# 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**

1. **INTRODUCTION**
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING**
   1. Feature 1
   2. Feature 2
8. **TESTING**
   1. Test Cases
   2. User Acceptance Testing
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

# 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.

# INTRODUCTION

* 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 the human 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.

# 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. |

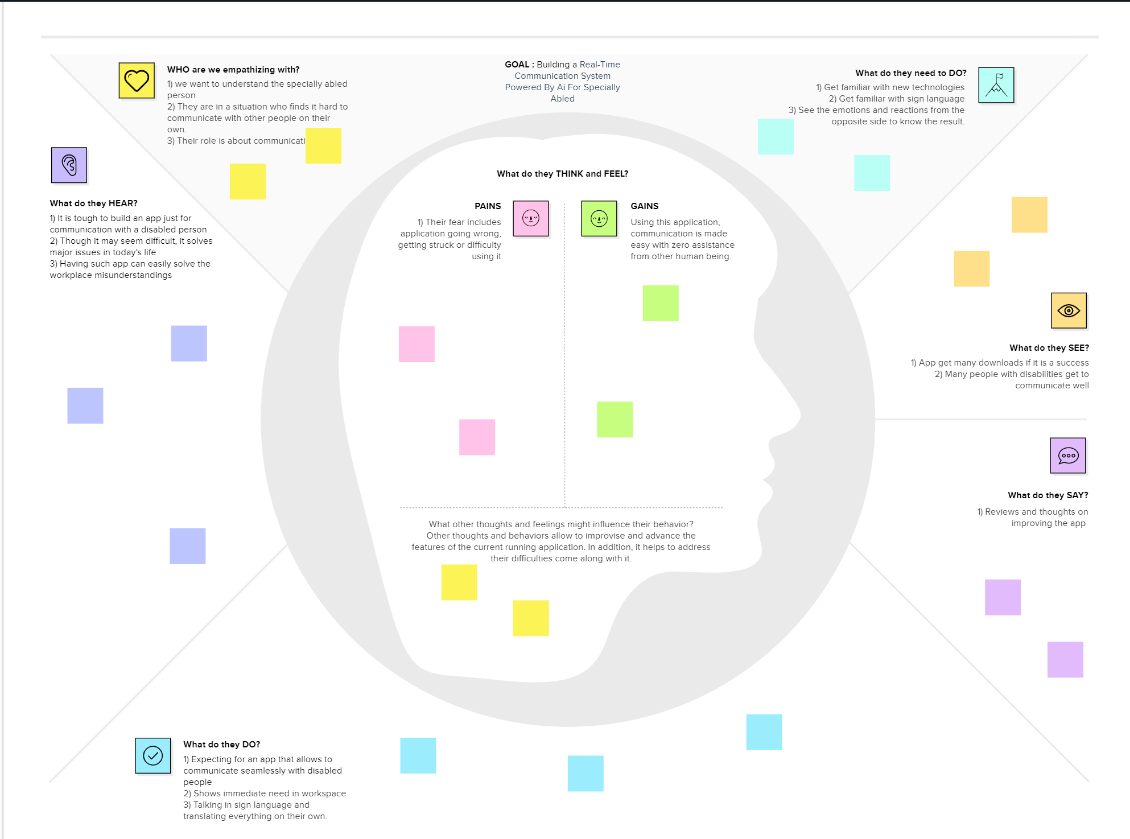
**2.1 EXISTING**

* 1. **PRO PROBLEM STATEMENT DEFINITION**

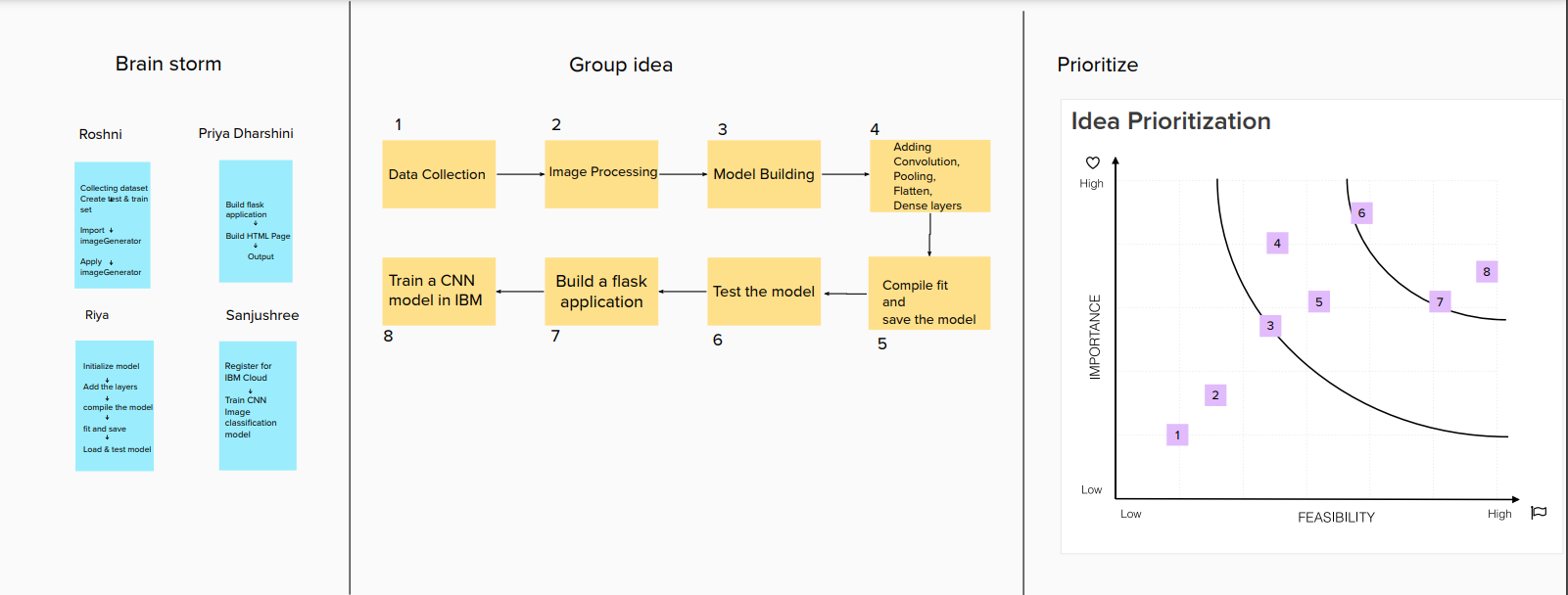
The sign language is used widely by people who are deaf-dumb these are used as a medium for communication. A sign language is nothing but composed of various gestures formed by different shapes of hand, its movements, orientations as well as the facial expressions. There are around 466 million people 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

# IDEATION & PROPOSED SOLUTION

* 1. **EMPATHY MAP CANVAS**

****

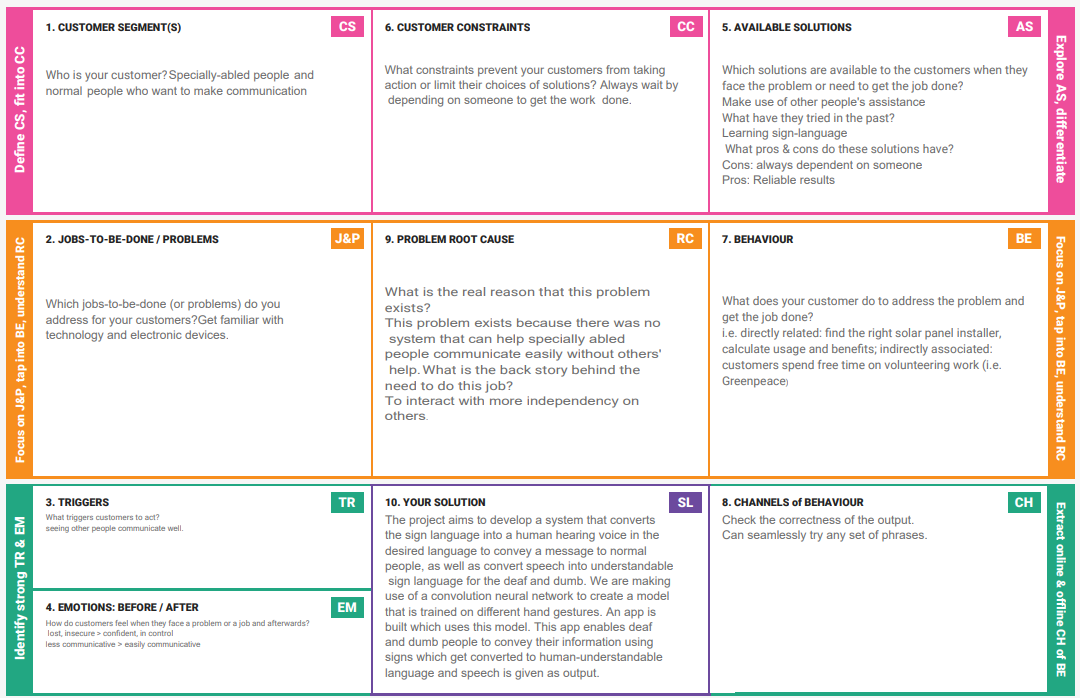
* 1. **IDEATION & BRAINSTORMING**

****

* 1. **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 have a 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 and an app is built for the use this mode.  • AI has been deployed/used in an efficient manner in the development of this idea. |
| **4** | Social Impact / Customer  Satisfaction | Communicating with others and being connected in the society and remove 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 with more 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 many top MNC’s around the world. |
| **5** | Problem Statement (Problem to be solved) | * + Useful in organization where communication between co-workers is much important.   + Can make collaborations with government.   + Example: Tata Group made a collaboration with Indian government in educating people through Cutting- Edge, which is a great success. |

**3.4 Problem Fit Solution**



# 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:**

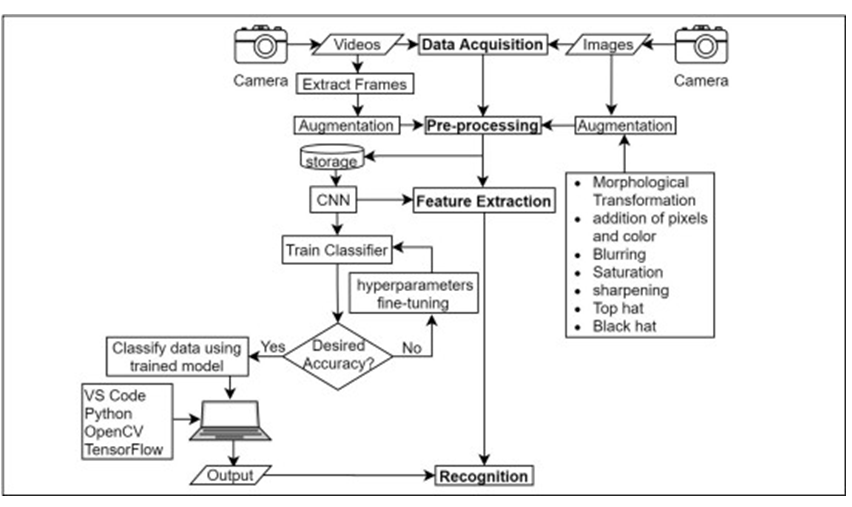
Following are the non-functional requirements of the proposed solution.

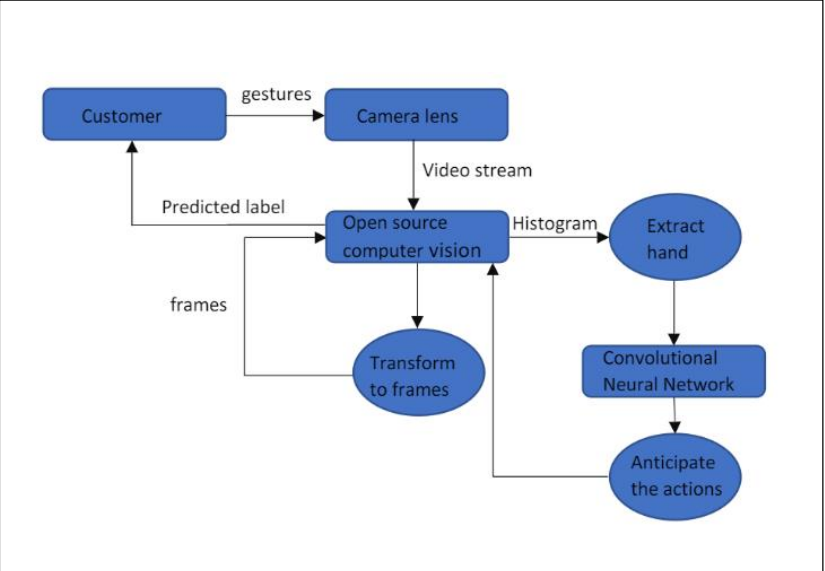
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | This system is easy to use.  It helps in translating sign language into our desired language. |
| NFR-2 | **Security** | Only the user can sign in to the system and unauthorised 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. |

# PROJECT DESIGN

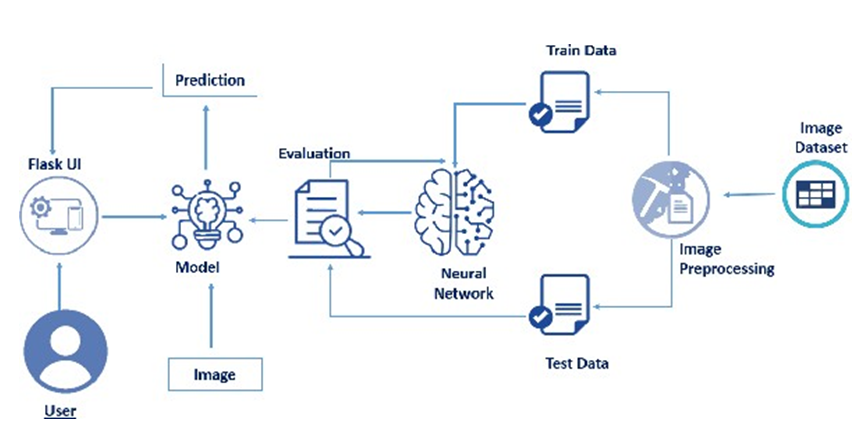
* 1. **DATA FLOW DIAGRAMS**

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.

****

****

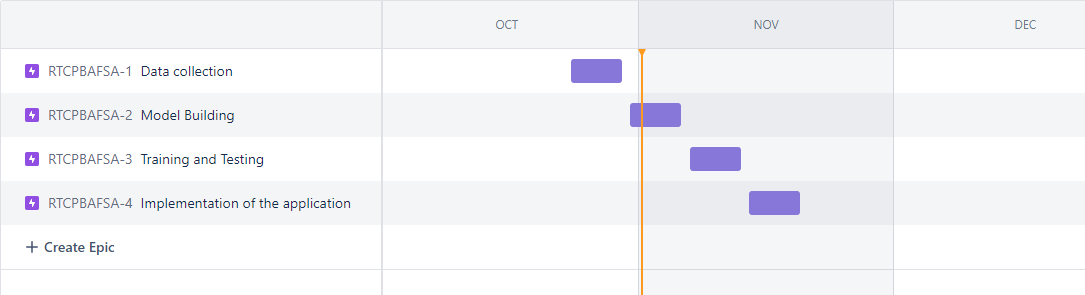
* 1. **SOLUTION & TECHNICAL ARCHITECTURE**



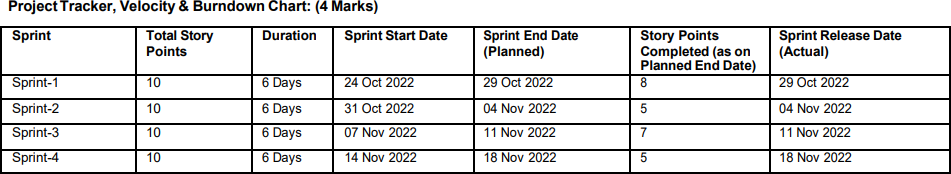
Software architecture involves the high level structure of software system abstraction, by using decomposition and composition, with architectural style and quality attributes. A software architecture design must conform to the major functionality and performance requirements of the system, as well as 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

* 1. **SPRINT PLANNING & ESTIMATION**



* 1. **SPRINT DELIVERY SCHEDULE**

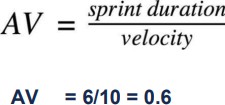


* 1. **REPORTS FROM JIRA**

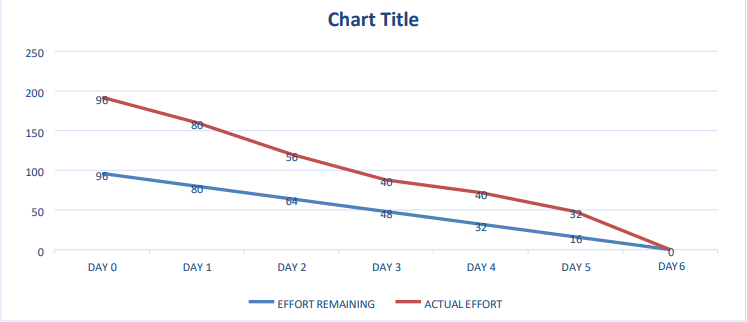
JIRA has categorized reports in four levels, which are

* Agile
* Issue Analysis
* Forecast & Management
* Others

**Velocity**:



**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**)

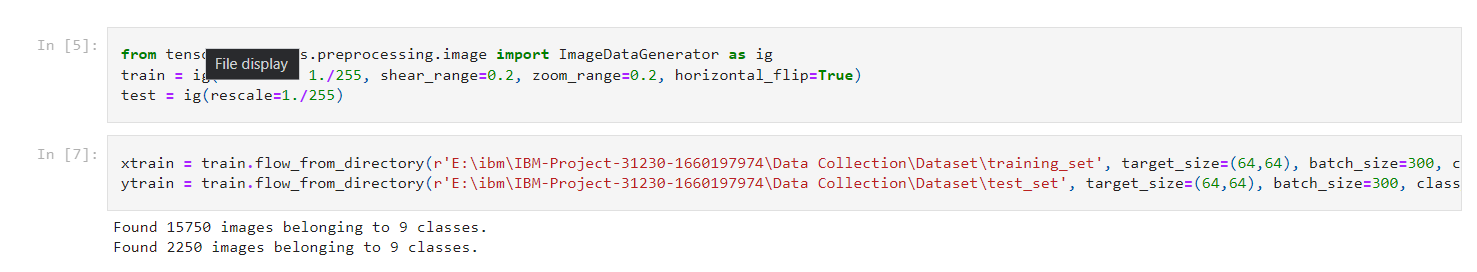
test **=** ig(rescale**=**1.**/**255)

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.



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]:

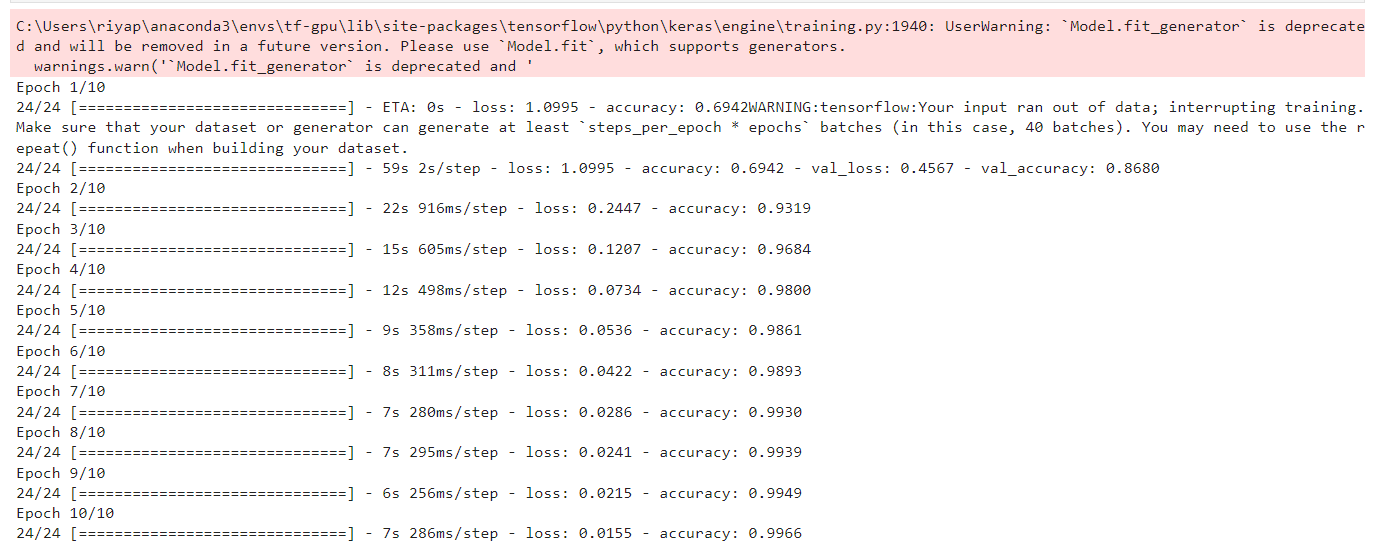
*#compile the model*

model**.**compile(loss**=**'categorical\_crossentropy', optimizer**=**'adam', metrics**=**['accuracy'])

In [20]:

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*



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)

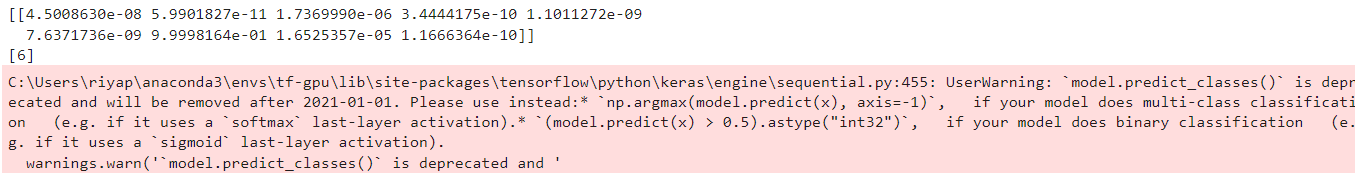
pred **=** model**.**predict\_classes(img)

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)



# 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:

* + 1. Accurate: Exacts the purpose.
    2. Economical: No unnecessary steps or words.
    3. Traceable: Capable of being traced to requirements.
    4. Repeatable: Can be used to perform the test over and over.
    5. Reusable: Can be reused if necessary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N O** | **FUNCTION** | **DESCRIPTIO N** | **EXPECTE D OUTPUT** | **ACTUA L OUTPU**  **T** | **STATU S** |
| 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 | Comment s in text  format | Success |
| 3 | Classification | Classify the  Datasets | Finger  Gestures | Finger  Gestures | Success |
| 4 | Rules  implementatio n | Block the  comments and friends | Block the users | Block the users | Success |

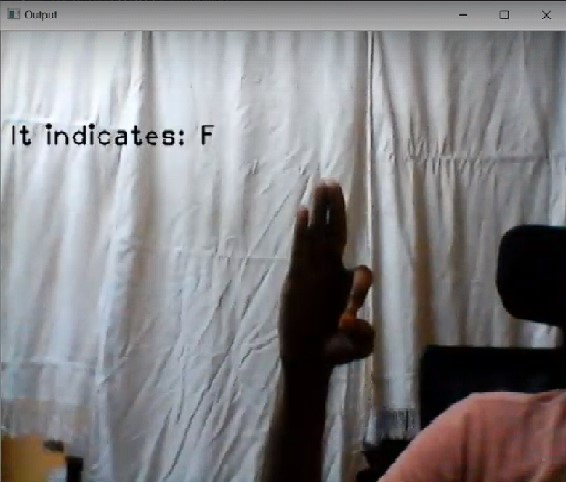
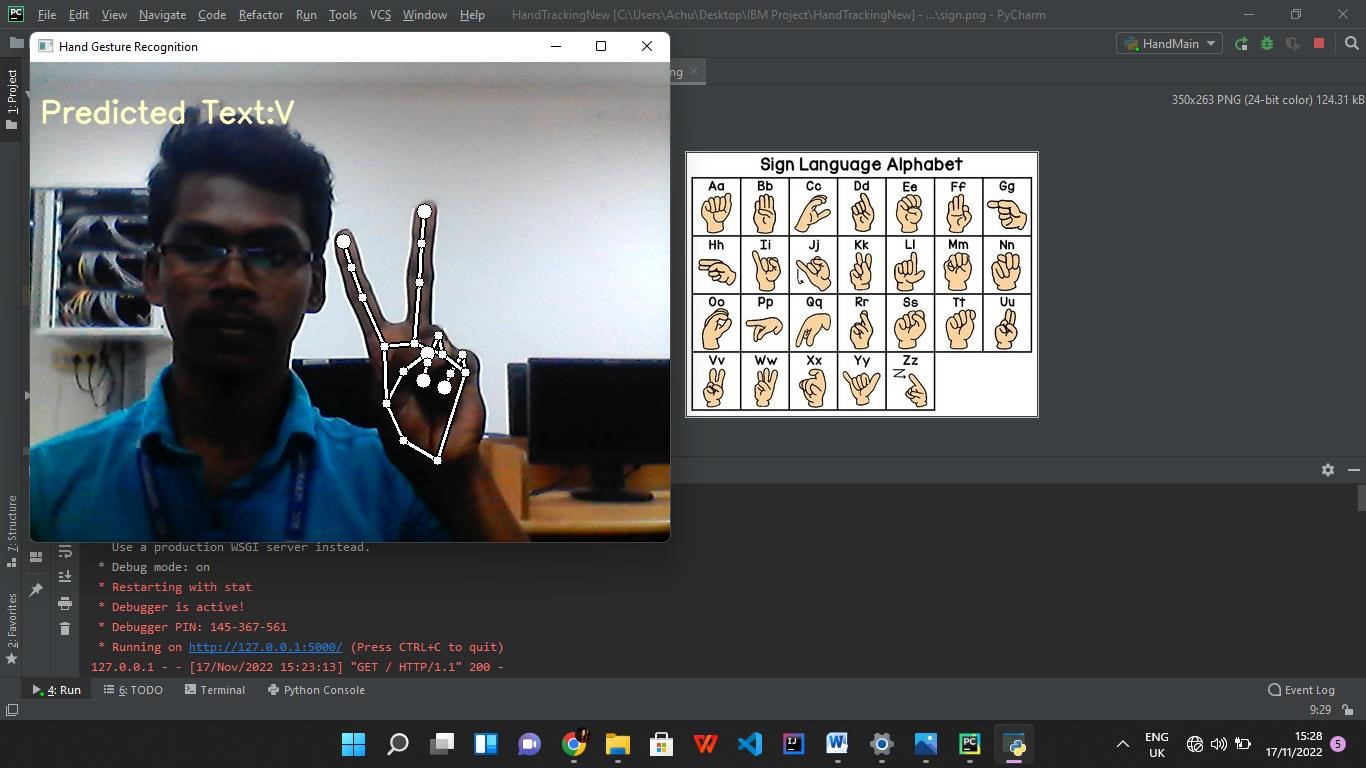
* 1. **USER ACCEPTANCE TESTING**

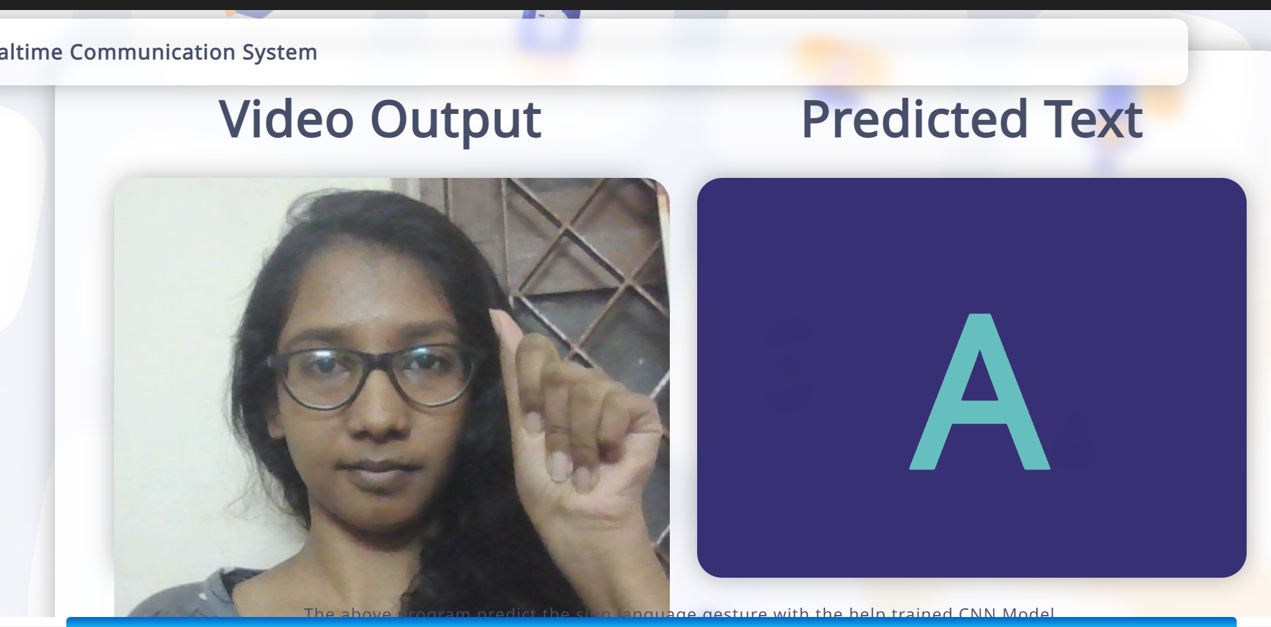
Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs 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.

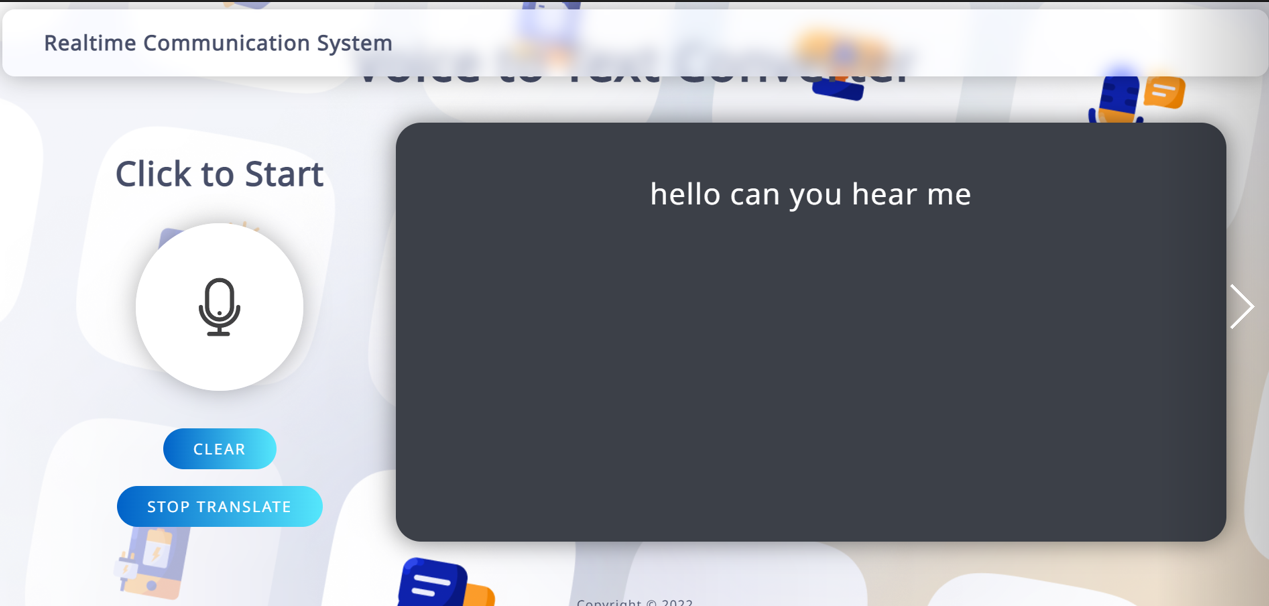


# RESULTS

**9.1 PERFORMANCE METRICS**







# ADVANTAGES & DISADVANTAGES

**DISADVANTAGES**

* + 1. Need hardware control to detect the hands
    2. Hand segmentation become complex of various backgrounds
    3. 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

# 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,**

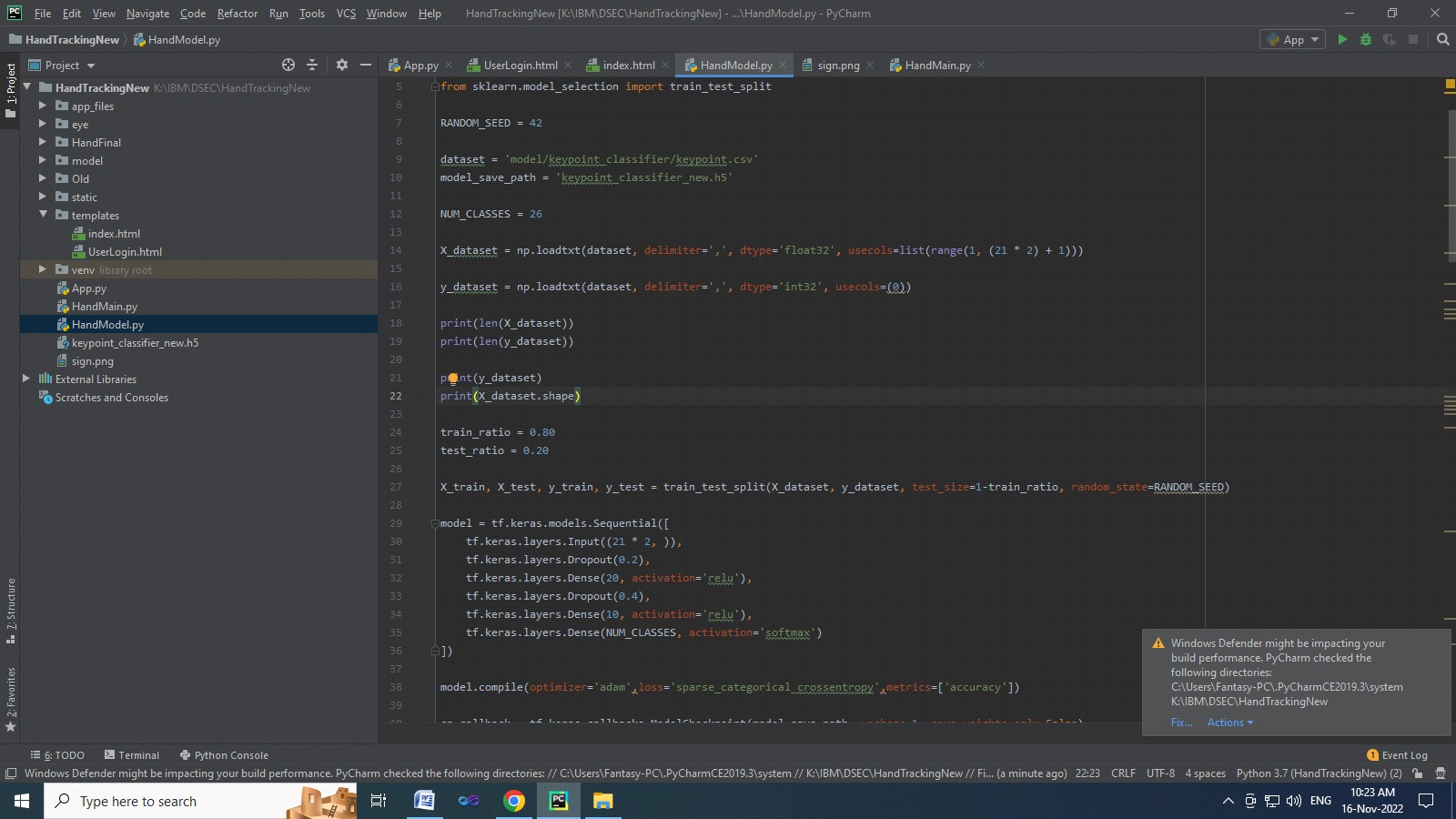
**76), thickness=2, circle\_radius=4), mp\_drawing.DrawingSpec(color=(24, 44, 250), thickness=2, circle\_radius=2),**

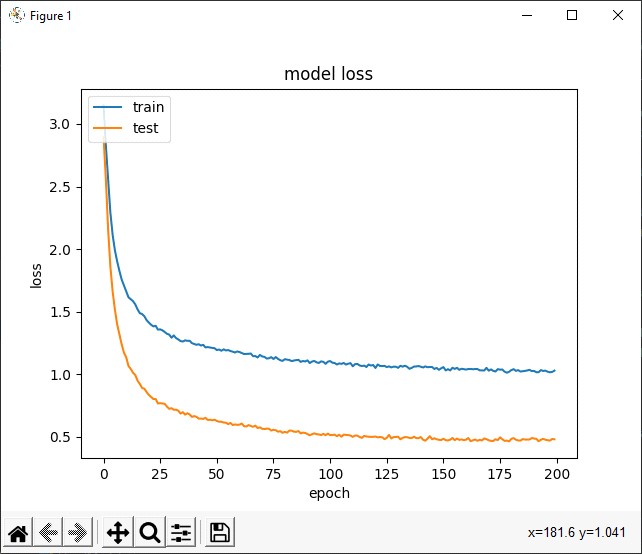
**)**

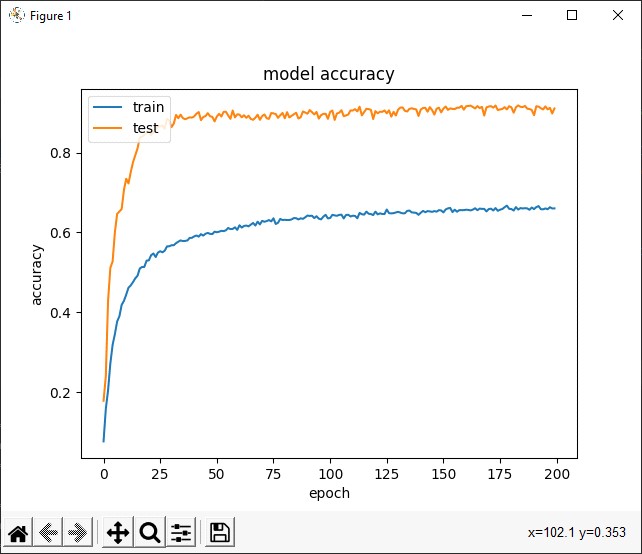
**cv2.imshow('Hand Tracking', image)**

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

**cap.release() cv2.destroyAllWindows()**







**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**