**Project Report**

**1. INTRODUCTION**

**1.1 Project Overview**

In our society, we have people with disabilities. Technology is developing day by day, but no significant developments are being undertaken for the betterment of these people. Communication between deaf-mute and a normal person has always been a challenging task. Since normal people are not trained in 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. A 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.

**1.2 Purpose**

The main purpose of this project is to make deaf and dumb people to communicate with normal people.

In emergency situations they can easily communicate with each other.

**2. LITERATURE SURVEY**

* **Sign language recognition using CNN** - This research paper is focused on the recognition of American Sign Language using Convolutional Neural Network.
* **Design of a Communication System using Sign Language aid for Differently Abled Peoples** - This research paper is focused on the design of a two-way communication system between the deaf and dumb and normal people.
* **SPEECH TO ISL (INDIAN SIGN LANGUAGE) TRANSLATOR -** This research paper is focused on converting speech into Indian Sign Language. The noise removal process is used in this project to improve output accuracy.
* **Sign Language Recognition Techniques: A survey -** This research paper describes the various sign language recognition techniques in brief.
* **SIGN LANGUAGE RECOGNITION SYSTEM -** This research paper overcomes the disadvantages of glove-based hand-sign recognition systems by detecting hand gestures with Raspberry Pi and a camera.

**2.1 Existing System**

The existing system recognizes gestures using flex sensors, and the Arduino is at the heart of the setup. The output will change as the flex sensor bends. Each sign will have a corresponding output in this system. Output will be displayed on the LCD as text.

Disadvantages

* The level of accuracy is low.
* Flex sensors will easily get damaged.

**2.1 References**

[1] Dr. Thamaraiselvi, Challa Sai Hemanth, j Hruday Vikas, “SIGN LANGUAGE RECOGNITION USING CNN”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 09, Issue 03, pp. 819-824, Mar 2022.

[2] Shrikant Temburwar, Payal Jaiswal, Shital Mande, Souparnika Patil, “Design of a Communication System using Sign Language aid for Differently Abled Peoples”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 04, Issue 03, pp. 1207-1209, Mar 2017.

[3] Kajal Jadhav, Shubham Gangdhar, Viraj Ghanekar, “SPEECH TO ISL (INDIAN SIGN LANGUAGE) TRANSLATOR”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 08, Issue 04, pp. 3696-3698, Apr 2021.

[4] Omkar Govalkar, Pratik Gaikar, Pramod Gavali, “Sign Language Recognition Techniques: A survey”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 07, Issue 12, pp. 621-624, Dec 2020.

[5] M.HEMANTH, K.EDWARD IRUDAYA RAJ, M.ABUBAKKER SITHIK, M.JENITH RUBAN, G.MADHUSUDANAN, “SIGN LANGUAGE RECOGNITION SYSTEM”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 07, Issue 03, pp. 4198-4200, Mar 2020.

[6] Sanket Bankar, Tushar Kadam, Vedant Korhale, Mrs. A. A. Kulkarni, “Real Time Sign Language Recognition Using Deep Learning”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 09, Issue 04, pp. 955-959, Apr 2022.

[7] Shubham Sawant, Ronak Sahay, Prathamesh Salunkhe, Prof. Roshan bauskar, “Real-Time Recognition of Sign Language Using Machine Learning”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 08, Issue 05, pp. 401-403, May 2021.

[8] Mr. R. Augustian Issac, S. Sri Gayathri, “Sign Language Interpreter”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 05, Issue 10, pp. 248-251, Oct 2018.

[9] Dr.M.P. Chitra, Vaishnavi Devi. R, Shalini M, Sriee Sathana. L.B, “Sign Language Recognition For Deaf and Mute”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 08, Issue 04, pp. 570-574, Apr 2021.

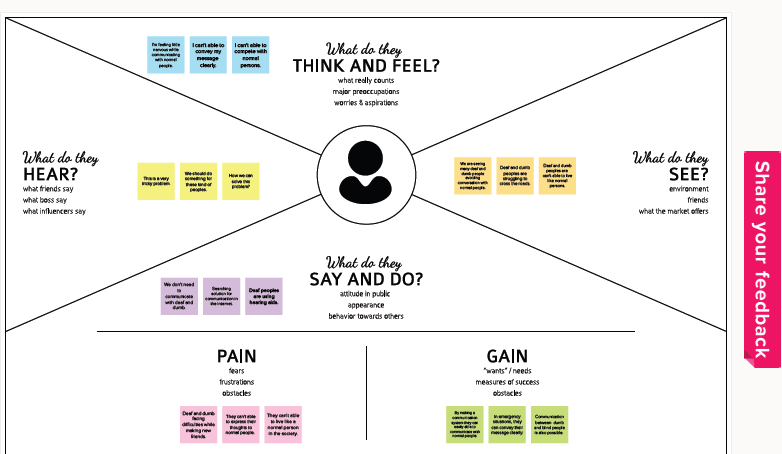
[10] Shailesh bachani, Shubham dixit, Rohin chadha, Prof. Avinash Bagul, “SIGN LANGUAGE RECOGNITION USING NEURAL NETWORK”, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 07, Issue 04, pp. 583-586, Apr 2020.

**2.3 Problem Statement**

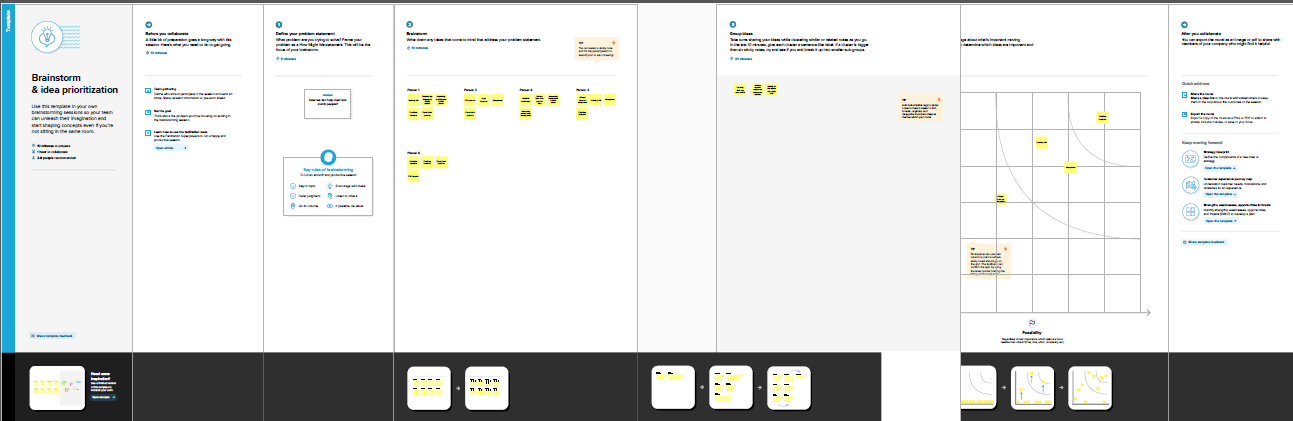
Communication is the only medium by which we can share our thoughts or convey the message but for a person with disability (deaf and dumb) faces difficulty in communication with normal person. Generally dumb people use sign language for communication but they find difficulty in communicating with others who don’t understand sign language. So there is a barrier in communication between these two communities.

**3. IDEATION & PROPOSED SOLUTION**

**3.1 Empathy Map Canvas**



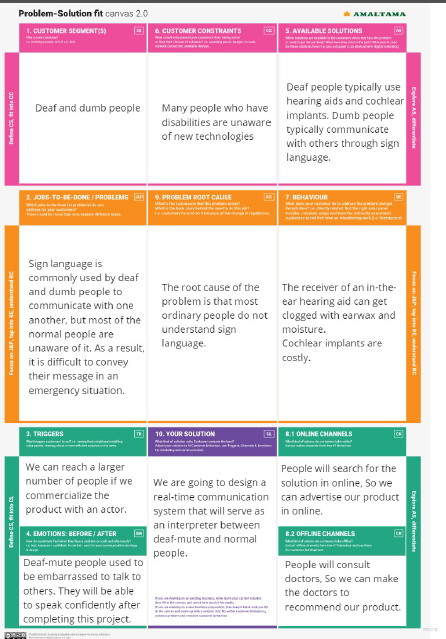
**3.2 Brainstorming**

****

**3.3 Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Parameter** | **Description** |
| 1. | Problem Statement | Generally deaf and dumb people use sign language to communicate with each other. But normal people don’t understand sign language. In emergency situations, deaf and dumb people cannot able to convey their message to normal people. So, there is a barrier between these two communities. |
| 2. | Idea / Solution description | We are going to design a real-time communication system using AI that is going to act as an interpreter between deaf-mute and normal people. |
| 3. | Novelty / Uniqueness | The existing system recognizes gestures using flex sensors, and the Arduino is at the heart of the setup. The output will change as the flex sensor bends. Each sign will have a corresponding output in this system. Output will be displayed on the LCD as text. There are lot of disadvantages in this system.  So, we propose a computer-vision system for sign language recognition. Our proposed system is not dependent on the use of gloves and microcontrollers like Arduino and Raspberry Pi. We are going to design a communication system by making use of Convolutional Neural Network. The hand gestures will be captured by the camera, and our pre-trained model will make predictions based on the input. Finally, the output will be delivered in the form of voice or text. |
| 4. | Social Impact / Customer Satisfaction | By using this application deaf and dumb people can confidently communicate with normal people without any interruption. The barrier between deaf-mute and normal people will be disappeared. |
| 5. | Business Model | We can build this project as a web application and fix a subscription cost. Subscription cost should be reasonable so that everyone can use this application. By this way, we can make revenue from this project. |
| 6. | Scalability of the Solution | We can build this web application using python flask and  newer AI technologies can be used to improve the functionality, features and performance of this application. |

**3.4 Problem Solution Fit**

****

**4. REQUIREMENT ANALYSIS**

**4.1 Functional Requirements**

|  |  |
| --- | --- |
| **FR NO.** | **Functional Requirement** |
| FR-1 | A camera should be used to capture the hand gestures. Deaf-mute people show hand gestures very fast during communication. So, the camera should be fast enough to capture the hand gestures. |
| FR-2 | Sign language should be converted into human understandable voice. So that normal people can understand what deaf-mute people are trying to say. |
| FR-3 | Voice should be converted into sign language. So that deaf-mute people can understand what normal trying to say. |
| FR-4 | We need to integrate the above functionalities and build this project as a web application. |

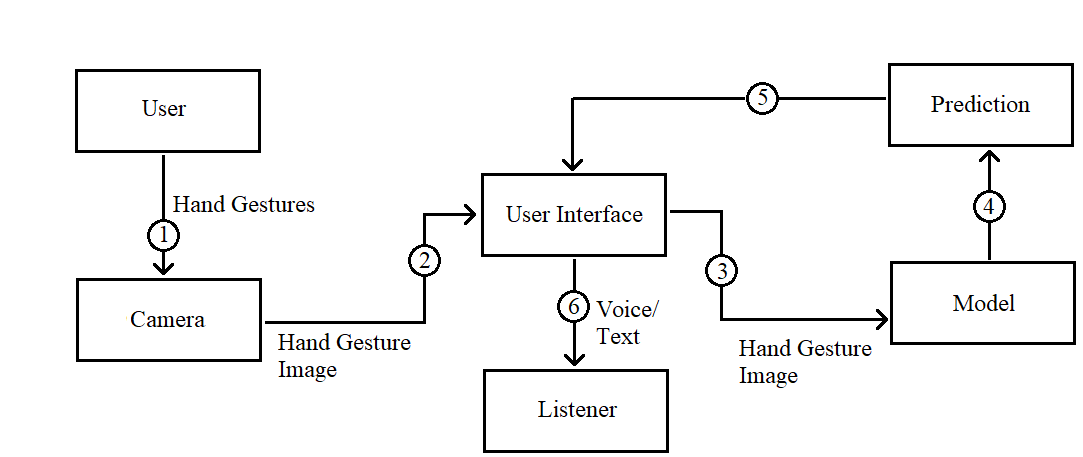
**4.2 Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **FR NO.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | Usability | We are going to use Flask UI in this application for better user experience. UI should be very simple. So that every people can easily use this application. |
| NFR-2 | Reliability | We are going to fix a subscription cost for this application. The subscription cost should be reasonable. So that every people can afford it. |
| NFR-3 | Performance | AI model needs some processing resources. So we need to have a decent hardware. The application should be well optimized. Load time and latency should be less. |
| NFR-4 | Availability | It is a web application. So it should be accessible to people anywhere anytime. |
| NFR-5 | Scalability | There are lot of deaf and mute people in our society. So definitely we can expect more number users for this application. We can also enhance this application for home automation by integrating this with Alexa smart home. |

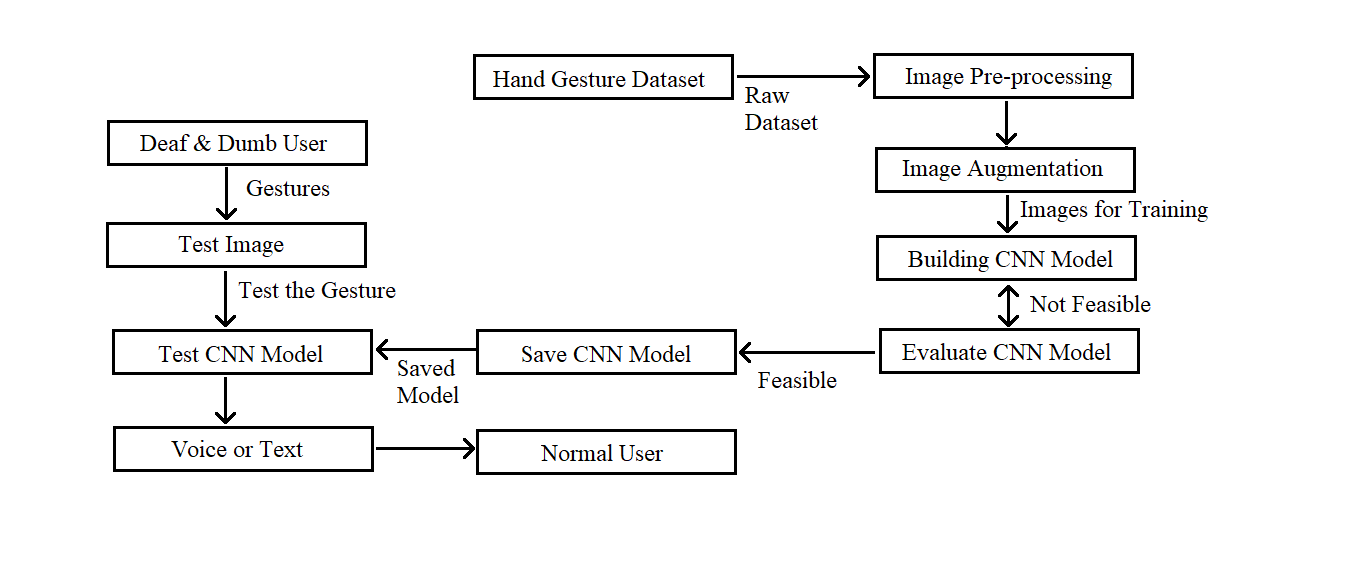
**5. PROJECT DESIGN**

**5.1 Data Flow Diagrams**

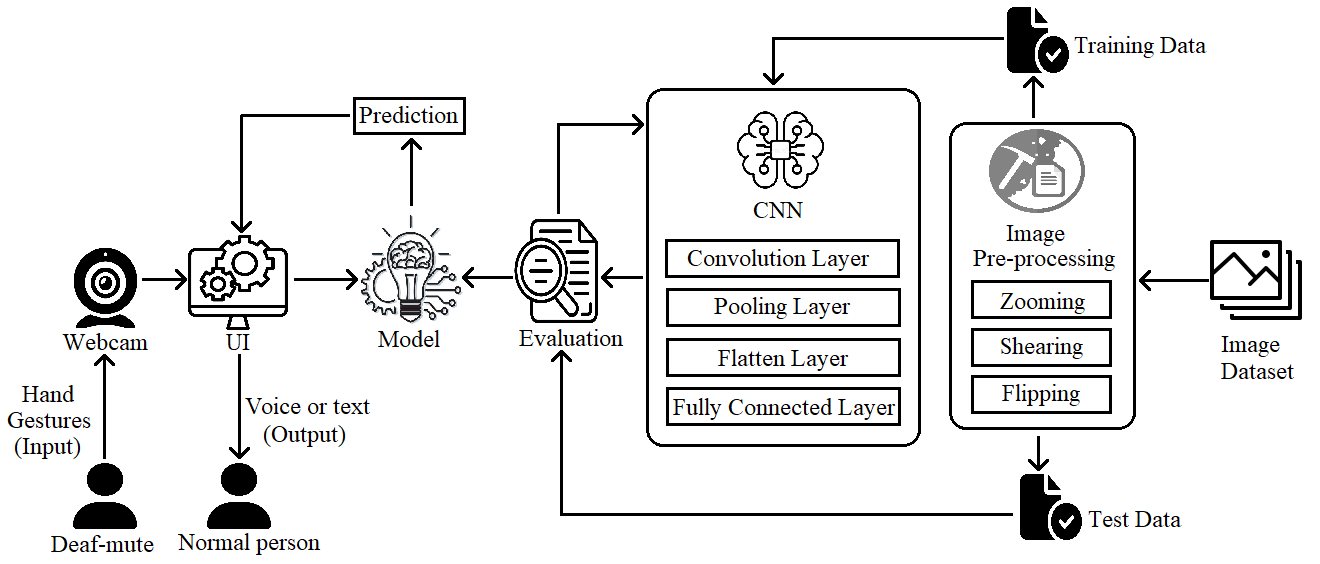
**Flow Chart:**

****

**Data Flow Diagram:**

****

**5.2 Solution and Technical Architecture**

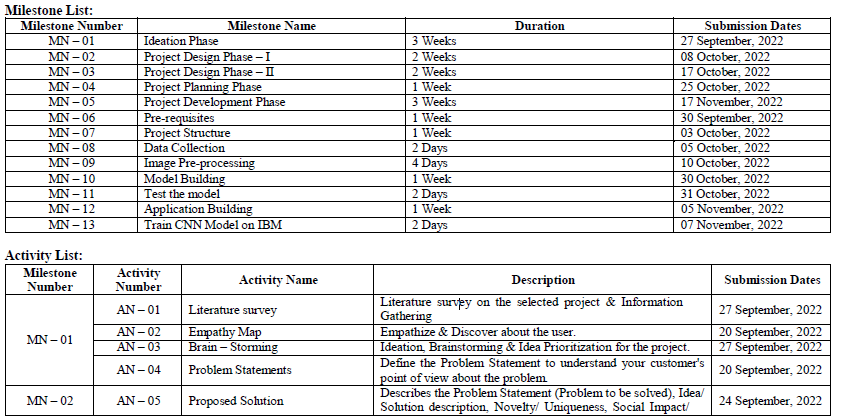


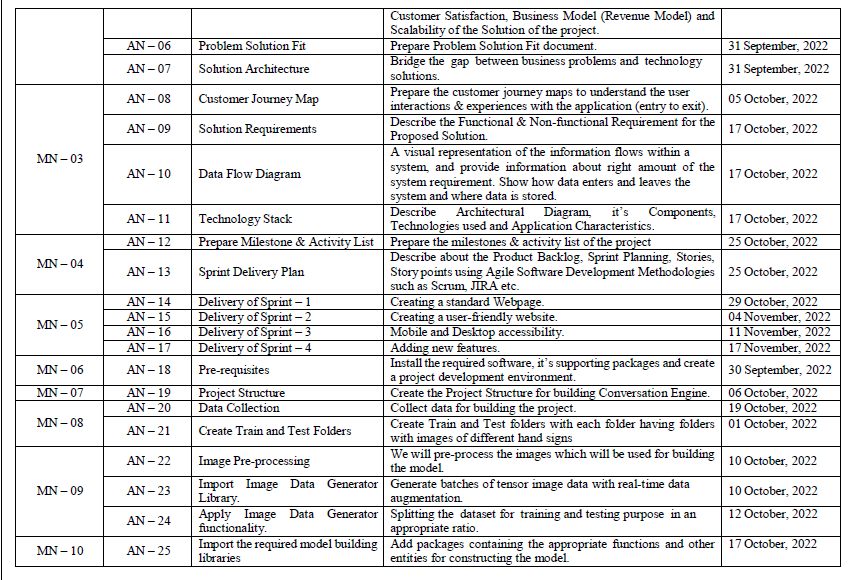
**5.3 User Stories:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement**  **(Epic)** | **User Story**  **Number** | **User story/**  **Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer  (Mobile user) | Registration | USN-1 | As a user, I can register for the application. | I can access my account. | Medium | Sprint-4 |
|  | Login | USN-2 | As a user, I can log in to the application. | I can successfully log in every time. | Medium | Sprint-4 |
| Customer  (Desktop  User) | Main page | USN-3 | As a user, I want use the application as website. | Website should be responsive and optimized. | High | Sprint-3 |
|  | Hand gesture detection | USN-4 | As a user, I want the application to detect sign language. | Application should detect all relevant hand gestures. | High | Sprint-1 |
|  | Sign language to text conversion | USN-5 | As a user, I want the application to convert sign language into text. | Application should convert quickly. | High | Sprint-2 |
|  | Sign language to voice conversion | USN-6 | As a user, I want the application to convert sign language into voice. | Voice output should be delivered accurately. | High | Sprint-2 |
|  | Customer Support | USN-7 | As a user, I want some immediate customer support. | Chat bot should answer all important queries. | Low | Sprint-4 |

**6. PROJECT PLANNING AND SCHEDULING**

**6.1 Sprint Planning & Estimation**

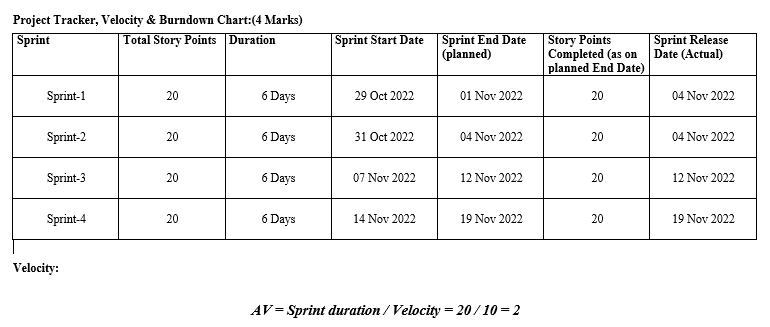
****

****

**6.2 Sprint Delivery Scheme**

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirment**  **(Epic)** | **User Story Number** | **User Story/Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | User Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password or through Gain, Facebook. | 20 | High | Daniel Durai J, Manikandan K |
| Sprint-2 | Data Input | USN-2 | As a user, I will be giving the input via Camera as sign language or via speech. | 20 | High | Rajesh T, Sam Willbert S |
| Sprint-3 | Data Verification | USN-3 | Once the user gives the data input via Camera it verifies the database. | 20 | High | Esakky Raja M,  Daniel Durai J |
| Sprint-4 | Final Delivery | USN-4 | Verifies with the data set and converts the input to text. | 20 | High | Manikandan K, Sam Willbert S |

****

**7. CODING & SOLUTIONING**

**7.1 Feature 1**

The AI model has been created using Convolutional Neural Networks. The model is trained on different hand gestures in American Sign Language. The model is able to detect 26 Alphabets in ASL.

**7.2 Feature 2**

The model is integrated with Flask UI. So that user can easily interact.

**8. TESTING**

**8.1 Test Cases**

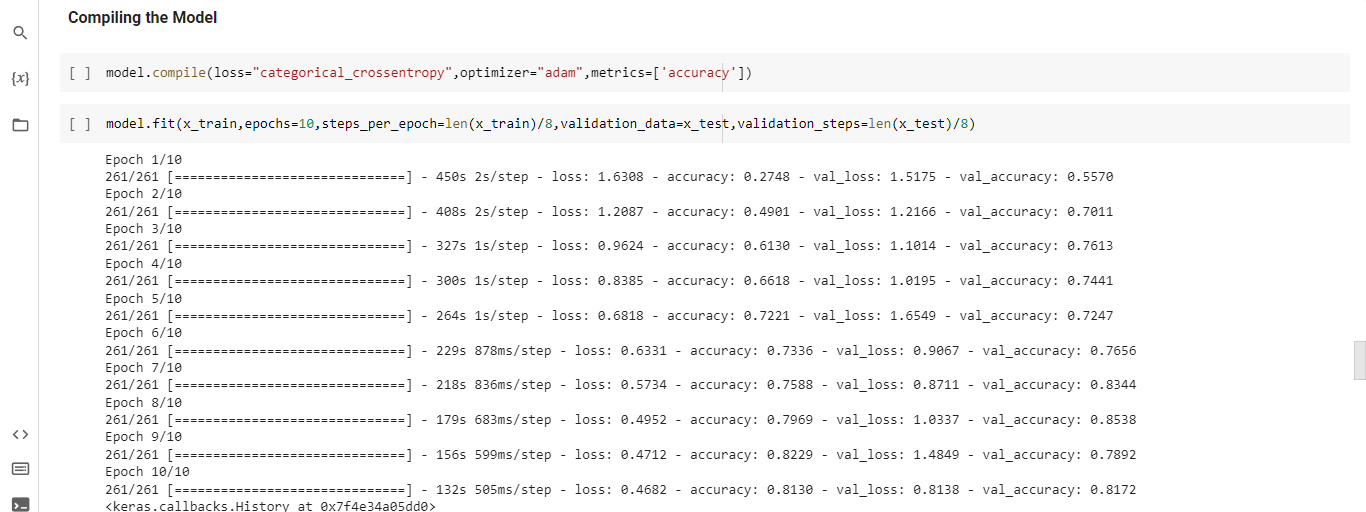
The pre-trained model is tested on different hand gestures of American Sign Language. The model is tested with placing hands in different distances from the camera.

**8.2 User Acceptance Testing**

The model should predict the output without any errors. The accuracy of the model should be more than 80%.

**9. RESULTS**

**9.1 Performance Metrics**

****

**10. ADVANTAGES AND DISADVANTAGES**

**Advantages**

* It does not require any hardwares like Arduino and Raspberry pi.
* It is feasible to work with an web application.

**Disadvantages**

* Powerful hardwares are required to run this loacally.
* The model may give some wrong predictions sometime.

**11. CONCLUSION**

This real-time communication system is an initiative for helping people with disabilities. This is a basic model of a real-time communication system. It can further enhanced to reach new heights. This communication system is used to convey the messages in emergency situations. However, it can be further developed to act as a full fledged interpreter between deaf-mute and normal people.

**12. FUTURE SCOPE**

The model used in this project is created using CNN. But, using RCNN , YOLO kind of algorithms may give some better results. The accuracy can be increased to obtain better predictions. This model can be integrated with Alexa and Amazon echo to implement smart homes.

**13. APPENDIX**

**Source Code**

**Flask app**

from flask import Flask, render\_template,Response

import cv2

from cvzone.HandTrackingModule import HandDetector

from cvzone.ClassificationModule import Classifier

import numpy as np

import math

app=Flask(\_\_name\_\_)

camera=cv2.VideoCapture(0)

detector = HandDetector(maxHands=1)

classifier = Classifier("D:\ASL Recognition\Model\keras\_model.h5",

"D:\ASL Recognition\Model\labels.txt")

offset = 20

imgSize = 300

# folder = "Data/C"

counter = 0

labels = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V",

"W", "X", "Y", "Z", "del", "nothing", "space"]

def gen\_frames():

while True:

success, frame=camera.read()

if not success:

break

imgOutput = frame.copy()

hands, frame = detector.findHands(frame)

if hands:

hand = hands[0]

x, y, w, h = hand['bbox']

imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) \* 255

imgCrop = frame[y - offset:y + h + offset, x - offset:x + w + offset]

imgCropShape = imgCrop.shape

aspectRatio = h / w

if aspectRatio > 1:

k = imgSize / h

wCal = math.ceil(k \* w)

imgResize = cv2.resize(imgCrop, (wCal, imgSize))

imgResizeShape = imgResize.shape

wGap = math.ceil((imgSize - wCal) / 2)

imgWhite[:, wGap:wCal + wGap] = imgResize

prediction, index = classifier.getPrediction(imgWhite, draw=False)

print(prediction, index)

cv2.putText(imgOutput, labels[index], (x, y - 26), cv2.FONT\_HERSHEY\_COMPLEX, 1.7, (255, 255, 255), 2)

else:

k = imgSize / w

hCal = math.ceil(k \* h)

imgResize = cv2.resize(imgCrop, (imgSize, hCal))

imgResizeShape = imgResize.shape

hGap = math.ceil((imgSize - hCal) / 2)

imgWhite[hGap:hCal + hGap, :] = imgResize

prediction, index = classifier.getPrediction(imgWhite, draw=False)

cv2.putText(imgOutput, labels[index], (x, y - 26), cv2.FONT\_HERSHEY\_COMPLEX, 1.7, (255, 255, 255), 2)

cv2.rectangle(imgOutput, (x - offset, y - offset - 50),

(x - offset + 90, y - offset - 50 + 50), (255, 0, 255), cv2.FILLED)

cv2.putText(imgOutput, labels[index], (x, y - 26), cv2.FONT\_HERSHEY\_COMPLEX, 1.7, (255, 255, 255), 2)

cv2.rectangle(imgOutput, (x - offset, y - offset),

(x + w + offset, y + h + offset), (255, 0, 255), 4)

ret,buffer=cv2.imencode('.jpg',frame)

frame = buffer.tobytes()

yield(b'--frame\r\n'

b'Content-Type:image/jpeg\r\n\r\n' + frame+b'\r\n')

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/video\_feed')

def video\_feed():

return Response(gen\_frames(),mimetype='multipart/x-mixed-replace;boundary=frame')

if \_\_name\_\_=='\_\_main\_\_':

app.run(debug=True)

**Hand Gesture Detection**

from cvzone.HandTrackingModule import HandDetector

from cvzone.ClassificationModule import Classifier

import numpy as np

import math

import cv2

cap = cv2.VideoCapture(0)

detector = HandDetector(maxHands=1)

classifier = Classifier("D:\ASL Recognition\Model\keras\_model.h5",

"D:\ASL Recognition\Model\labels.txt")

offset = 20

imgSize = 300

# folder = "Data/C"

counter = 0

labels = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V",

"W", "X", "Y", "Z", "del", "nothing", "space"]

while True:

success, img = cap.read()

if img is None:

break

imgOutput = img.copy()

hands, img = detector.findHands(img)

if hands:

hand = hands[0]

x, y, w, h = hand['bbox']

imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) \* 255

imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]

imgCropShape = imgCrop.shape

aspectRatio = h / w

if aspectRatio > 1:

k = imgSize / h

wCal = math.ceil(k \* w)

imgResize = cv2.resize(imgCrop, (wCal, imgSize))

imgResizeShape = imgResize.shape

wGap = math.ceil((imgSize - wCal) / 2)

imgWhite[:, wGap:wCal + wGap] = imgResize

prediction, index = classifier.getPrediction(imgWhite, draw=False)

print(prediction, index)

cv2.putText(imgOutput, labels[index], (x, y - 26), cv2.FONT\_HERSHEY\_COMPLEX, 1.7, (255, 255, 255), 2)

else:

k = imgSize / w

hCal = math.ceil(k \* h)

imgResize = cv2.resize(imgCrop, (imgSize, hCal))

imgResizeShape = imgResize.shape

hGap = math.ceil((imgSize - hCal) / 2)

imgWhite[hGap:hCal + hGap, :] = imgResize

prediction, index = classifier.getPrediction(imgWhite, draw=False)

cv2.putText(imgOutput, labels[index], (x, y - 26), cv2.FONT\_HERSHEY\_COMPLEX, 1.7, (255, 255, 255), 2)

cv2.rectangle(imgOutput, (x - offset, y - offset - 50),

(x - offset + 90, y - offset - 50 + 50), (255, 0, 255), cv2.FILLED)

cv2.putText(imgOutput, labels[index], (x, y - 26), cv2.FONT\_HERSHEY\_COMPLEX, 1.7, (255, 255, 255), 2)

cv2.rectangle(imgOutput, (x - offset, y - offset),

(x + w + offset, y + h + offset), (255, 0, 255), 4)

cv2.imshow("ImageCrop", imgCrop)

cv2.imshow("ImageWhite", imgWhite)

cv2.imshow("Image", imgOutput)

cv2.waitKey(1)

**Github link**

<https://github.com/IBM-EPBL/IBM-Project-46411-1660746749>

**Demo Video Link**

https://drive.google.com/file/d/1Rzk68wa-v1kgJUAfhpt4bJICUHP3E7AT/view?usp=share\_link