827e-11 1.7369990e-06 3.4444175e-10 1.1011272e-09
  7.6871736e-09 9.9998164e-01 1.6525357e-05 1.1666364e-10]]
[6]
C:\Users\riyap\anaconda3\envs\tf-gpu\lib\site-packages\tensorflow\python\keras\engine\sequential.py:455: UserWarning: `model.predict_classes()` is depr
ecated and will be removed after 2021-01-01. Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class classificati
on    (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.
g. if it uses a `sigmoid` last-layer activation).
  warnings.warn('`model.predict_classes()` is deprecated and '
```

8. TESTING

TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.
Characteristics of a good test case:

3.5.1 Accurate: Exacts the purpose.

3.5.2 Economical: No unnecessary steps or words.

3.5.3 Traceable: Capable of being traced to requirements.

3.5.4 Repeatable: Can be used to perform the test over and over.

3.5.5 Reusable: Can be reused if necessary

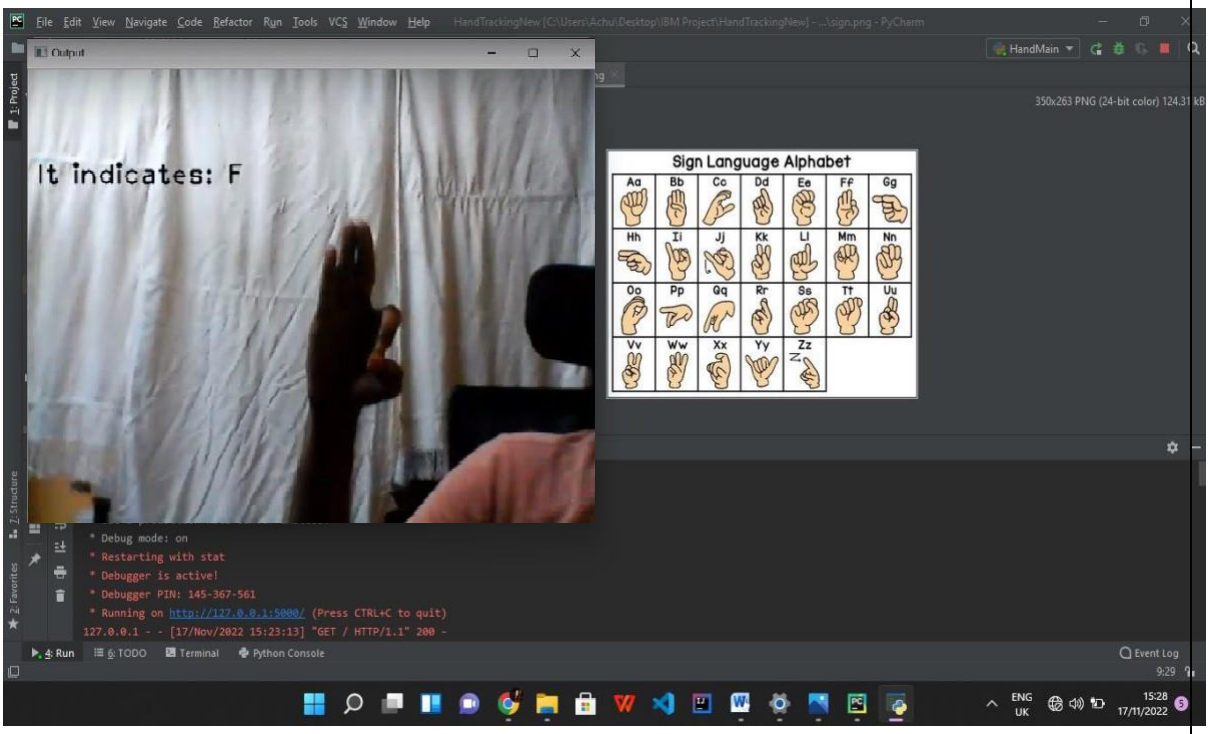
S.NO	FUNCTION	DESCRIPTION	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
1	Framework construction	Generate the GUI for admin and user	Individual page for admin and user	Individual page for admin and user	Success
2	Read the comments	Comments Analysis	Comments in text format	Comments in text format	Success
3	Classification	Classify the Datasets	Finger Gestures	Finger Gestures	Success
4	Rules implementation	Block the comments and friends	Block the users	Block the users	Success

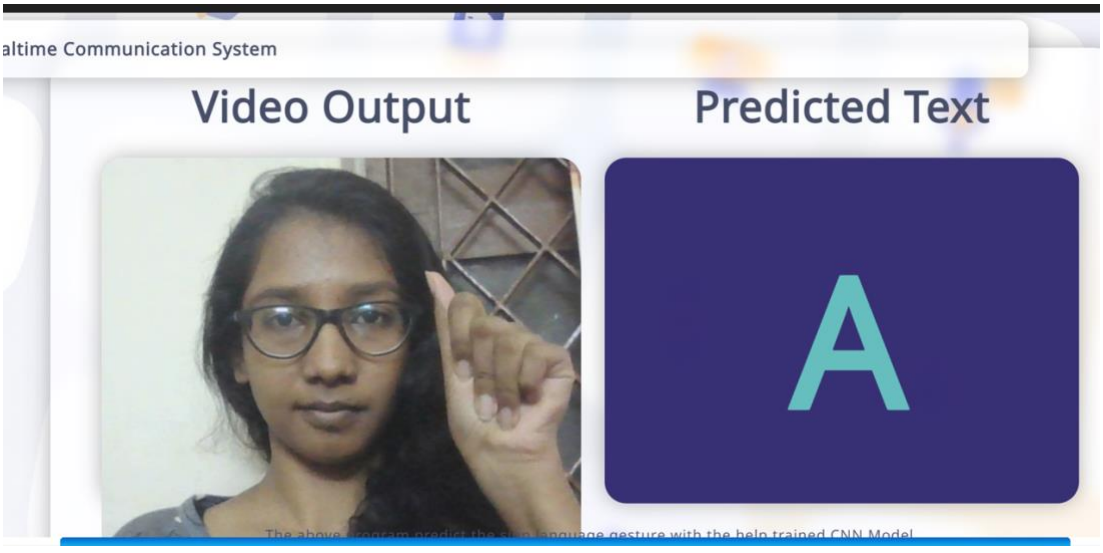
3.6 USER ACCEPTANCE TESTING

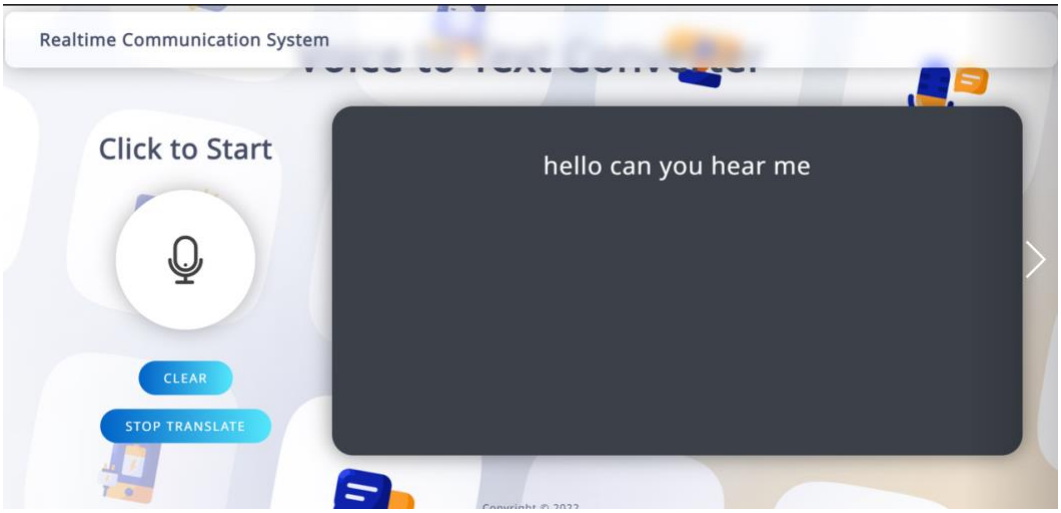
Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs are accepted by the database or other validations. For example accept only numbers in the numeric field, date format data in the date field. Also the null check for the not null fields. If any error occurs then show the error messages. The function of performance characteristics to specification and is accepted. A deviation from specification is uncovered and a deficiency list is created. User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

9. RESULTS

9.1 PERFORMANCE METRICS






Realtime Communication System
Click to Start
hello can you hear me
CLEAR
STOP TRANSLATE

10. ADVANTAGES & DISADVANTAGES

DISADVANTAGES

i. Need hardware control to detect the hands

ii. Hand segmentation become complex of various backgrounds

iii. Segmentation accuracy is less in hand tracking

ADVANTAGES

- Segmentation accuracy is high
- Easy to detect the finger postures
- Track fingers and sign recognition with less computational steps
  No need for additional hardware system

11. CONCLUSION

The ability to look, listen, talk, and respond appropriately to events is one of the most valuable gifts a human being can have. However, some unfortunate people are denied this opportunity. People get to know one another through sharing their ideas, thoughts, and experiences with others around them. There are several ways to accomplish this, the best of which is the gift of "Speech." Everyone can very persuasively transfer their thoughts and comprehend each other through speech. Our initiative intends to close the gap by including a lowcost computer into the communication chain, allowing sign language to be captured, recognised, and translated into speech for the benefit of blind individuals. An image processing technique is employed in this paper to recognise the handmade movements. This application is used to present a modern integrated planned system for hear impaired people. The camera-based zone of interest can aid in the user's data collection. Each action will be significant in its own right.

12. FUTURE SCOPE

Despite it having average accuracy, our system is still well-matched with the existing systems, given that it can perform recognition at the given accuracy with larger vocabularies and without an aid such as gloves or hand markings. In future, we can extend the framework to implement various deep learning algorithms to recognize the signs and implement in real time applications.

13. APPENDIX

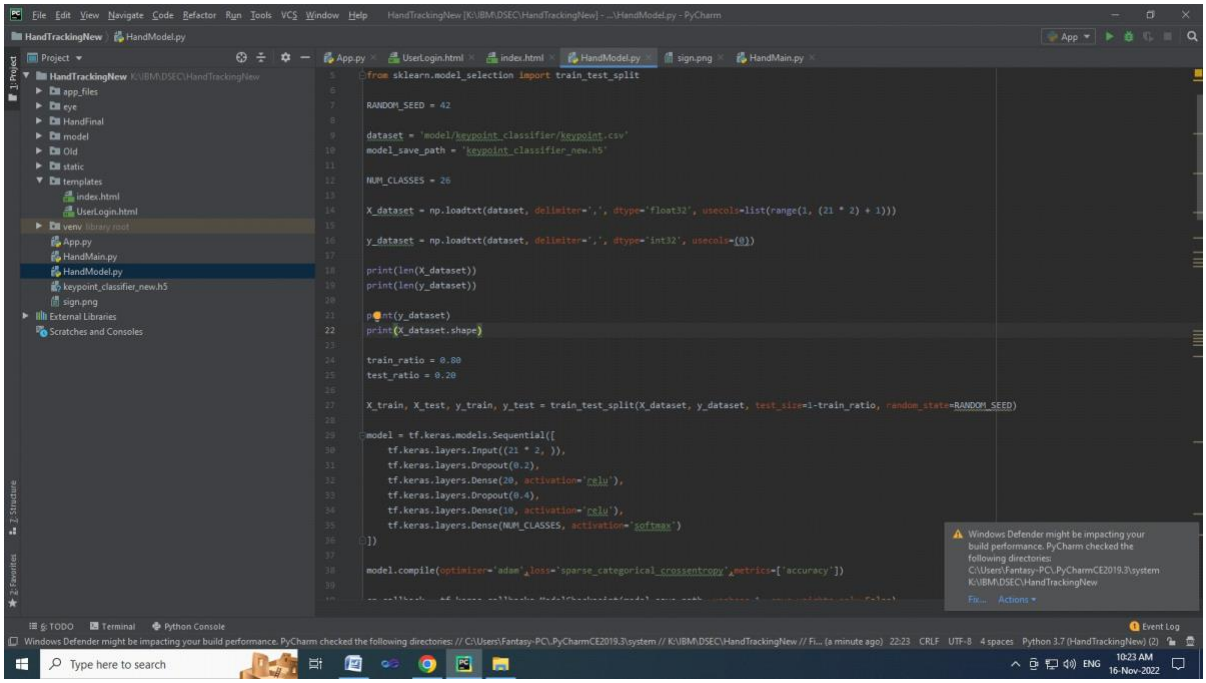
SOURCE CODE

```
from flask import Flask, render_template, flash, request, session from flask import render_template,
redirect, url_for, request


import smtplib app = Flask(__name__) app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
app.config['DEBUG']


@app.route("/") def homepage():
   return render_template('index.html')
@app.route("/UserLogin") def UserLogin():
   return render_template('UserLogin.html')


@app.route("/start", methods=['GET', 'POST']) def start():
   error = None if request.method == 'POST':
      import csv import copy import cv2 as
cv import mediapipe as mp from model
import KeyPointClassifier from app_files
import calc_landmark_list, draw_info_text,
draw_landmarks, get_args, pre_process_landmark
from PIL import Image, ImageDraw, ImageFont
import numpy as np


   args = get_args() cap_device = args.device
cap_width = args.width cap_height = args.height
```

```
    use_static_image_mode = args.use_static_image_mode      min_detection_confidence =
args.min_detection_confidence      min_tracking_confidence = args.min_tracking_confidence

    cap = cv.VideoCapture(cap_device)      cap.set(cv.CAP_PROP_FRAME_WIDTH, cap_width)
cap.set(cv.CAP_PROP_FRAME_HEIGHT, cap_height)

    mp_hands = mp.solutions.hands      hands = mp_hands.Hands(
static_image_mode=use_static_image_mode,          max_num_hands=1,
min_detection_confidence=min_detection_confidence,
min_tracking_confidence=min_tracking_confidence,
    )

    keypoint_classifier = KeyPointClassifier()

    with
open('model/keypoint_classifier/keypoint_classifier_label.csv',  encoding='utf-8-sig') as f:
        keypoint_classifier_labels = csv.reader(f)          keypoint_classifier_labels = [
row[0] for row in keypoint_classifier_labels
        ]

    flag = 0

    import win32com.client as wincl      speak =
wincl.Dispatch("SAPI.SpVoice")

    while True:
        key = cv.waitKey(10)          if key == 27:  # ESC
break

        ret, image = cap.read()          if not ret:              break          image =
cv.flip(image, 1)          debug_image = copy.deepcopy(image)
        # print(debug_image.shape)
        # cv.imshow("debug_image",debug_image)
        image = cv.cvtColor(image, cv.COLOR_BGR2RGB)
```

```
        image.flags.writeable = False            results = hands.process(image)
image.flags.writeable = True

        if results.multi_hand_landmarks is not None:
           for hand_landmarks, handedness in zip(results.multi_hand_landmarks, results.multi_handedness):
              landmark_list = calc_landmark_list(debug_image, hand_landmarks)

              # print(hand_landmarks)
              pre_processed_landmark_list = pre_process_landmark(landmark_list)

              hand_sign_id =
keypoint_classifier(pre_processed_landmark_list)

              debug_image = draw_landmarks(debug_image, landmark_list)
              flag += 1                print(flag)                if
(flag == 100):
                 flag = 0

                 speak.Speak(keypoint_classifier_labels[hand_sign_id])

              debug_image = draw_info_text(
                 debug_image,                    handedness,
keypoint_classifier_labels[hand_sign_id])

        cv.imshow('Hand Gesture Recognition', debug_image)

     cap.release()      cv.destroyAllWindows            return
render_template('UserLogin.html')

if __name__ == '__main__':
app.run(debug=True, use_reloader=True)

import mediapipe as mp import cv2 import numpy
as np import uuid import os

'''import subprocess as sp programName = "notepad.exe"
```

```
#fileName = "sms.txt"
#sp.Popen([programName, fileName]) sp.Popen([programName])'''
mp_drawing = mp.solutions.drawing_utils mp_hands = mp.solutions.hands
cap = cv2.VideoCapture(0)

 with mp_hands.Hands(min_detection_confidence=0.8, min_tracking_confidence=0.5) as
hands:     while cap.isOpened():         ret, frame = cap.read()

      # BGR 2 RGB        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

      # Flip on horizontal        image = cv2.flip(image, 1)

      # Set flag        image.flags.writeable = False

      # Detections        results = hands.process(image)

      # Set flag to true        image.flags.writeable = True

      # RGB 2 BGR        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
     # print(results)

      # Rendering results        if results.multi_hand_landmarks:            for num, hand in
enumerate(results.multi_hand_landmarks):                mp_drawing.draw_landmarks(image, hand,
mp_hands.HAND_CONNECTIONS,
                                mp_drawing.DrawingSpec(color=(14, 22,
76), thickness=2, circle_radius=4),                                mp_drawing.DrawingSpec(color=(24, 44, 250),
thickness=2, circle_radius=2),
                                )

      cv2.imshow('Hand Tracking', image)

      if cv2.waitKey(10) & 0xFF == ord('q'):            break

cap.release() cv2.destroyAllWindows()
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
HandTrackingNew [K:\IBM\DSEC\HandTrackingNew] - ...\HandModel.py - PyCharm
HandTrackingNew | HandModel.py
App.py | UserLogin.html | index.html | HandModel.py | sign.png | HandMain.py
Project
HandTrackingNew K:\IBM\DSEC\HandTrackingNew
app_files
eye
HandFinal
model
Old
static
templates
index.html
UserLogin.html
venv library root
App.py
HandMain.py
HandModel.py
keypoint_classifier_new.h5
sign.png
External Libraries
Scratches and Consoles
from sklearn.model_selection import train_test_split
RANDOM_SEED = 42
dataset = 'model/keypoint_classifier/keypoint.csv'
model_save_path = 'keypoint_classifier_new.h5'
NUM_CLASSES = 26
X_dataset = np.loadtxt(dataset, delimiter=',', dtype='float32', usecols=list(range(1, (21 * 2) + 1)))
y_dataset = np.loadtxt(dataset, delimiter=',', dtype='int32', usecols=(0))
print(len(X_dataset))
print(len(y_dataset))
print(y_dataset)
print(X_dataset.shape)
train_ratio = 0.80
test_ratio = 0.20
X_train, X_test, y_train, y_test = train_test_split(X_dataset, y_dataset, test_size=1-train_ratio, random_state=RANDOM_SEED)
model = tf.keras.models.Sequential([
tf.keras.layers.Input((21 * 2,)),
tf.keras.layers.Dropout(0.2),
tf.keras.layers.Dense(20, activation='relu'),
tf.keras.layers.Dropout(0.4),
tf.keras.layers.Dense(10, activation='relu'),
tf.keras.layers.Dense(NUM_CLASSES, activation='softmax')
])
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
Windows Defender might be impacting your build performance. PyCharm checked the following directories:
C:\Users\Fantasy-PC\.PyCharmCE2019.3\system
K:\IBM\DSEC\HandTrackingNew
Fix... Actions
TODO Terminal Python Console
Event Log
Windows Defender might be impacting your build performance. PyCharm checked the following directories: // C:\Users\Fantasy-PC\.PyCharmCE2019.3\system // K:\IBM\DSEC\HandTrackingNew // Fi... (a minute ago) 22:23 CRLF UTF-8 4 spaces Python 3.7 (HandTrackingNew) (2)
Type here to search
ENG
10:23 AM
16-Nov-2022

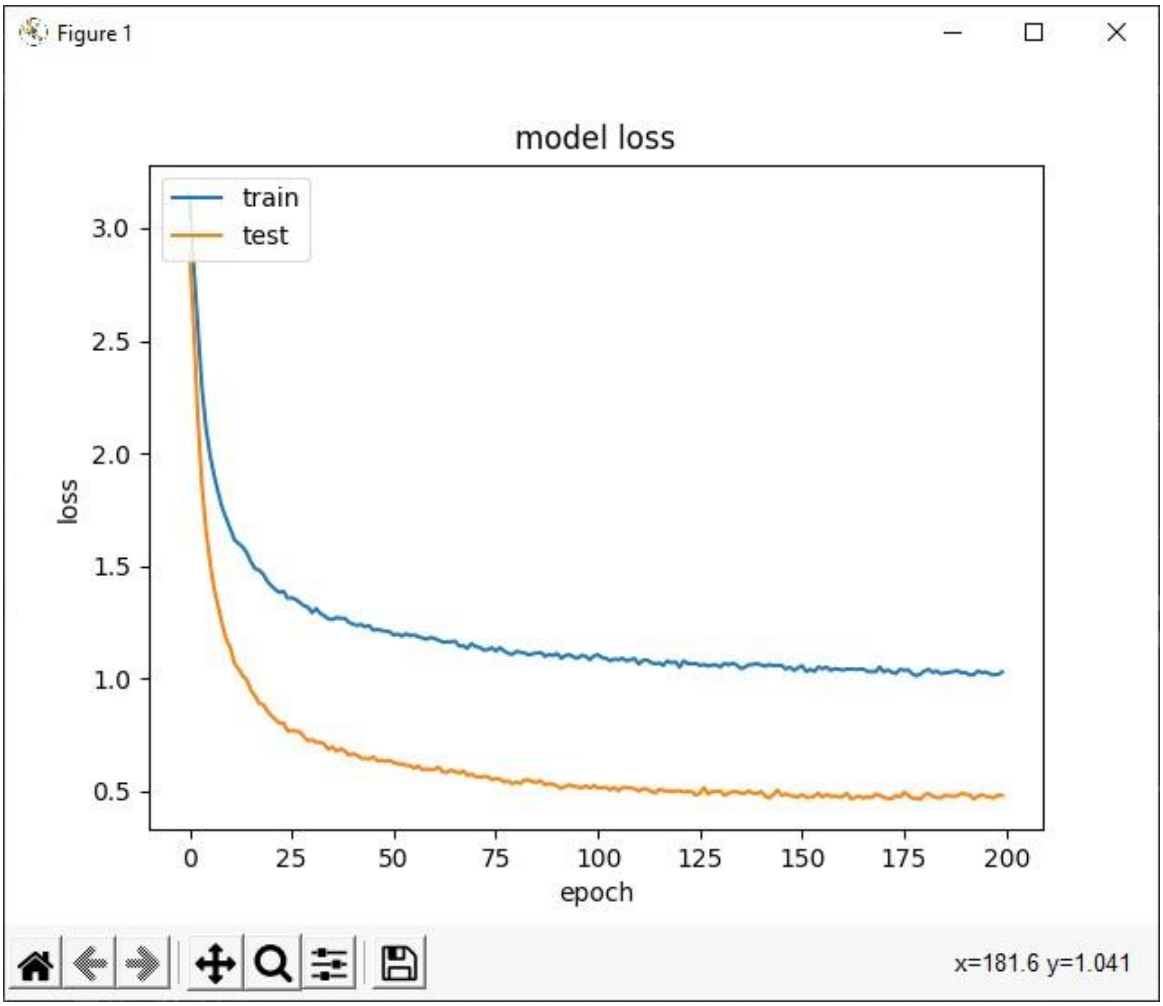
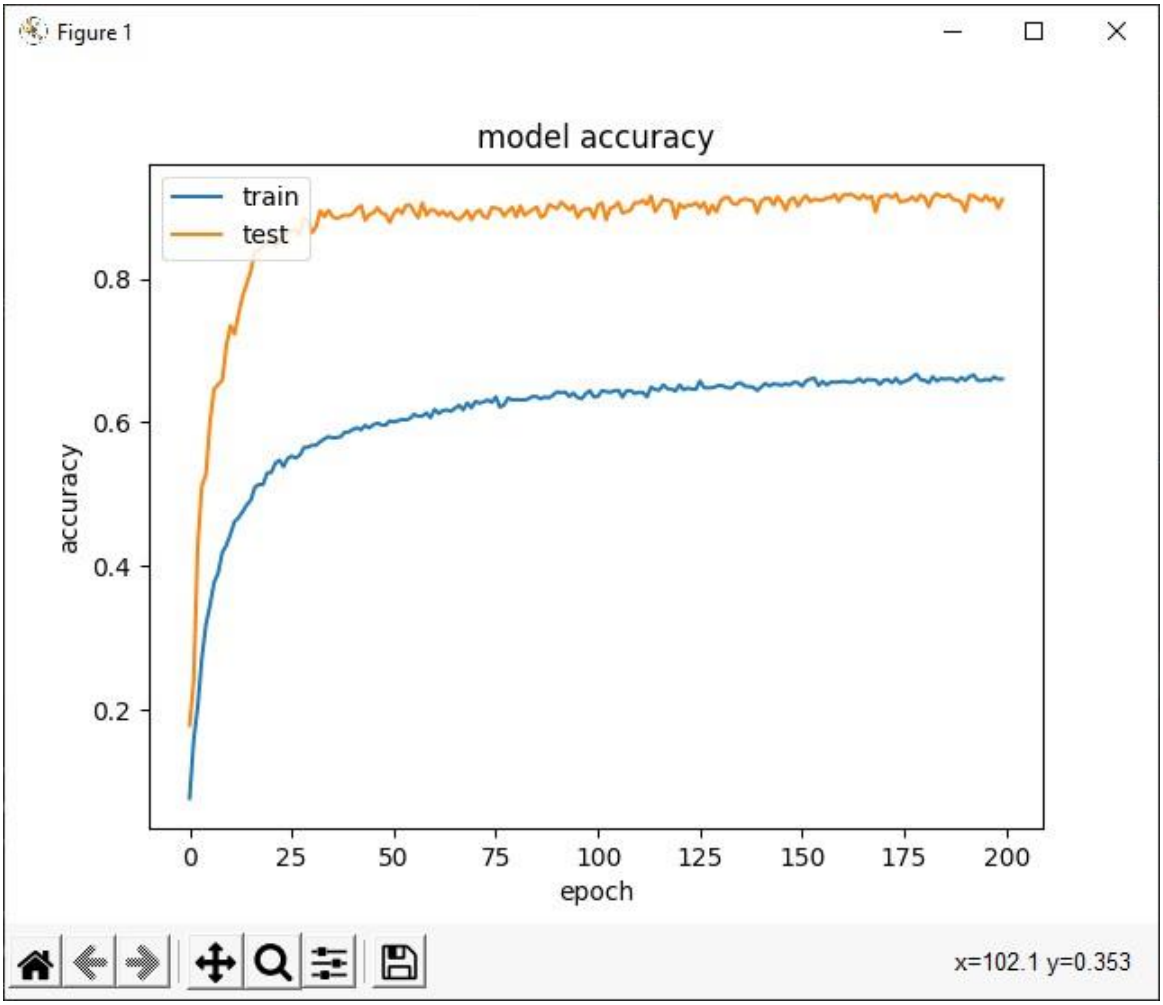


Figure 1
model loss
train
test
loss
epoch
3.0
2.5
2.0
1.5
1.0
0.5
0
25
50
75
100
125
150
175
200
x=181.6 y=1.041



GITHUB ACCOUNT LINK:

https://github.com/IBM-EPBL/IBM-Project-31230-1660197974

PROJECT DEMO LINK:

https://drive.google.com/file/d/116vqvuPqQaR70KjE_GJqUNQ9QILbjWVV/view?usp=share_link