PROJECT REPORT

Real-Time Communication System Powered by AI for Specially Abled

submitted by

PNT2022TMID52686

Clifford Lebo. J - CITC1907005

Kirubakaran. G - CITC1907021

Selva. V - CITC1907042

Vishal. R - CITC1907058

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and Artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions, and in many more areas.

People get to know one another by sharing their ideas, thoughts, and experiences with those around them. There are numerous ways to accomplish this, the best of which is the gift of "Speech." Everyone can very convincingly transfer their thoughts and understand each other through speech. It will be unjust if we overlook those who are denied this priceless gift: the deaf and dumb. In such cases, the human hand has remained the preferred method of communication.

1.2 PURPOSE

The project aims to develop a system that converts 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 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

In our society, we have people with disabilities. Technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communication between a 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 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. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

2.2 REFERENCES

Title: A Comprehensive Study on Deep Learning-based Methods for Sign Language Recognition

Author : Nikolas Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th. Papadopoulos, Vassia Zacharopoulou, George J. Xydopoulos, Klimnis Atzakas, Dimitris Papazachariou, Petros Daras

In this paper, a comparative experimental assessment of computer vision-based methods for sign language recognition is conducted. By implementing the most recent deep neural network methods in this field, a thorough evaluation of multiple publicly available datasets is performed. The present study aims to provide insights into sign language recognition, focusing on mapping non-segmented video streams to glosses. For this task, two new sequence training criteria, known from the fields of speech and scene text recognition, are introduced. Furthermore, a plethora of pretraining schemes are thoroughly discussed. Finally, a new RGB+D dataset for the Greek sign language is created. To the best of our knowledge, this is the first sign language dataset where sentence and gloss-level annotations are provided for video capture.

Title: INCLUDE: A Large-Scale Dataset for Indian Sign Language Recognition

Author : Sridhar, Advaith; Ganesan, Rohith Gandhi; Kumar, Pratyush; Khapra, Mitesh

Indian Sign Language (ISL) is a complete language with its grammar, syntax, vocabulary, and several unique linguistic attributes. It is used by over 5 million deaf people in India. Currently, there is no publicly available dataset on ISL to evaluate Sign Language Recognition (SLR) approaches. In this work, we present the Indian Lexicon Sign Language Dataset - INCLUDE - an ISL dataset that contains 0.27 million frames across 4,287 videos over 263-word signs from 15 different word categories. INCLUDE is recorded with the help of experienced signers to provide a close resemblance to natural conditions. A subset of 50-word signs is chosen across word categories to define INCLUDE-50 for rapid evaluation of SLR methods with hyperparameter tuning. The best-performing model achieves an accuracy of 94.5% on the INCLUDE-50 dataset and 85.6% on the INCLUDE dataset

Title: Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation
Author: Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden

Prior work on Sign Language Translation has shown that having a mid-level sign gloss representation (effectively recognizing the individual signs) improves translation performance drastically. The current state-of-the-art in translation requires gloss-level tokenization to work. We introduce a novel transformer-based architecture that jointly learns Continuous Sign Language Recognition and Translation while being trainable in an end-to-end manner. This is achieved by using a Connectionist Temporal Classification (CTC) loss to bind the recognition and translation problems into a single unified architecture. This joint approach does not require any ground-truth timing information, simultaneously solving two co-dependant sequence-to-sequence learning problems and leading to significant performance gains.
We evaluate the recognition and translation performances of our approaches on the challenging RWTH-PHOENIX-Weather-2014T (PHOENIX14T) dataset. We report state-of-the-art sign language recognition and translation results achieved by our Sign Language Transformers. Our translation networks outperform both sign video to spoken language and gloss to spoken language translation models, in some cases more than doubling the performance (9.58 vs. 21.80 BLEU-4 Score). We also share new baseline translation results using transformer networks for several other text-to-text sign language translation tasks

Title: Dynamic Sign Language Recognition Based on Video Sequence With BLSTM-3D Residual Networks
Author: Yanqiu liao , Pengwen xiong , Weidong min,and jiahao lu

Sign language recognition aims to recognize meaningful movements of hand gestures and is a significant solution in intelligent communication between the deaf community and hearing societies. However, until now, the current dynamic sign language recognition methods still have some drawbacks with difficulties in recognizing complex hand gestures, low recognition accuracy for most dynamic sign language recognition, and potential problems in larger video sequence data training. To solve these issues, this paper presents a multimodal dynamic sign language recognition method based on a deep 3-dimensional residual ConvNet and bi-directional LSTM networks, which is named BLSTM-3D residual network (B3D ResNet). This method consists of three main parts. First, the hand object is localized in the video frames to toduce the time and space complexity of network calculation. Then, the B3D ResNet automatically extracts the spatiotemporal features from the video sequences and establishes an intermediate score corresponding to each action in the video sequence after feature analysis. Finally, by classifying the video sequences, the dynamic sign language is accurately identified. The experiment is conducted on test datasets, including the DEVISIGN_D dataset and SLR_Dataset. The results show that the proposed method can obtain state-of-the-art recognition accuracy (89.8% on the DEVISIGN_D dataset and 86.9% on SLR_Dataset). In addition, the B3D ResNet can effectively recognize complex hand gestures through larger video sequence data, and obtain high recognition accuracy for 500 vocabularies from Chinese hand sign language.

Title: Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues
Author: Muhammad Al-Quraish , Thariq Khalid, and Riad Souissi

People with hearing impairments are found worldwide; therefore, the development of effective local-level sign language recognition (SLR) tools is essential. We conducted a comprehensive review of automated sign language recognition based on machine/deep learning methods and techniques published between 2014 and 2021 and concluded that the current methods require conceptual classification to interpret all available data correctly. Thus, we turned our attention to elements that are common to almost all sign language recognition methodologies. This paper discusses their relative strengths and weaknesses, and we propose a general framework for researchers. This study also indicates that input modalities bear great significance in this field; it appears that recognition based on a combination of data sources, including vision-based and sensor-based channels, is superior to a unimodal analysis. In addition, recent advances have allowed researchers to move from simple recognition of sign language characters and words towards the capacity to translate continuous sign language communication with minimal delay. Many of the presented models are relatively effective for a range of tasks, but none currently possess the necessary generalization potential for commercial deployment. However, the pace of research is encouraging, and further progress is expected if specific difficulties are resolved.

2.3 PROPOSED STATEMENT DEFINITION

This paper describes the system that overcomes the problem faced by the speech and hearing impaired. The objectives of the research are as follows:

1. To design and develop a system that lowers the communication gap between the speech-hearing impaired and the normal world.

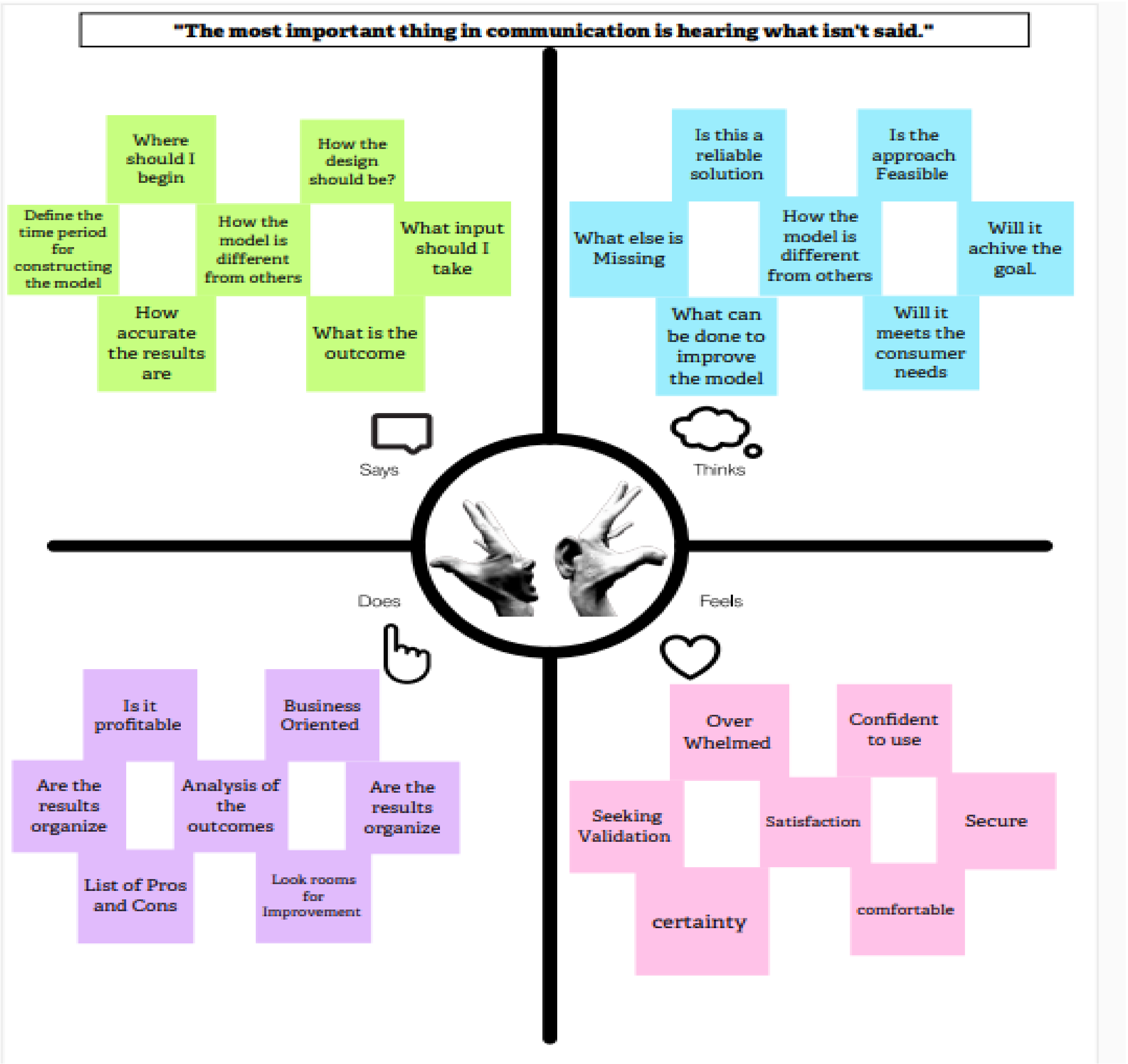
2. To build a communication system that enables the communication between the deaf-dumb person and a normal person.

3. A convolution neural network is being used to develop a model that is trained on various hand movements. This model is used to create an app. This program allows deaf and hard of hearing persons to communicate using signs that are then translated into human-readable text.

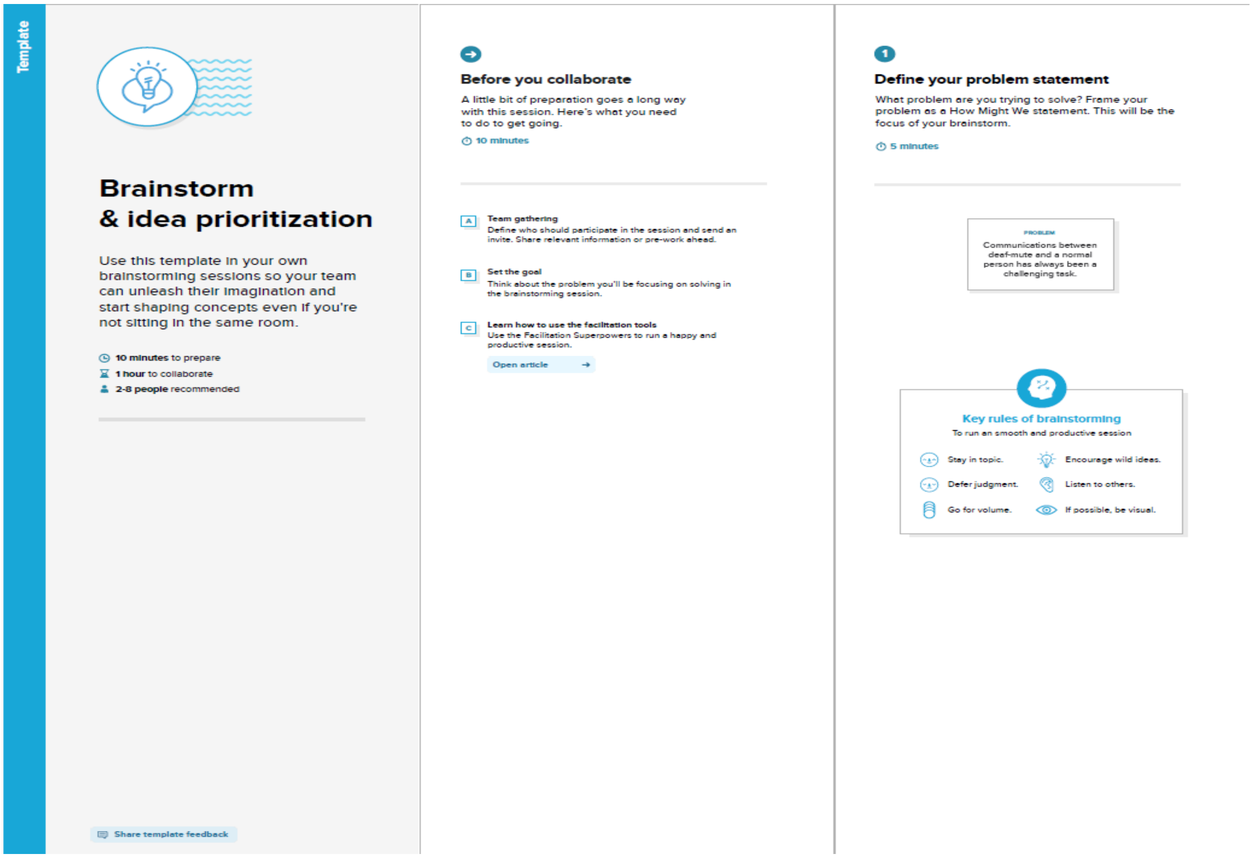
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS


"The most important thing in communication is hearing what isn't said."
Where should I begin
How the design should be?
Define the time period for constructing the model
How the model is different from others
What input should I take
How accurate the results are
What is the outcome
Says
Is this a reliable solution
Is the approach Feasible
What else is Missing
How the model is different from others
Will it achive the goal.
What can be done to improve the model
Will it meets the consumer needs
Thinks
Does
Is it profitable
Business Oriented
Are the results organize
Analysis of the outcomes
Are the results organize
List of Pros and Cons
Look rooms for Improvement
Feels
Over Whelmed
Confident to use
Seeking Validation
Satisfaction
Secure
certainty
comfortable

3.2 IDEATION & BRAINSTORMING:



Template

Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

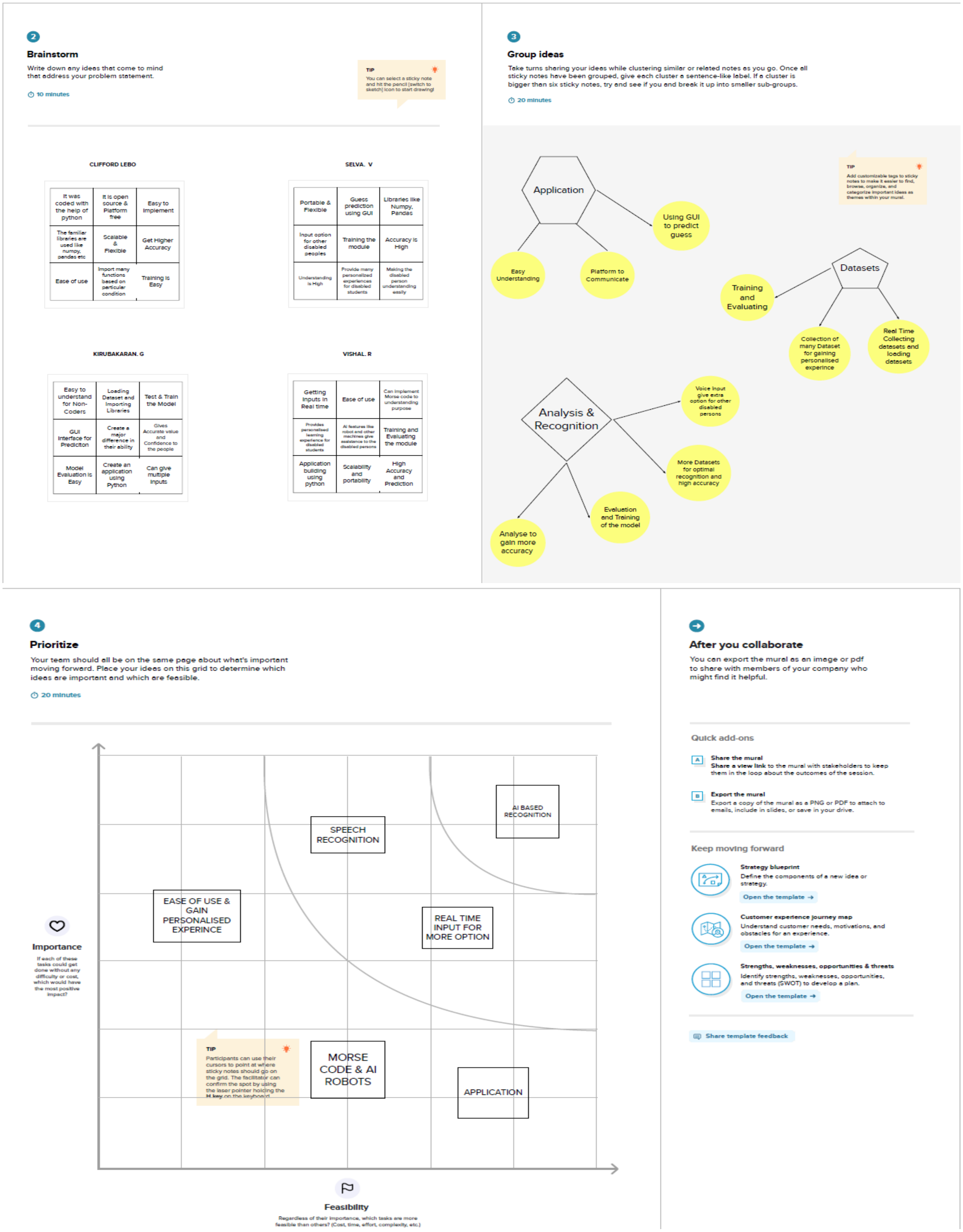
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
Communications between deaf-mute and a normal person has always been a challenging task.

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.



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!

CLIFFORD LEBO

It was coded with the help of python	It is open source & Platform free	Easy to implement
The familiar libraries are used like numpy, pandas etc	Scalable & Flexible	Get Higher Accuracy
Ease of use	Import many functions based on particular condition	Training is Easy

SELVA. V

Portable & Flexible	Guess prediction using GUI	Libraries like Numpy, Pandas
Input option for other disabled peoples	Training the module	Accuracy is High
Understanding is High	Provide many personalized experiences for disabled students	Making the disabled person understanding easily

KIRUBAKARAN. G

Easy to understand for Non-Coders	Loading Dataset and Importing Libraries	Test & Train the Model
GUI Interface for Prediciton	Create a major difference in their ability	Gives Accurate value and Confidence to the people
Model Evaluation is Easy	Create an application using Python	Can give multiple inputs

VISHAL. R

Getting inputs in Real time	Ease of use	Can implement Morse code to understanding purpose
Provides personalised learning experience for disabled students	AI features like robot and other machines give assistance to the disabled persons	Training and Evaluating the module
Application building using python	Scalability and portability	High Accuracy and Prediction

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

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Application
Using GUI to predict guess
Easy Understanding
Platform to Communicate
Datasets
Training and Evaluating
Collection of many Dataset for gaining personalised experince
Real Time Collecting datasets and loading datasets
Analysis & Recognition
Voice Input give extra option for other disabled persons
More Datasets for optimal recognition and high accuracy
Evaluation and Training of the model
Analyse to gain more accuracy

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

AI BASED RECOGNITION
SPEECH RECOGNITION
EASE OF USE & GAIN PERSONALISED EXPERINCE
REAL TIME INPUT FOR MORE OPTION
MORSE CODE & AI ROBOTS
APPLICATION

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

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

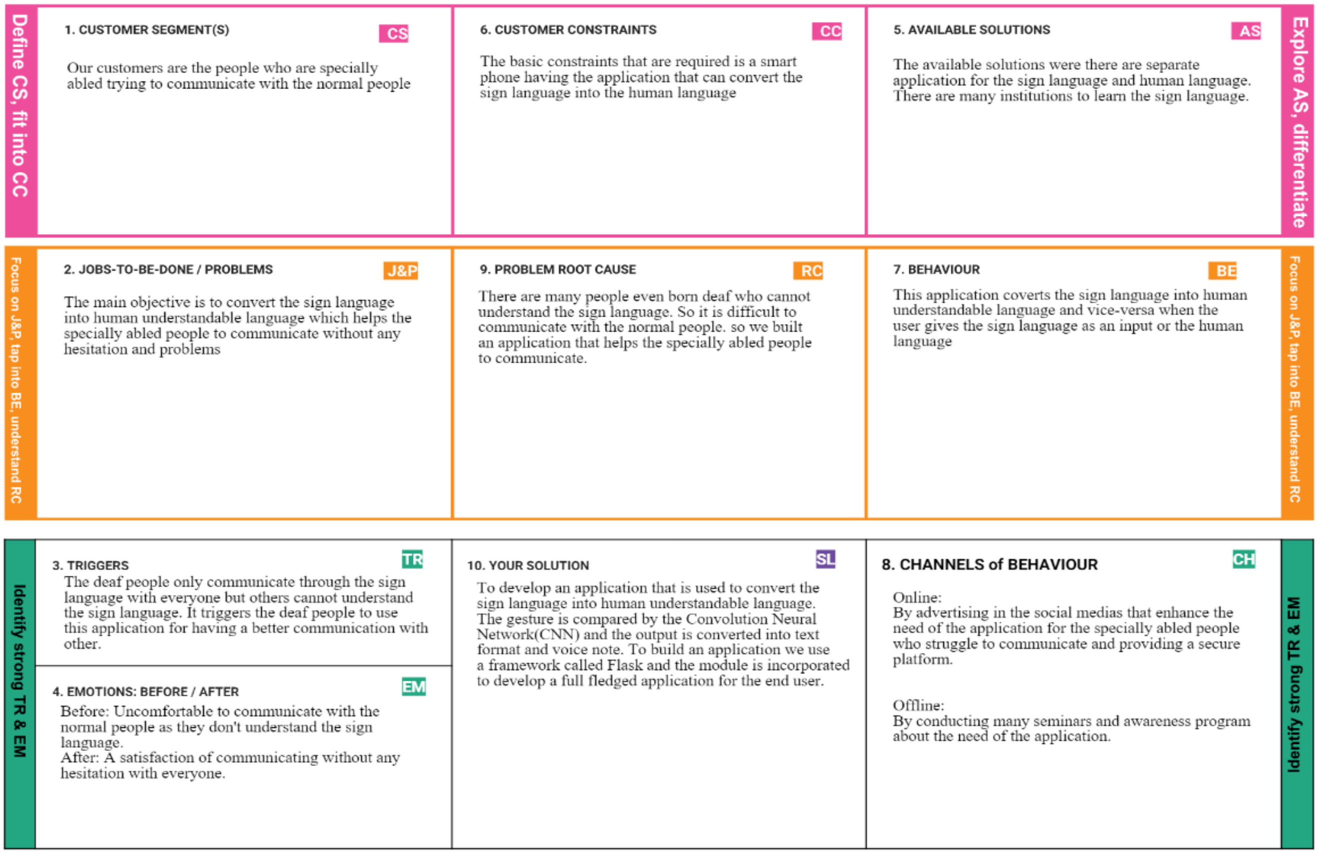
Share template feedback

3.3 PROPOSED SOLUTION

S.NO	Parameter	Description
1	Problem Statement (Problem to be solved)	In our society, we have people with disabilities. Technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communication between deaf-mute and a normal person has always been a challenging task. It is very difficult for the mute people to convey their message to normal people. 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. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.
2	Idea / Solution description	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 that provides a user Interface for the model.
3	Novelty / Uniqueness	We make use of Convolution Neural Network to make deaf and dumb people convey their message through hand gestures that convert into human understandable language
4	Social Impact / Customer Satisfaction	Ease-of-use of the latest technology. Bridges the communication gap. Eliminates the target user’s mind-set to “fit in”. Improves awareness about sign language. Companies widen their target audience to include people with different needs. Removes comprehending related insecurities.
5	Business Model (Revenue Model)	We are using a flask framework where we can integrate our module with the framework and make an application that can be used for specially abled people. They can access the application through the app store or play store and use this application by subscription, which is effective and cost efficient.
6	Scalability of the Solution	For image-recognition problems, convolution neural networks

		(CNNs) are powerful techniques that offer accuracy. Our module helps in providing a efficient software development tool that gives a high accuracy and performance on the gesture recognition and converting into human understandable language

3.4 PROBLEM SOLUTION FIT


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Identify strong TR & EM
CS
CC
AS
J&P
RC
BE
TR
EM
SL
CH

1. CUSTOMER SEGMENT(S)

Our customers are the people who are specially abled trying to communicate with the normal people

6. CUSTOMER CONSTRAINTS

The basic constraints that are required is a smart phone having the application that can convert the sign language into the human language

5. AVAILABLE SOLUTIONS

The available solutions were there are separate application for the sign language and human language. There are many institutions to learn the sign language.

2. JOBS-TO-BE-DONE / PROBLEMS

The main objective is to convert the sign language into human understandable language which helps the specially abled people to communicate without any hesitation and problems

9. PROBLEM ROOT CAUSE

There are many people even born deaf who cannot understand the sign language. So it is difficult to communicate with the normal people. so we built an application that helps the specially abled people to communicate.

7. BEHAVIOUR

This application coverts the sign language into human understandable language and vice-versa when the user gives the sign language as an input or the human language

3. TRIGGERS

The deaf people only communicate through the sign language with everyone but others cannot understand the sign language. It triggers the deaf people to use this application for having a better communication with other.

4. EMOTIONS: BEFORE / AFTER

Before: Uncomfortable to communicate with the normal people as they don't understand the sign language.
After: A satisfaction of communicating without any hesitation with everyone.

10. YOUR SOLUTION

To develop an application that is used to convert the sign language into human understandable language. The gesture is compared by the Convolution Neural Network(CNN) and the output is converted into text format and voice note. To build an application we use a framework called Flask and the module is incorporated to develop a full fledged application for the end user.

8. CHANNELS of BEHAVIOUR

Online:
By advertising in the social medias that enhance the need of the application for the specially abled people who struggle to communicate and providing a secure platform.

Offline:
By conducting many seminars and awareness program about the need of the application.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR.NO	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Disability	Confirmation of the user whether he/she is a deaf or dumb.
FR-4	User Validation	Validating the acquired information of the user
FR-5	User Location	Confirming the location of the user for the security purposes
FR-6	User Hand gesture to text	The gesture of the user is converted into text format via the camera
FR-7	User Hand gesture to voice	The gesture of the user is converted into voice format via the camera

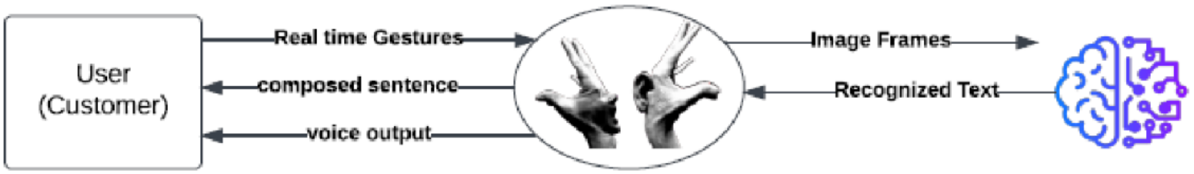
4.2 NON FUNCTIONAL REQUIREMENTS

NFR.NO	**Non-Functional Requirement (Epic)**	**Description**
NFR-1	Usability	The system provides a natural interaction with the users. The specially abled person need not to worry, They can just converse using the language they know (i.e. using hand gestures) which can be understandable by the person in opposite.
NFR-2	Security	The model enables with the high security system, as the user's data won't be shared to the other sources. Special security features like One time Password and 2- step verification are also an integrated part of the system.
NFR-3	Reliability	As the system is build using a rich and state-of-Art of Dataset mostly all the user input can be processed and since all the processing are done on cloud the system is consider to be highly reliable.
NFR-4	Performance	Our system should run on 32 bit (x86) or 64 bit (x64) Dual-core 2.66-GHZ or faster processor.
NFR-5	Availability	The system should be available for the duration of the user access the system until the user terminate the access. The system response to request of the user in less time and the recovery is done is less time
NFR-6	Scalability	It provides an efficient outcome and has the ability to increase or decrease the performance of the system based on the datasets.

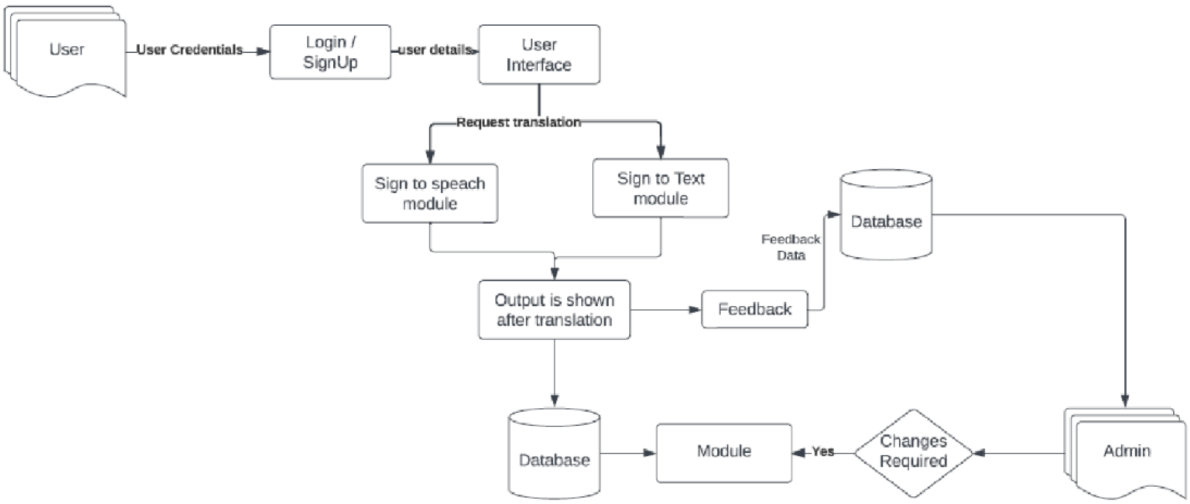
CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

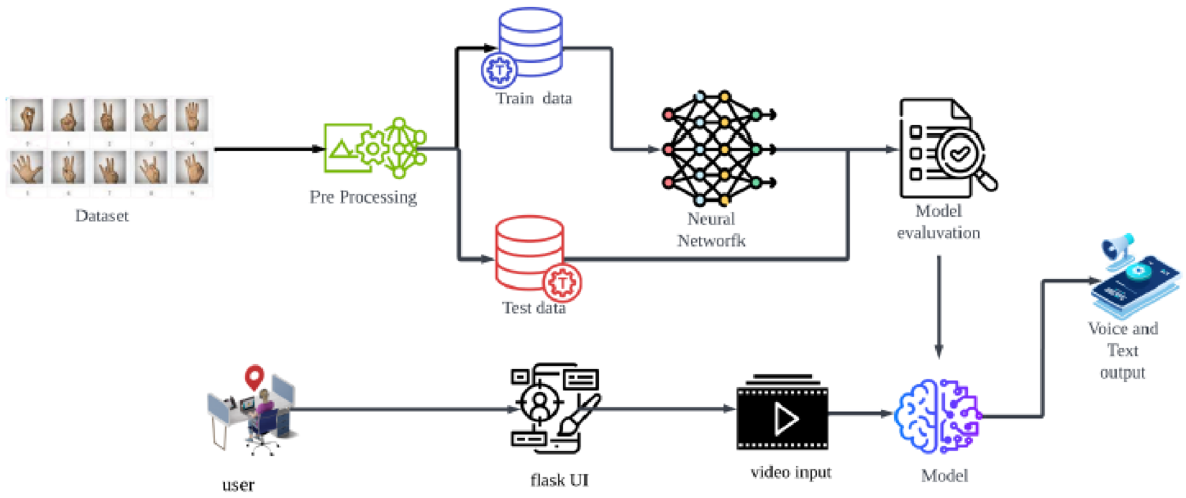


LEVEL 0



DFD - LEVEL 1

5.2 SOLUTION & 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 by entering my email, password, and confirming my password	I can access my account / dashboard	High	
	Authentication	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Low	
	Login	USN-3	As a user, I can log into the application by entering email & password	I am able to get into the Dashboard	High	

	Dashboard	USN-4	One place to explore all available features.	I can access my dashboard	High	
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	I can access my account / dashboard	High	
	Authentication	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Low	
	Login	USN-3	As a user, I can log into the application by entering email & password	I am able to get into the Dashboard	Low	
	Dashboard	USN-4	One place to explore all available features.	I can access my dashboard	Low	
	Upload image	USN-5	As a user, I can upload the sign language image for translating into text format	I can be able to see the appropriate text for the sign language	High	
Administrator	Manage	USN-6	Do-it-yourself service for delivering Everything.	Set of predefined requirements that must be met to mark a user story complete	High	

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Registration	Communi-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Clifford lebo.J Vishal.R
Sprint-1	Registration	Communi-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Kirubakaran.G Selva.V
Sprint-2	Registration	Communi-3	As a user, I can register for the application through Phone number	2	Medium	Kirubakaran.G
Sprint-2	User Interface	Communi-4	Professional responsible for user requirements and needs	2	Medium	Selva.V
Sprint-3	Login	Communi-5	As a user, I can log into the application by entering email & password	1	High	Vishal.R Selva.V
Sprint-3	Dashboard	Communi-6	As a user, I must receive any updates or pop ups in my dashboard	2	High	Clifford lebo.J
Sprint-4	Details	Communi-7	As a user, I should get notification about the progress and any updates via email or sms	1	Medium	Vishal.R
Sprint-4	Privacy	Communi-8	The developed application should be secure for the users	2	High	Kirubakaran.G Clifford lebo.J

6.2 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	6 Days	22 Oct 2022	27 Oct 2022	10	27 Oct 2022
Sprint-2	10	6 Days	29 Oct 2022	03 Nov 2022	10	03 Nov 2022
Sprint-3	10	6 Days	05 Nov 2022	10 Nov 2022	10	11 Nov 2022
Sprint-4	10	6 Days	12 Nov 2022	17 Nov 2022	10	27 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

SOURCE CODE:

FLASK CODE:

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        self.model = load_model('asl_model.h5') # Execute Local Trained Model
        # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM
Trained Model
        self.index=['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',
'I','J','K','L','M','N','O','P','Q','R','S','T','W','X','Y','Z','Del','Space']

        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is:
'+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
```

```
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

APP.PY:

```
from flask import Flask, Response, render_template
from camera import Video

app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary =
frame')


if __name__ == '__main__':
    app.run()
```

CHAPTER 8

TESTING

a. TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS

BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	PASS

b. USER ACCEPTANCE TESTING

i. DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’t Fix	1	0	1	0	2
Total	6	1	4	3	14

ii. TEST CASE ANALYSIS

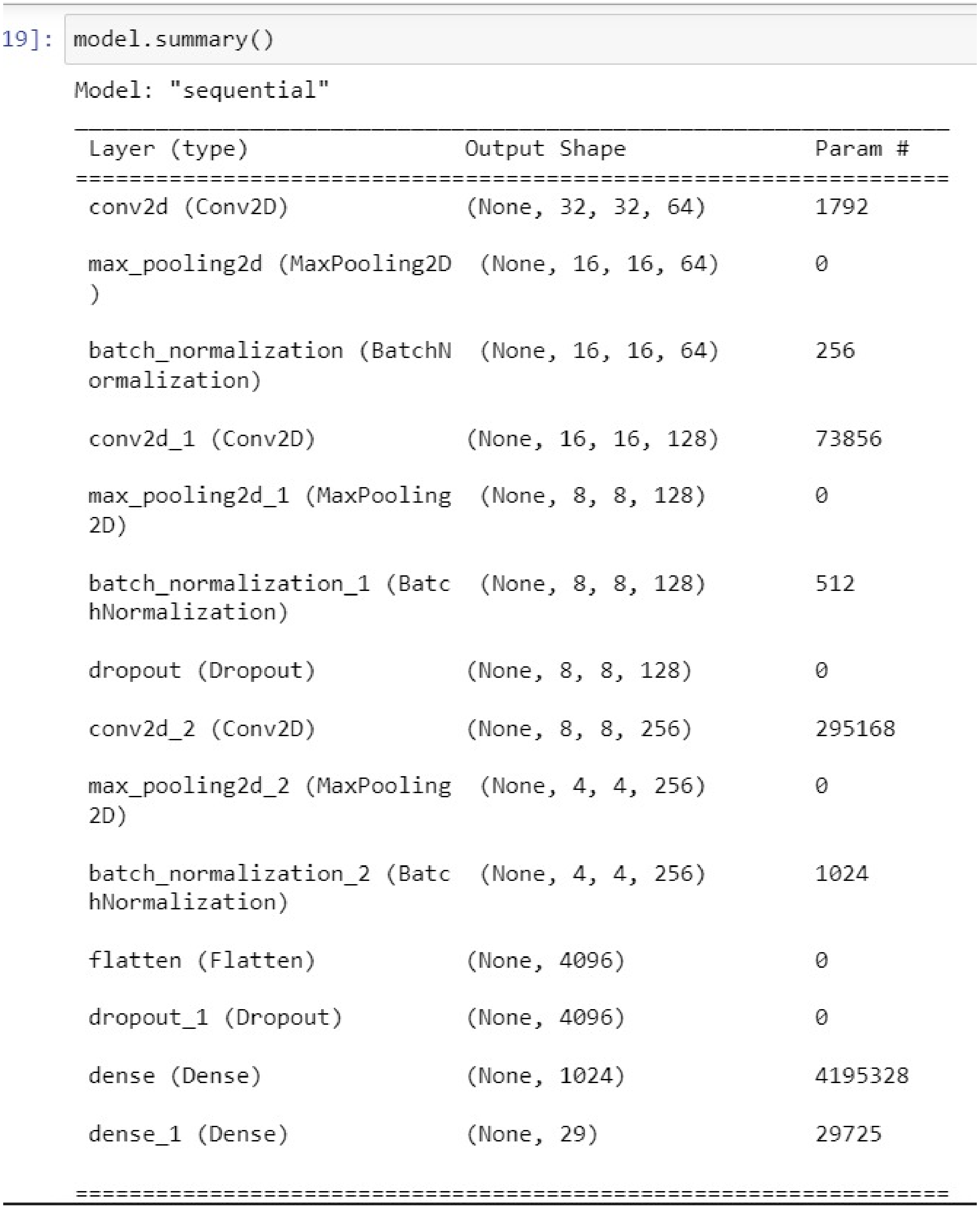
Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

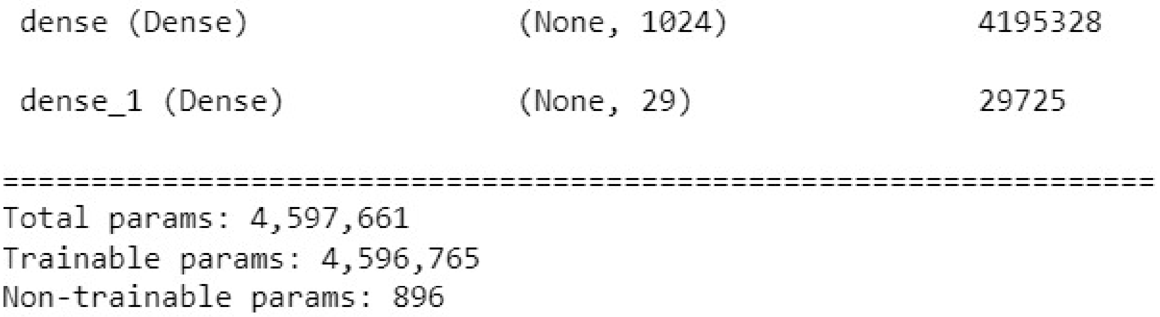
RESULTS

9.1 PERFORMANCE METRICS:

MODEL SUMMARY

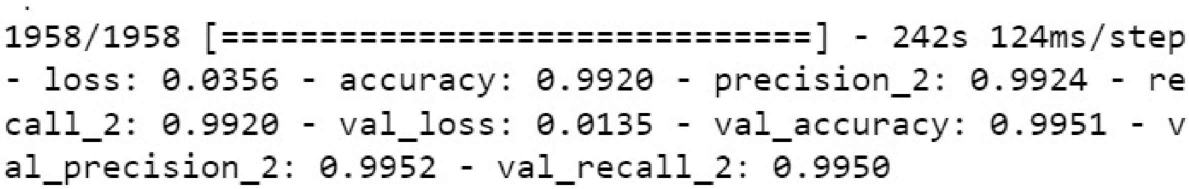


```
19]: model.summary()

     Model: "sequential"
     _________________________________________________________________
      Layer (type)                Output Shape              Param #
     =================================================================
      conv2d (Conv2D)             (None, 32, 32, 64)        1792

      max_pooling2d (MaxPooling2D  (None, 16, 16, 64)       0
      )

      batch_normalization (BatchN  (None, 16, 16, 64)       256
      ormalization)

      conv2d_1 (Conv2D)           (None, 16, 16, 128)       73856

      max_pooling2d_1 (MaxPooling  (None, 8, 8, 128)        0
      2D)

      batch_normalization_1 (Batc  (None, 8, 8, 128)        512
      hNormalization)

      dropout (Dropout)           (None, 8, 8, 128)         0

      conv2d_2 (Conv2D)           (None, 8, 8, 256)         295168

      max_pooling2d_2 (MaxPooling  (None, 4, 4, 256)        0
      2D)

      batch_normalization_2 (Batc  (None, 4, 4, 256)        1024
      hNormalization)

      flatten (Flatten)           (None, 4096)              0

      dropout_1 (Dropout)         (None, 4096)              0

      dense (Dense)               (None, 1024)              4195328

      dense_1 (Dense)             (None, 29)                29725

     =================================================================
```



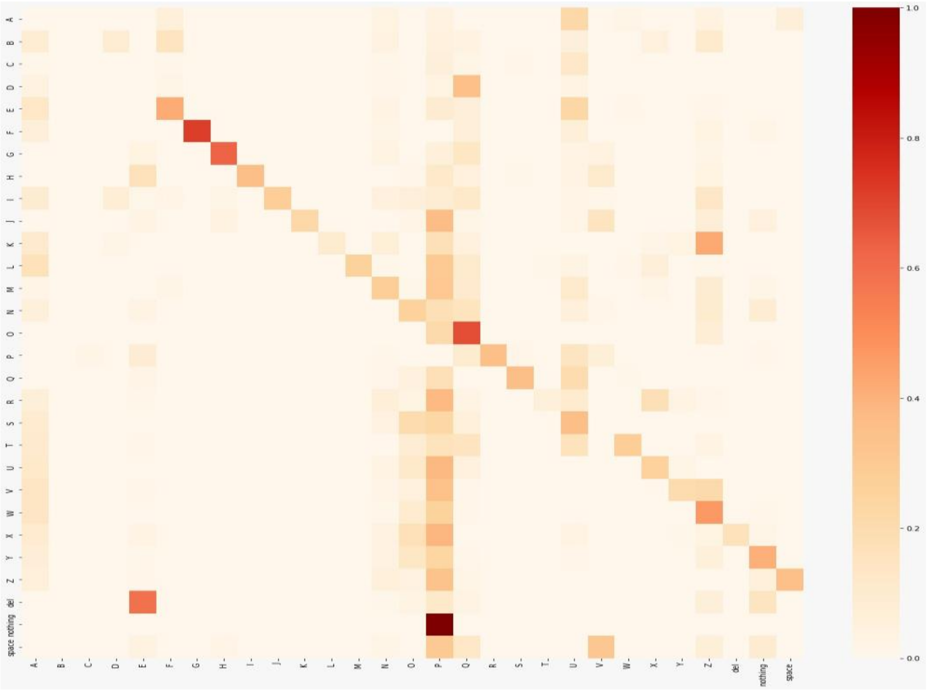
```
 dense (Dense)               (None, 1024)              4195328

 dense_1 (Dense)             (None, 29)                29725

=================================================================
Total params: 4,597,661
Trainable params: 4,596,765
Non-trainable params: 896
```

ACCURACY

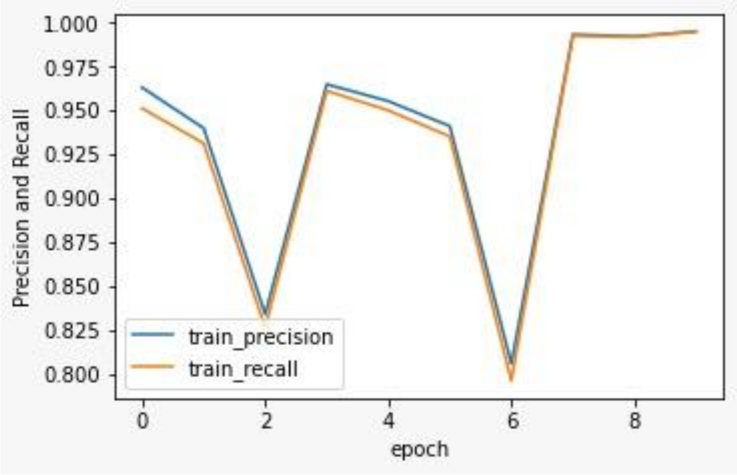


```
1958/1958 [==============================] - 242s 124ms/step
- loss: 0.0356 - accuracy: 0.9920 - precision_2: 0.9924 - re
call_2: 0.9920 - val_loss: 0.0135 - val_accuracy: 0.9951 - v
al_precision_2: 0.9952 - val_recall_2: 0.9950
```

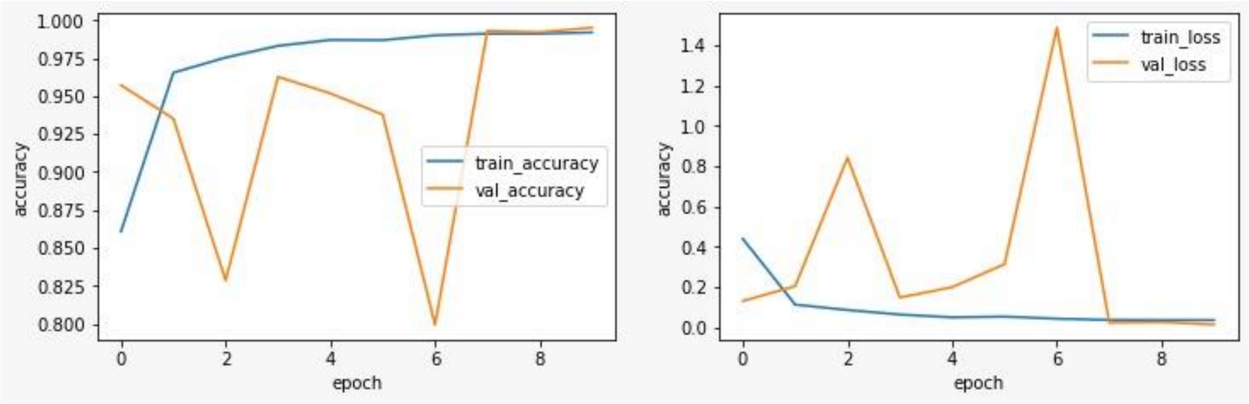
CONFUSION MATRIX



PR CURVE



ROC CURVE



CHAPTER 10

ADVANTAGE & DISADVANTAGE

ADVANTAGE:

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGE:

- A major challenge in gesture recognition is the proper segmentation of skin-colored objects (e.g. hands, face) against a complex static background.
- Illumination variation: Illumination variation affects the accuracy of methods.

CHAPTER 11

CONCLUSION

Communication between Deaf and Dumb people and ordinary people is the aim of the proposed communication system. While some work has been done in this area before, this project adds complete two-sided communication in an efficient manner by implementing the system as an application on a portable device. Thus, it fulfills all its needs. It is evident that the above strategies are time and accuracy efficient. There are still Opportunities for improvement, such as implementing the communicator with other sign languages such as American Sign Language, recognizing accents across the globe, recognizing emotions in sign language, and translating languages.

CHAPTER 12

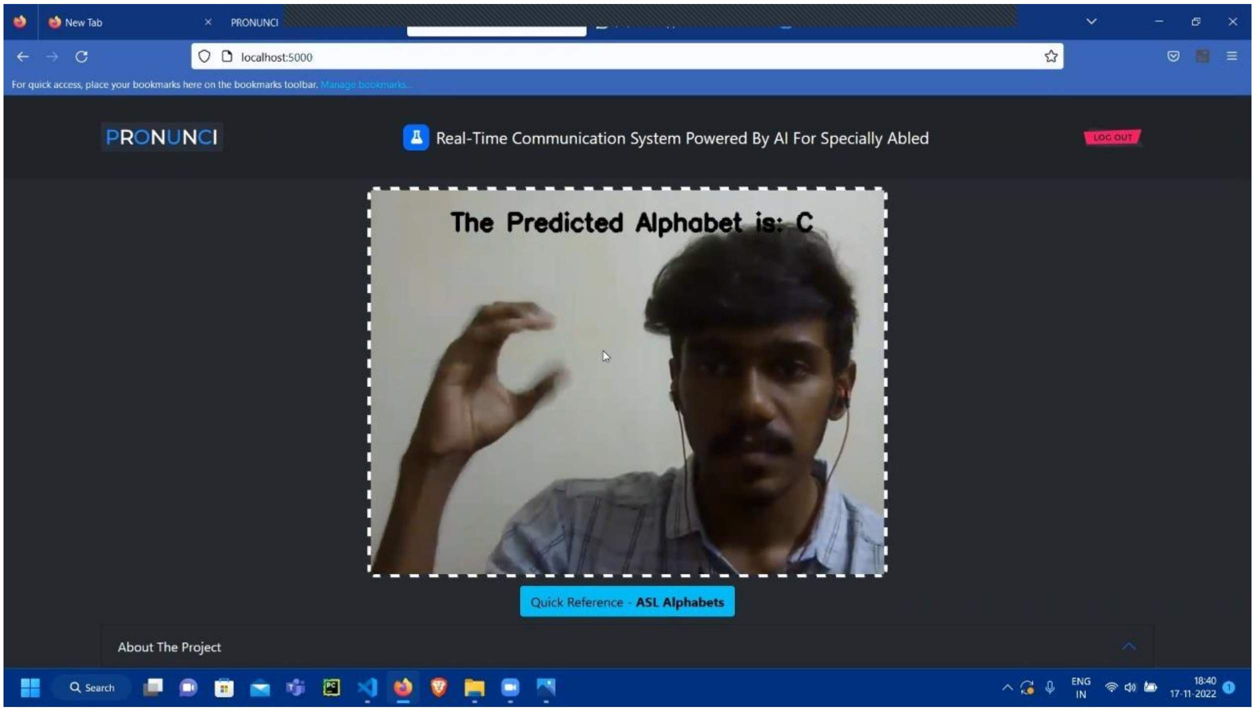
FUTURE SCOPE

This project has a broader scope than what is been implemented.
Some of the future scopes can be:

- Add support to detect words directly.
- Identify text that can be outputted as Voice.
- The model can be improvised to work faster in real time.

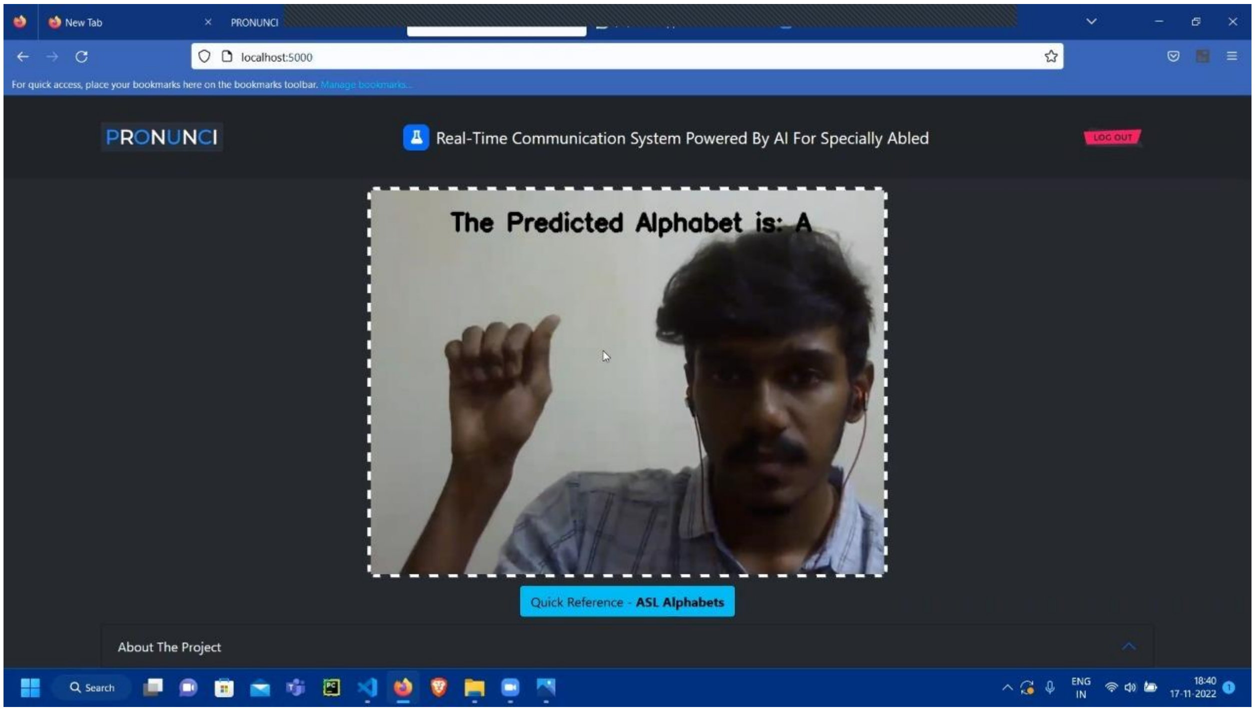
This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

APPENDIX



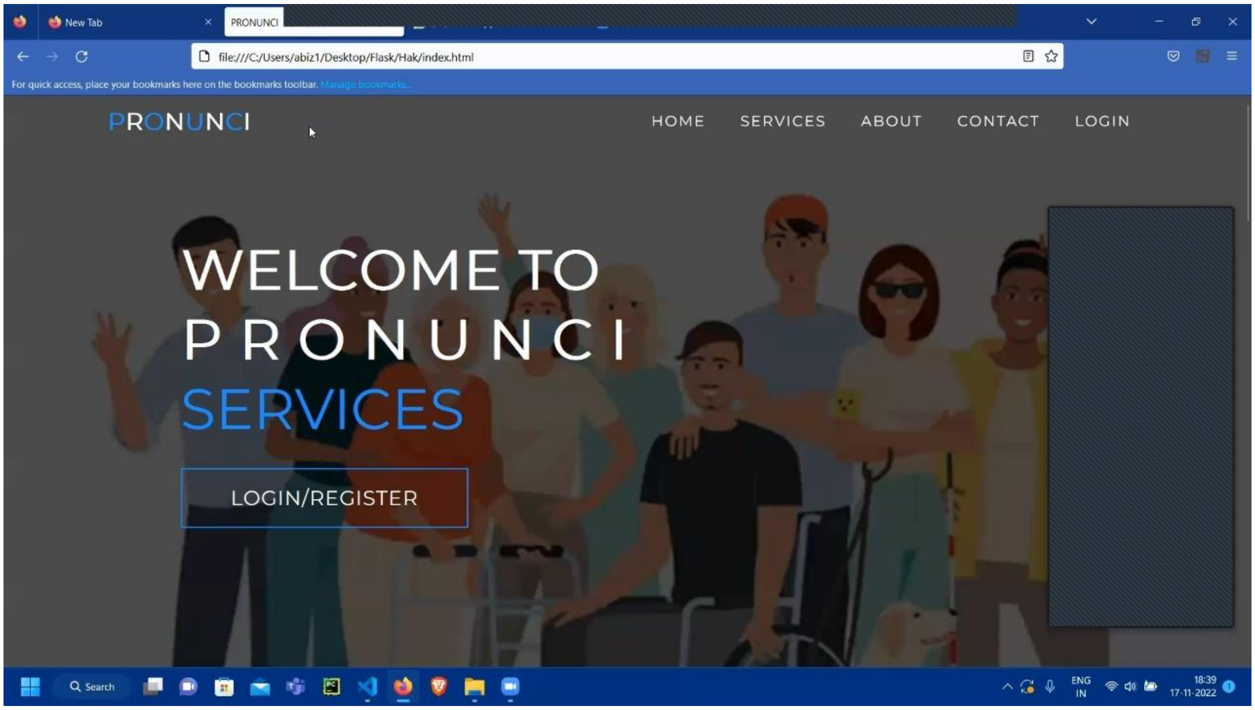
The Predicted Alphabet is: C

New Tab
PRONUNCI
localhost:5000
For quick access, place your bookmarks here on the bookmarks toolbar. Manage bookmarks...
PRONUNCI
Real-Time Communication System Powered By AI For Specially Abled
LOG OUT
Quick Reference - ASL Alphabets
About The Project
Search
ENG
IN
18:40
17-11-2022



The Predicted Alphabet is: A

New Tab
PRONUNCI
localhost:5000
For quick access, place your bookmarks here on the bookmarks toolbar. Manage bookmarks...
PRONUNCI
Real-Time Communication System Powered By AI For Specially Abled
LOG OUT
Quick Reference - ASL Alphabets
About The Project
Search
ENG
IN
18:40
17-11-2022



[redacted]

file:///C:/Users/abiz1/Desktop/Flask/Hak/index.html

For quick access, place your bookmarks here on the bookmarks toolbar. Manage bookmarks...

WELCOME TO
PRONUNCI
SERVICES

[redacted]

SI NO	CODE

1	# To check if GPU is active
	from tensorflow.python.client import device_lib
	import tensorflow as tf
	
	# Load Data
	import os
	import cv2
	import numpy as np
	
	# Data Visualisation
	import matplotlib.pyplot as plt
	
	# Model Training
	from tensorflow.keras import utils
	from tensorflow.keras.optimizers import Adam
	from tensorflow.keras.models import Sequential
	from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D, BatchNormalization
	from sklearn.model_selection import train_test_split

2	# To check if GPU is active from tensorflow.python.client import device_lib import tensorflow as tf # Load Data import os import cv2 import numpy as np # Data Visualisation import matplotlib.pyplot as plt # Model Training from tensorflow.keras import utils

	from tensorflow.keras.optimizers import Adam
	from tensorflow.keras.models import Sequential
	from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D, BatchNormalization
	from sklearn.model_selection import train_test_split

3	#train_dir = '../input/asl-alphabet/asl_alphabet_train/asl_alphabet_train'
	#test_dir = '../input/asl-alphabet/asl_alphabet_test/asl_alphabet_test'
	train_dir='dataset/training_set'
	test_dir='dataset/test_set'

4	def get_data(data_dir) :
	    images = []
	    labels = []
	
	    dir_list = os.listdir(data_dir)
	    for i in range(len(dir_list)):
	        print("Obtaining images of", dir_list[i], "...")
	        for image in os.listdir(data_dir + "/" + dir_list[i]):
	            img = cv2.imread(data_dir + '/' + dir_list[i] + '/' + image)
	            img = cv2.resize(img, (32, 32))
	            images.append(img)
	            labels.append(i)
	
	    return images, labels
	
	X, y = get_data(train_dir)

5	classes = ['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']

6	def plot_sample_images():
	    figure = plt.figure()
	    plt.figure(figsize=(16,5))
	
	    for i in range (0,29):
	        plt.subplot(3,10,i+1)

	plt.xticks([])
	plt.yticks([])
	path = train_dir + "/{0}/{0}1.jpg".format(classes[i])
	img = plt.imread(path)
	plt.imshow(img)
	plt.xlabel(classes[i])
	
	plot_sample_images()

7	def preprocess_data(X, y):
	np_X = np.array(X)
	normalised_X = np_X.astype('float32')/255.0
	
	label_encoded_y = utils.to_categorical(y)
	
	x_train, x_test, y_train, y_test = train_test_split(normalised_X, label_encoded_y, test_size = 0.1)
	
	return x_train, x_test, y_train, y_test
	
	x_train, x_test, y_train, y_test = preprocess_data(X, y)

8	print("Training data:", x_train.shape)
	print("Test data:", x_test.shape)

9	classes = 29
	batch = 32
	epochs = 15
	learning_rate = 0.001

10	model = Sequential()
	
	model.add(Conv2D(64, (3, 3), padding='same', input_shape=(64, 64, 3), activation='relu'))
	model.add(MaxPooling2D(pool_size=(2, 2)))
	model.add(BatchNormalization())

	model.add(Conv2D(128, (3, 3), padding='same', input_shape=(64, 64, 3), activation='relu'))
	model.add(MaxPooling2D(pool_size=(2, 2)))
	model.add(BatchNormalization())
	model.add(Dropout(0.2))
	
	model.add(Conv2D(256, (3, 3), padding='same', input_shape=(64, 64, 3), activation='relu'))
	model.add(MaxPooling2D(pool_size=(2, 2)))
	model.add(BatchNormalization())
	
	model.add(Flatten())
	model.add(Dropout(0.2))
	model.add(Dense(1024, activation='relu'))
	model.add(Dense(classes, activation='softmax'))

11	adam = Adam(lr=learning_rate)
	model.compile(optimizer=adam, loss='categorical_crossentropy', metrics=['accuracy'])

12	model.summary()
13	history = model.fit(x_train, y_train, batch_size=batch, epochs=epochs, validation_split=0.2, shuffle = True, verbose=1)

14	#save model
	model.save('aslmodel.h5')

15	test_loss, test_acc = model.evaluate(x_test, y_test)
	print('Test accuracy:', test_acc)
	print('Test loss:', test_loss)

16	def plot_results(model):
	
	plt.figure(figsize=(12, 12))
	plt.subplot(3, 2, 1)
	plt.plot(history.history['accuracy'], label = 'train_accuracy')
	plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
	plt.xlabel('epoch')
	plt.ylabel('accuracy')

	plt.legend()
	  plt.subplot(3, 2, 2)
	  plt.plot(history.history['loss'], label = 'train_loss')
	  plt.plot(history.history['val_loss'], label = 'val_loss')
	  plt.xlabel('epoch')
	  plt.ylabel('accuracy')
	  plt.legend()
	  plt.show()
	
	plot_results(model)

17	files = []
	labels = []
	directory = 'C:\\\\Users\\\\Cliff\\\\Desktop\\\\Real-Time-Communication-Specially-Abled-main\\\\Real-Time-Communication-Specially-Abled-main\\\\ProjectFiles\\\\dataset\\\\training_set'
	sub_directory = os.listdir(directory)
	for s in sub_directory:
	    for f in os.listdir(os.path.join(directory, s)):
	        labels.append(s)
	        files.append(os.path.join(directory, s, f))
	print(labels[0], files[0]) VALIDATION_SPLIT = 0.2
	SEED = 2
	IMAGE_SIZE = (32, 32)
	#EPOCHS = 20
	LEARNING_RATE = 0.05
	BATCH_SIZE = 128
	#PATIENCE = 3
	train_ds = tf.keras.preprocessing.image_dataset_from_directory(
	            directory,
	            validation_split = VALIDATION_SPLIT,
	            subset = 'training',
	            seed = SEED,

	image_size = IMAGE_SIZE,
	batch_size = BATCH_SIZE)
	
	val_ds = tf.keras.preprocessing.image_dataset_from_directory(
	directory,
	validation_split = VALIDATION_SPLIT,
	subset = 'validation',
	seed = SEED,
	image_size = IMAGE_SIZE,
	batch_size = BATCH_SIZE)

18	from tensorflow.data import AUTOTUNE
	normalization_layer = tf.keras.layers.experimental.preprocessing.Rescaling(1./255)
	class_names = train_ds.class_names
	
	train_ds = train_ds.map(lambda x, y: (normalization_layer(x), y), num_parallel_calls = AUTOTUNE)
	val_ds = val_ds.map(lambda x, y: (normalization_layer(x), y),num_parallel_calls = AUTOTUNE)

19	class_names num_classes = len(class_names)

20	plt.figure(figsize=(10, 10))
	for images, labels in train_ds.take(1):
	  for i in range(9):
	    ax = plt.subplot(3, 3, i + 1)
	    plt.imshow(images[i].numpy().astype("float32"))
	    plt.title(class_names[labels[i]])
	    plt.axis("off")

21	train_ds = train_ds.cache().prefetch(buffer_size=AUTOTUNE)
	val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)

22	predictions = model.predict(val_ds)
	predictions = tf.argmax(predictions, axis = 1)
	validations = tf.concat([y for x, y in val_ds], axis = 0)

	conversions = {index : letter for letter, index in zip(class_names, range(len(class_names)))}

23	test = predictions[predictions == 1]
	pred_labels = np.empty(shape = predictions.numpy().shape, dtype = 'U10')
	val_labels = np.empty(shape = predictions.numpy().shape, dtype = 'U10')
	
	for k, v in conversions.items():
	    predictions_n = predictions.numpy()
	    validations_n = validations.numpy()
	    pred_labels[predictions_n == k] = v
	    val_labels[validations_n == k] = v

24	from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
	import pandas as pd
	import seaborn as sns
	conf = confusion_matrix(val_labels, pred_labels)
	row_sums = conf.sum(axis = 1, keepdims = True)
	norm_conf = conf/row_sums
	np.fill_diagonal(norm_conf, 0)
	index = class_names
	column = class_names
	cm_df = pd.DataFrame(norm_conf, index, column)
	fig, ax = plt.subplots(1, figsize = (20, 20))
	sns.heatmap(cm_df)