REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR ESPECIALLY ABLED

A PROJECT REPORT

Submitted by

AGATH CLAFIO.P (814719104001)

ESWAR MARKX.P.K (814719104008)

APSARAA.S (814719104006)

EZHILARASHI.T.S (814719104009)

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRM TRP Engineering College , TRICHY

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW:

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

1.2 PURPOSE:

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:

S.No	TITLE	PROPOSED WORK	TOOLS USED/ ALGORITHM	TECHNOLOGY	ADVANTAGES/ DISADVANTAGES
1	Artificial Intelligence enabled virtual sixth sense application for the disabled	The sixth sense is a multiplatform app for aiding people in need that is people who are handicapped in the form of lack of speech (dumb), lack of hearing (deaf), or lack of sight (blind).	❖ ML OCR kit ❖ Firebase ML toolkit ❖ Google Web toolkit TTS	❖ Android smart phones ❖ Object Detection ❖ Text Recognition ❖ AP	Help dumb people to easily and quickly communicate with normal people./The application still does depend on the camera picture quality for object detection
	Design of a Communication System using	Our goal is to design a human computer	❖ Feature Extraction ❖ Sign to text	❖ Blob Detection ❖ Skin color recognition	Hand gestures of deaf people by normal peoples this system is

2	Sign Language aid for Differently Abled Peoples.	interface a system that can accurately identify the language of the deaf and dumb.	and Speech Conversion ❖Image preprocessing and segmentation.	❖Template Matching	proposed and it gives output in the form of sound./A mediator is required to know the sign language
3	D-Talk: Sign Language Recognition System for People with Disability using Machine Learning and Image Processing	D-talk is a system that allows people who are unable to talk and hear and for them to learn their language easier and also for the people that would interact with them.	❖Image Recognition process ❖ Object Detection ❖ Gesture Recognition ❖ HSV Algorithm.	❖ Machine learning ❖ Deep learning ❖ Decision tree	Speech interpretation is helpful for sign language non speakers who wants the hand sign to understand./ The type of inaccuracy can emerge from users, such as poor web camera
4	Real-time Communication System for the Deaf and Dumb	Aims to aid the deaf-mute by creation of a new system that helps convert sign language to text and speech for easier communication with audience.	❖ Flex sensor ❖ Arduino Uno ❖ Arduino IDE	❖Python Programming Language ❖Gesture recognition	The system forms the base infrastructure for a complete communicational aid system for the deaf and mute./it requires logical mechanism for classification of letters based on sensor values.
5	AN ANDROID APPLICATION	In this paper, we introduce	❖ Sign language keyboard app	❖ Eclipse	Offer a great tool for parents to teach their

	TO AID UNEDUCATED DEAF DUMB PEOPLE	an integrated android application to blend uneducated Deaf Dumb people within society, and help them to communicate with normal people.		❖ SQL Lite ❖ Java	deaf and dumb kids And Introduce Sign language keyboard./this application introduces an easy translator from sign language to English or Arabic language.
6	A Sign Language Recognition for The Deaf and Dumb	It is a software which presents a system prototype that is able to automatically recognize sign language to help deaf and dumb people to communicate more effectively with each other or normal people	❖Gesture classification ❖ CNN model ❖ Hand gesture ❖ ANN	❖ OpenCV feature ❖ extraction	A functional real time vision based American sign language recognition for Deaf and Dumb people have been developed with accuracy of 92%. /We couldn't find any existing dataset.

2.2 REFERENCES:

[1] S. -H. Han and H. -J. Choi, "Checklist for Validating Trustworthy AI," *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2022, pp. 391-394, doi: 10.1109/BigComp54360.2022.00088.

[2] Q. Ye, "A Novel FPGA Control Process with AI-Fuzzy VCR Algorithm," *2010 International Conference on Electrical and Control Engineering*, 2010, pp. 190-193, doi: 10.1109/iCECE.2010.53.

[3] T. Charrot, J. Guegan, A. Napoli and C. Ray, "Port Type Prediction Based on Machine Learning and AIS Data Analysis," *OCEANS 2021: San Diego – Porto*, 2021, pp. 1-5, doi: 10.23919/OCEANS44145.2021.9705864.

[4] L. Xuemei, L. Yan and L. Jincheng, "Application of AI Algorithm in Video Indexing and Retrieval," *2009 Third International Symposium on Intelligent Information Technology Application*, 2009, pp. 686-688, doi: 10.1109/IITA.2009.522.

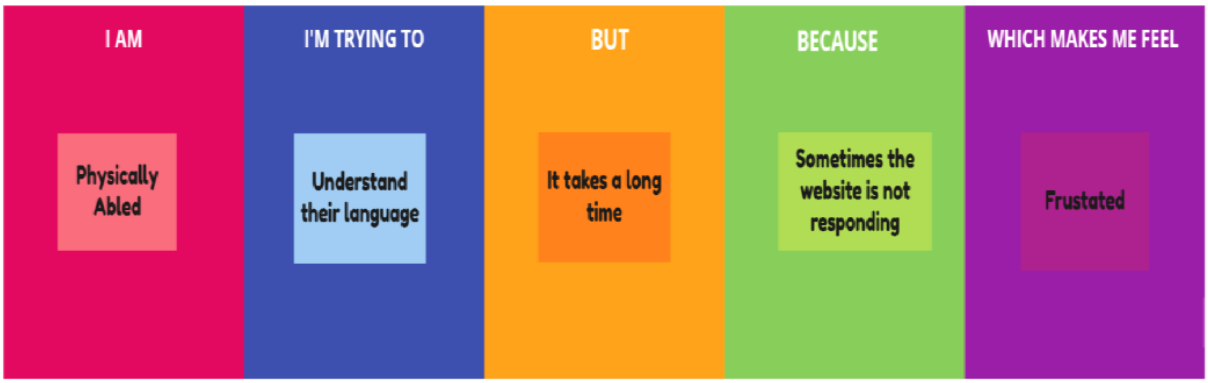
[5] R. H. Rachmadi, R. Azzahra, R. A. Darmawan, P. A. Nigo and N. N. Qomariyah, "Developing AI Bots with Minimax Algorithm for Surakarta Board Game," *2021 International Conference on ICT for Smart Society (ICISS)*, 2021, pp. 1-6, doi: 10.1109/ICISS53185.2021.9533206.

2.3 PROBLEM STATEMENT DEFINITION:

A problem statement is a concise description of an issue to be addressed or a condition to be improved upon. It identifies the gap between the current state and desired state of a process or product.

For the given project, the following set of questions are raised .

- I AM?
- I'M TRYING TO?
- BUT
- BECAUSE
- WHICH MAKES ME FEEL



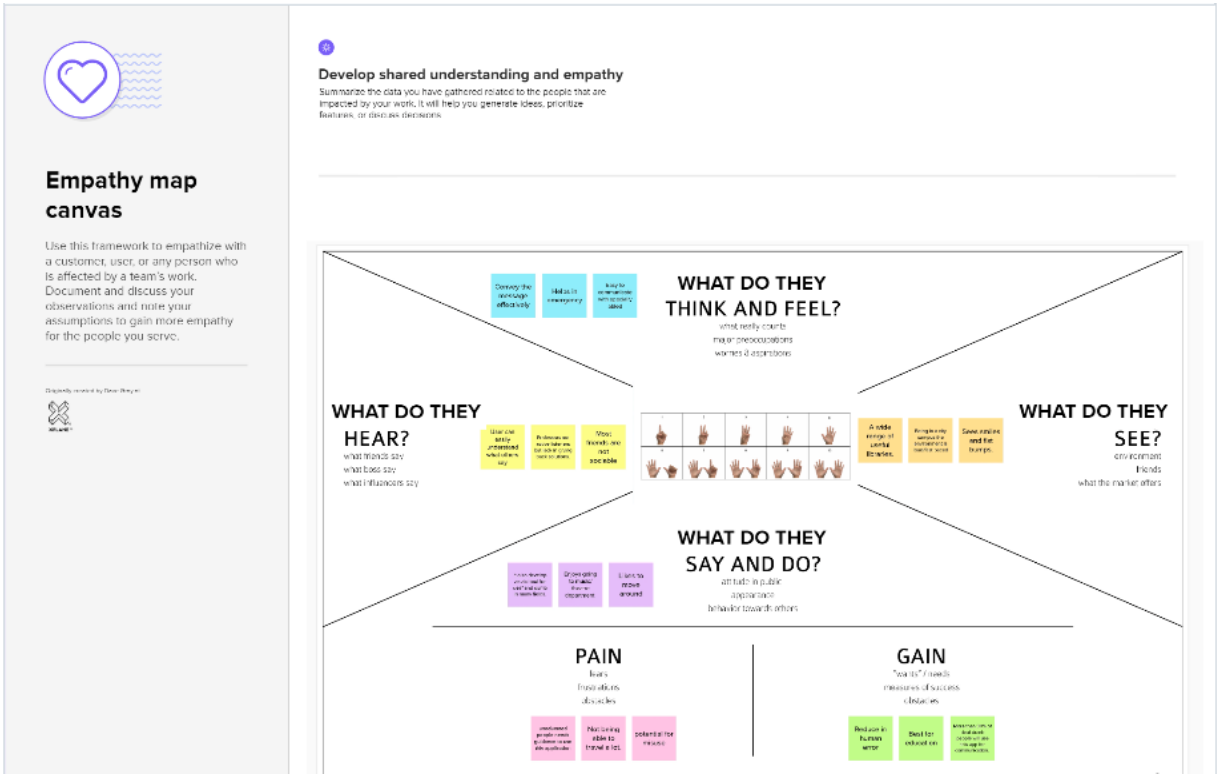
CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. Empathy maps should be used throughout any UX process to establish common ground among team members and to understand and prioritize user needs. In user-centered design, empathy maps are best used from the very beginning of the design process.It externalizes knowledge about users in order to

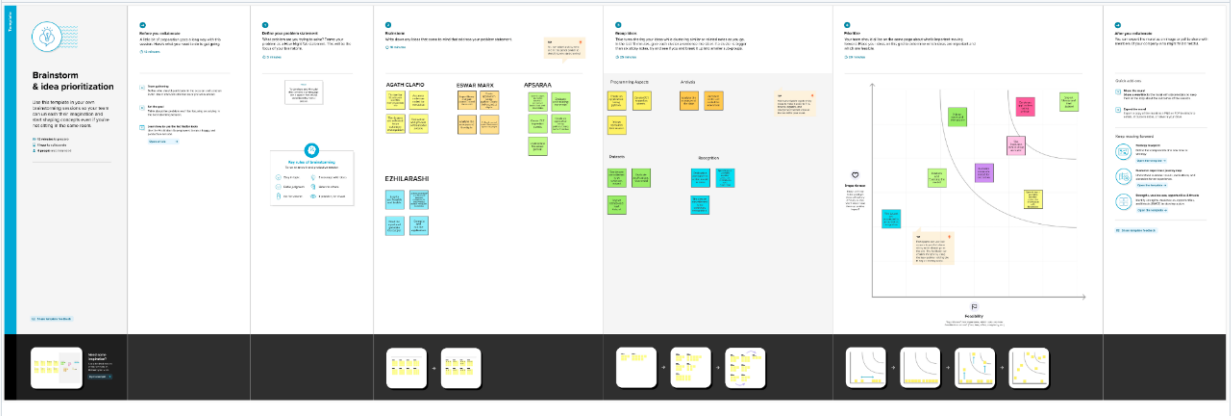
- Create a shared understanding of user needs,
- Aid in decision making.



3.2 IDEATION & BRAINSTORMING:

Brainstorming combines a relaxed, informal approach to problem solving with lateral thinking. It encourages people to come up with thoughts and ideas that can, at first, seem a bit crazy. Some of these ideas can be crafted into original, creative solutions to a problem, while others can spark even more ideas. This helps to get people unstuck by "jolting" them out of their normal ways of thinking.

For the given Project, different ideas and methods were suggested in order to come up with a solution for the existing problem.



3.3 PROPOSED SOLUTION

The main objective of the project is to contribute to the field of automatic sign language recognition. Our focus is mainly on the recognition of the real time sign language gestures. This work focused on deep learning approach to numbers, alphabets and most often used words. Through this work we want to ease the interaction for people with speech and hearing disabilities and also other objective is to convert sign language into text.

Problem statement:

To provide an Efficient communication app which translates the hand signs into text and voice mode for deaf and dumb people.

Idea / Solution description:

- Convolution Neural Networks are to be used to take hand sign as an input to extract edges, corners.
- Dataset is used for training CNN. One dataset for hand detection and the other for gesture detection.
- Voice assistant is implemented that take input as speech patterns and convert the text into voice

Novelty / Uniqueness:

We have number of symbols to be trained for our project and many of them look similar to each other like the gesture for symbol ‘V’ and digit ‘2’ .To produce better accuracies, we keep the background of hand a stable single colour, so that we don’t need to segment it on basis of skin colour.

Social Impact / Customer Satisfaction:

- AI enables people with disabilities to lead an independent life with this app.
- Supporting them in activities of daily living .
- It changes the mind set of the disabled, that even they can to be involved in a common conversation like others.

Business Model (Revenue Model):

- Faster and efficient , the concerned text or voice as output is produced, the more it leads to optimize the app with new advancements.
- The productivity is gained and at the same time, leads to improved speed of business.

Scalability of the Solution:

- A convolutional neural network can be scaled in three dimensions:depth, width, resolution.
- Depth of the network corresponds to the number of layers in a network.
- Width is associated with the number of neurons in a layer.
- Resolution is the image resolution that is being passed to CNN. Increasing the depth, by stacking more convolutional layers, allows the network to learn more complex features.

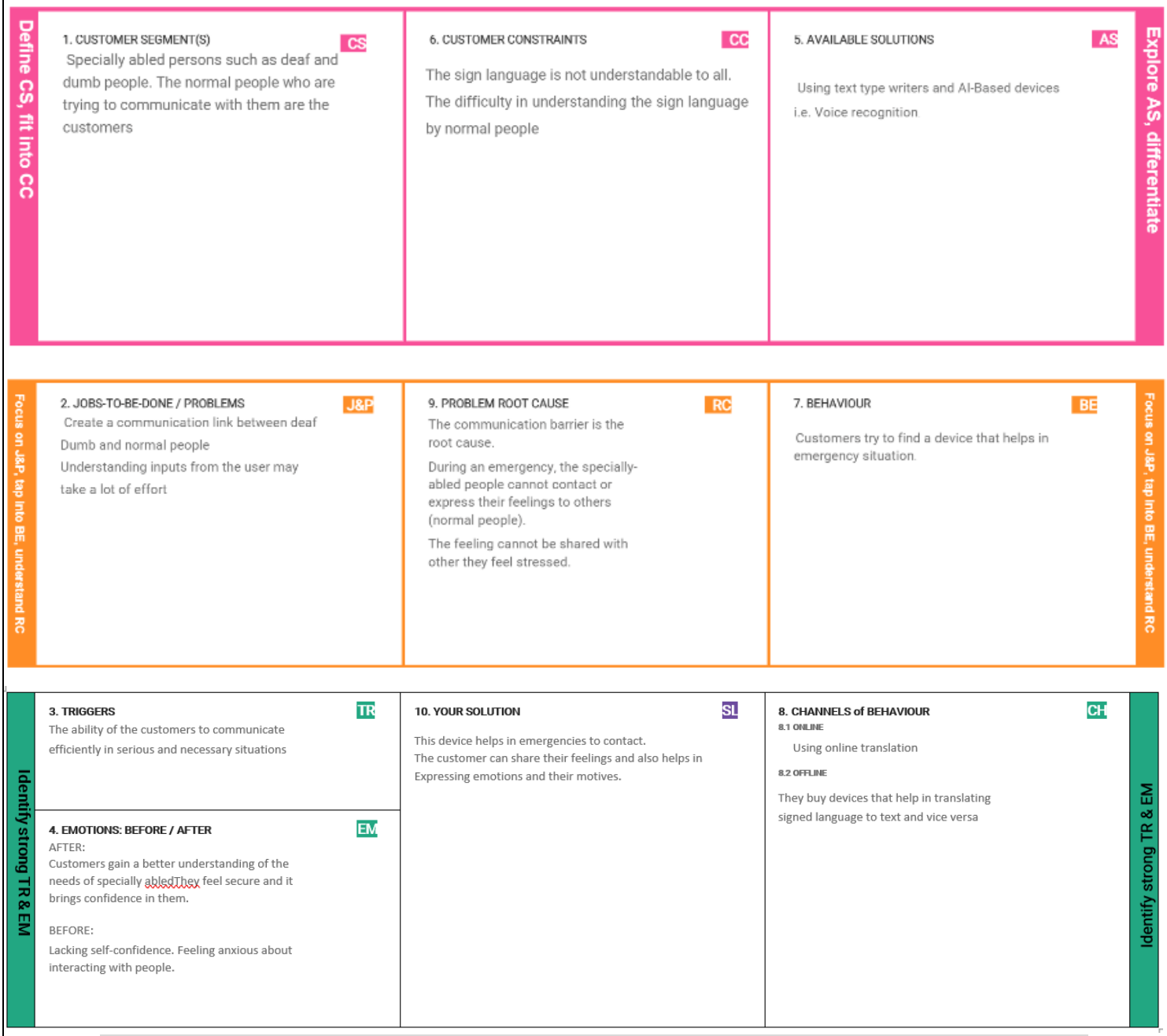
The project team shall fill in the following information in the proposed solution template.

S.NO.	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	Differently able like dump and mute people can communicate only through sign language, normal people who do not know sign language feels difficult to communicate with them.
2	Idea / Solution description	To overcome this problem we have an idea that an application is created to communicate with the normal people.
3	Novelty / Uniqueness	This process the image of the person who is using sign language and converts it into the voice by analyzing the sign used.
4	Social Impact / Customer Satisfaction	Differently able people feel free to communicate and it brings a huge difference compared past.
5	Business Model (Revenue Model)	Many people in the world are differently able, this application will become more popular among them and it will be installed by all and it will be used, and so it will produce more money.
6	Scalability of the Solution	Thus this would bring a new evolution in Real-Time Communication System Powered by AI for Specially Able with less time and safe enough resources.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that we have found a problem with the customer and the solution we realized for it actually solves the customer's problem.

For the given project, the following situations are taken into consideration and at last proposed the innovative ideas in order to provide solution to the problem



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

- Here , Desktop along with Camera is presented as black box.
- Deaf/Dumb is the person, who will show different signs based on the type of information being conveyed.
- Normal Person is the passive user of the desktop.

The System requirements that are required are specified below,

- Deaf/Dumb person should be able to perform a sign that represents digit/number.
- Deaf/Dumb person should be able to perform a sign that represents a character.
- Deaf/Dumb person should be able to perform a sign , where group of characters forms a word.
- Deaf/Dumb person should be able to perform a sign, where group of words forms a sentence.
- Especially Deaf person should be able to see the translation ofsign to text format.
- Dumb person should be able to understand the conversion of text into voice mode.
- Normal user should be able to understand the corresponding information conveyed by disabled through sign language.

4.2 NON-FUNCTIONAL REQUIREMENTS:

Some of the non -functional requirements are mentioned as shown below.

I. SPEED:

Speed determines how fast an application responds to commands. For example, it can be considered the amount of time , the concerned application gets opened once clicked.

II. COMPATIBILITY:

Highly compatible systems typically function well when other applications are running on a device. Compatibility also allows people who have different operating systems to use the same applications.

III. CAPACITY:

The capacity of a system refers to the amount of storage it offers. When using some applications, users can adjust and save settings based on their preferences.

IV. RELIABILITY:

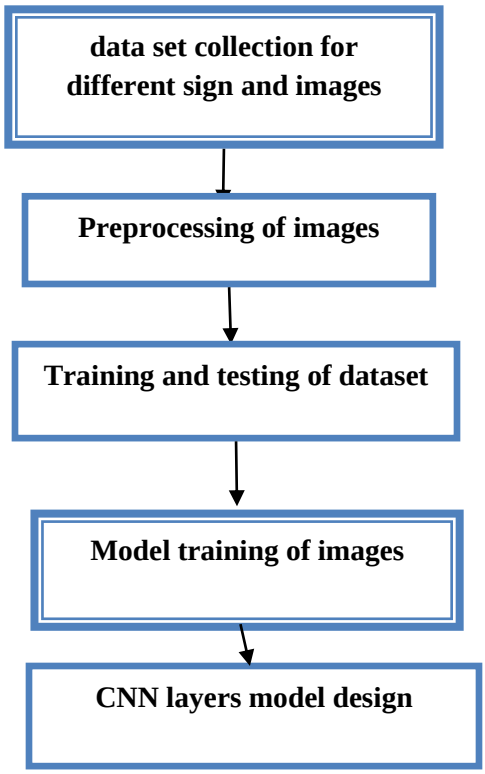
Technology that is highly reliable functions with the same or similar efficiency after extensive use.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

A data flow diagram (DFD) maps out the flow of information for any process or it uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.DFD for the given project is presented as shown below,



5.2 SOLUTION & TECHNICAL ARCHITECTURE:

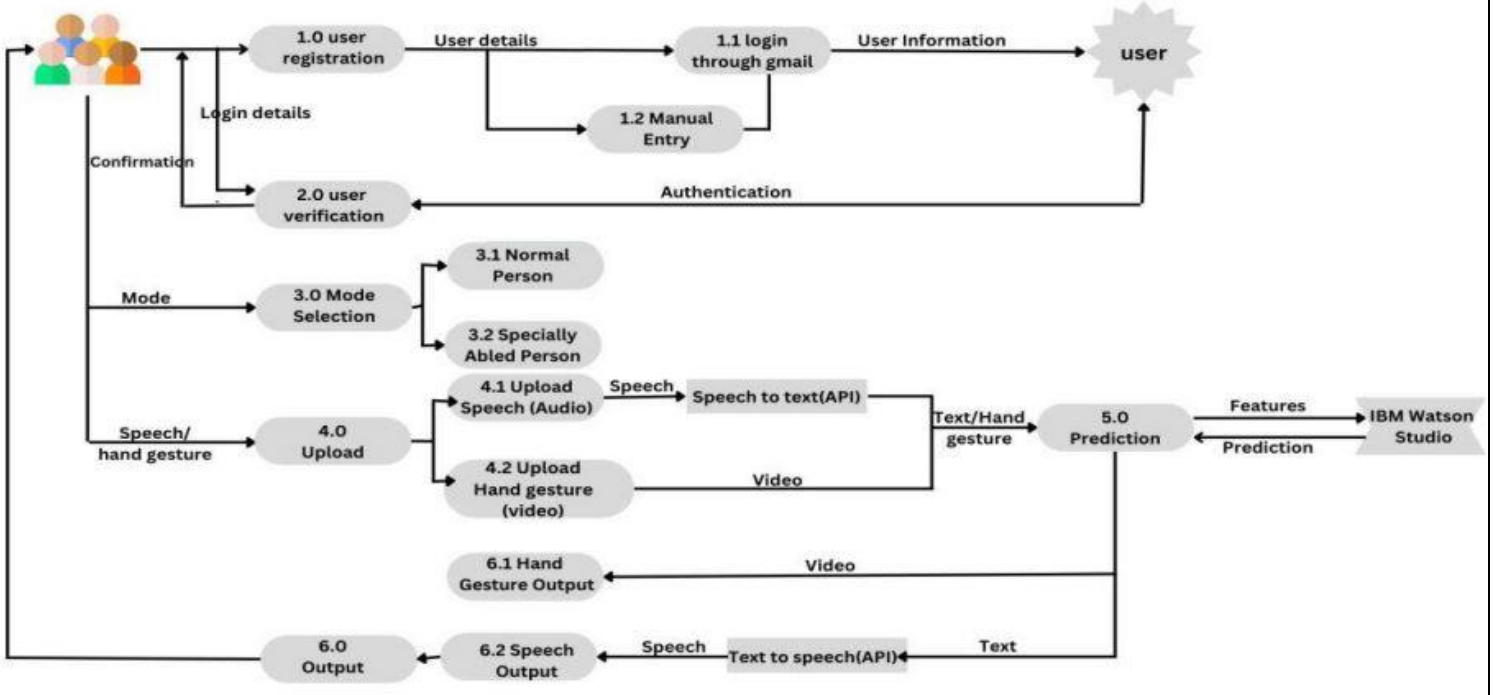
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• Find the best tech solution to solve existing business problems.

• Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

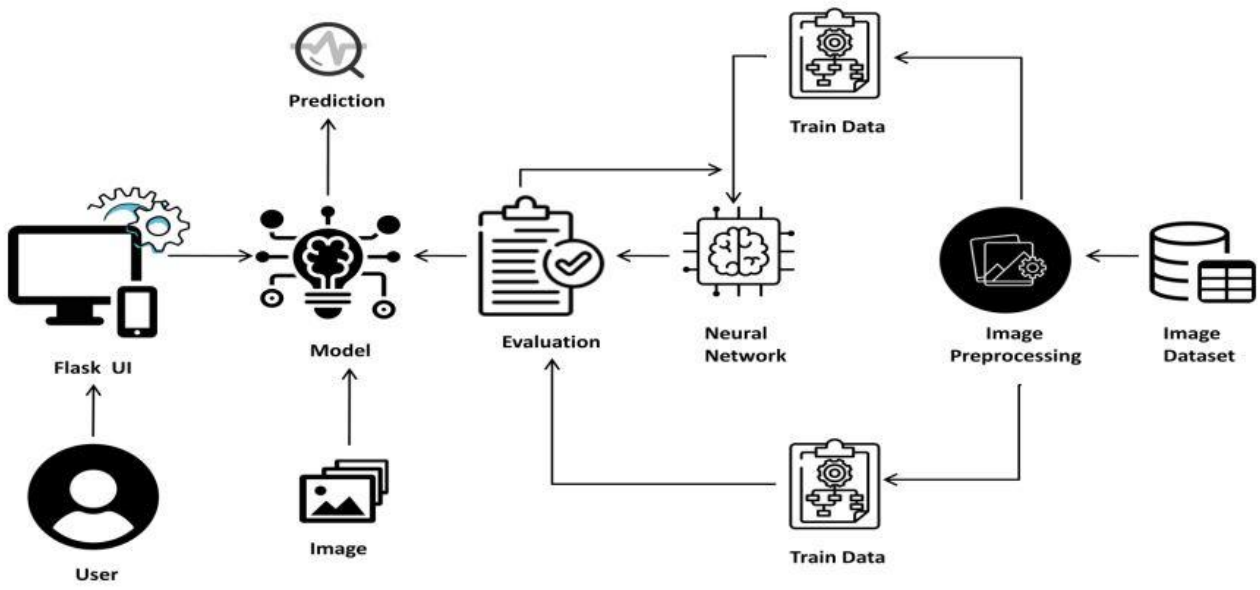
• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.



TECHNICAL ARCHITECTURE:

Technical Architecture (TA) involves development of technical blueprint with regard to the arrangement, interaction and interdependence of all elements so that system –relevant requirements are met.



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Specially Abled Person)	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 and select the mode of usage.	High	Sprint-1
		USN-2	As a user, I can register for the Application through Gmail.		High	Sprint-1
	Confirmation	USN-3	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm.	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	Credentials has to be matched	Medium	Sprint-2
	Mode Selection	USN-5	As a user, I will be prompted to select the mode of communication and I will select the specially abled mode (Gesture to Speech)	Either of the modes has to be chosen for further processing	High	Sprint-3
	Video Capturing	USN-6	As a user of this mode I will capture my hand gesture as video	Minimum video quality criteria has to be met	High	Sprint-1
	Gesture interpretation	USN-7	As a user of this mode, I will be able to receive and interpret the translated gestures from the other end.	Must be a valid gesture	Low	Sprint-1

Customer (Normal Person)	**Registration**	USN-8	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account and select the mode of usage	High	Sprint-1
		USN-9	As a user, I can register for the application through Gmail		High	Sprint-1
	Confirmation	USN-10	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-2
	Login	USN-11	As a user, I can log into the application by entering email & password	Credentials has to be matched	Medium	Sprint-2
	Mode Selection	USN-12	As a user, I will be prompted to select the mode of communication and I will select the specially abled mode (Gesture to Speech)	Either of the modes has to be chosen for further processing	High	Sprint-3
	Speech Recording	USN-13	As a user of this mode I will record the speech in order to convert it into gesture	Minimum audio quality criteria have to be met	High	Sprint-1
	Speech recognition	USN-14	As a user of this mode, I will be able to receive and interpret the translated speech from the other end.	The words must be a recognizable	Low	Sprint-1
Administrator	**Application monitoring and controlling**	USN-15	As an admin, I will be responsible for controlling the user activities and further upgradations of the application	Admin level privilege	Medium	Sprint-3

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect Dataset .	9	High	AGATH CLAFIO.P ESWAR MARKX.P.K
		USN-2	Image preprocessing	8	Medium	APSARAA.S EZHILARASHI.T.S
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	EZHILARASHI.T.S AGATH CLAFIO.P
		USN-4	Training the image classification model using CNN	7	Medium	APSARAA.S ESWAR MARKX.P.K
Sprint-3	Training andTesting	USN-5	Training the model and testing the model's performance	9	High	AGATH CLAFIO.P APSARAA.S
Sprint-4	Implementation ofthe application	USN-6	Converting the input sign language images into English alphabets	8	Medium	EZHILARASHI.T.S ESWAR MARKX.P.K

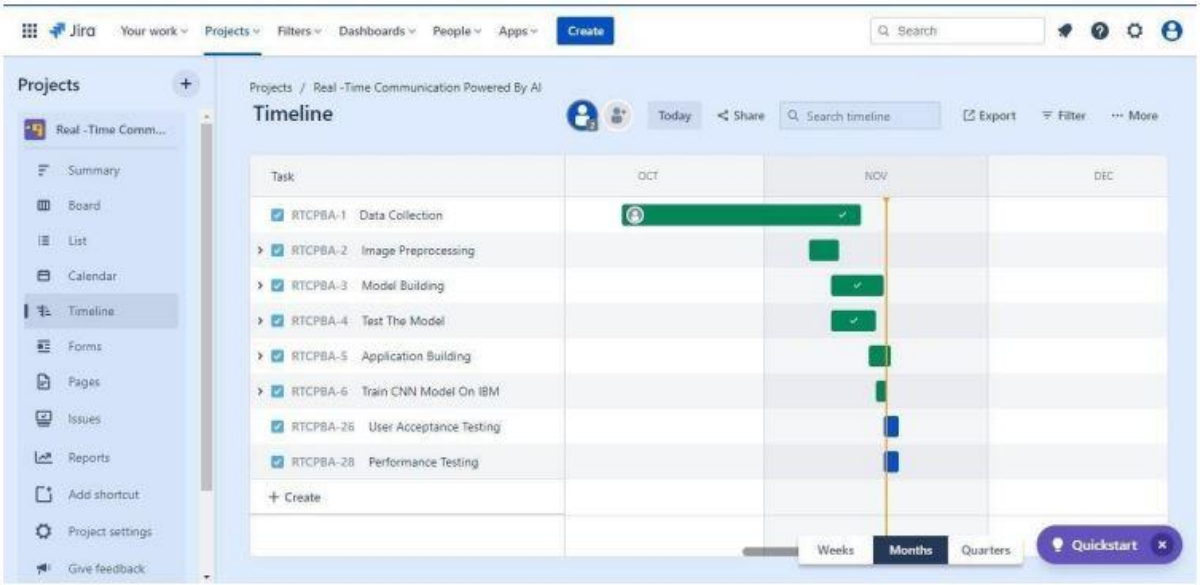
6.2 SPRINT DELIVERY SCHEDULE:

The estimated time of delivery required for each sprint is mentioned clearly in the following tabular column.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	04 Nov 2022	5	04 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	11 Nov 2022	7	11 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	6	18 Nov 2022

6.3 REPORTS FROM JIRA:

JIRA Software is part of a family of products designed to help teams of all types manage work. It can be accepted as an task scheduler which describes the To-do, In-Progress and Done tasks.



CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE 1

Machine Learning cannot be extensively used for feature extraction because its algorithm cannot handle high dimensional data and also it is one of the big challenges in object detection, image classification, etc. Deep Learning inspired from how biological neural network have been playing a key role in the field of machine learning. As we are aware of the fact that images are high dimensional vectors. It would take a huge number of parameters to characterize the network, if we take flatten image feature vector 1-D. To address this problem (of taking long vector), CNN are proposed to reduce the number of parameters and adapting the network architecture specifically to vision tasks. CNN works well on tasks. Convolution Neural Network basically includes set of layers each having their own functionalities. CNN works by extracting features from images.

There is no need for manual feature extraction. Features are trained while the network trains on the set of images.

CNNs key components includes:

Convolutional layer:

Consist of grouped neuron in a rectangular grid. It is the application of Filter to an input that results in an activation. Repeated activation of it results into a feature map, which indicating the locations and strength of a detected features in an input that is image.

Activation function:

Decides whether a neuron should fire an output or not and it is also used to increase the non-linearity into the output of the neuron.

Pooling layers :

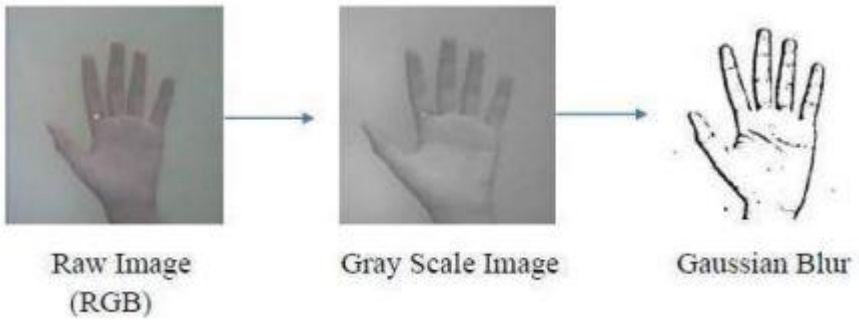
Are present after a single or a set of convolution layers. Its function is to down sample the detection of features in feature map.

Fully connected layer:

Dense layer which is the final learning phase where classification takes place. Every node of it is connected to every nodes of previous layer.

Dataset Generation:

For this project we tried to make our own dataset for the ASL language but due to lack of resources we were unable to do so. Then we find out the already existing datasets that matched our requirements. All we could find were the datasets in the form of RGB values. Hence, we decided to transform it into our required form. By using batch mode transformation/data augmentation we try to convert it into Gray scale Image as shown below:

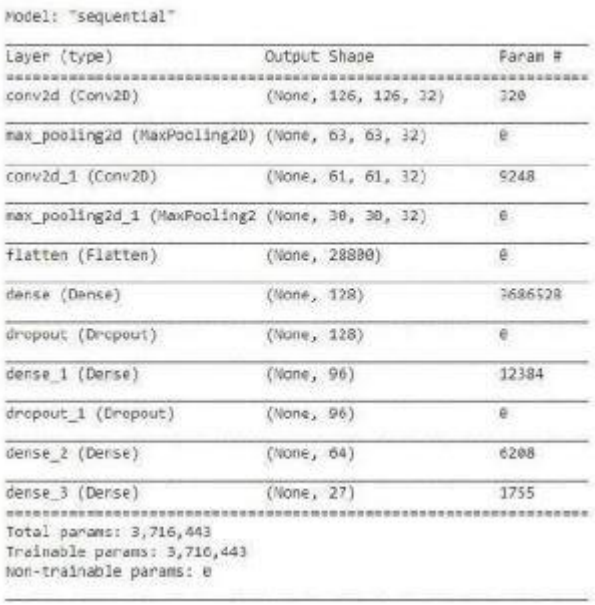


CNN architecture sum

It is one of the most important part for our network i.e. in defining our required architecture. A brief explanation of CNN is provided below along with a figure of our model architecture summary:

- Starts with an input image.
- Applies many different filters to obtain a feature map.

- Applies a RELU function to increase non-linearity.
- Applies pooling layer to each future map.
- Flattening the pooled images into one long vector.
- Inputs the vector into fully connected ANN.
- Dropout is also used to mitigate the over fitting.
- Final fully connected layer provides the voting of the classes.
- Trains through forward propagation and back propagation for many epochs



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 63, 63, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 61, 61, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 30, 30, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 28800)             0
_________________________________________________________________
dense (Dense)                (None, 128)               3686528
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 96)                12384
_________________________________________________________________
dropout_1 (Dropout)          (None, 96)                0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                6208
_________________________________________________________________
dense_3 (Dense)              (None, 27)                1755
=================================================================
Total params: 3,716,443
Trainable params: 3,716,443
Non-trainable params: 0
_________________________________________________________________
```

Training our network:

After successfully completing all the steps of constructing our network. Now the next step is to train our network. To train the model effectively we have set epochs and other necessary required configurations to train it effectively.

Saving model:

The metadata file (or model.json) in a Common Data Model folder describes the data in the folder, metadata and location, as well as how the file was generated and by which data producer. Metadata summarizes basic information about data, which

can make finding and working with particular instances of data easier. JSON (JavaScript Object Notation) is a popular data format used for representing structured data. So, keeping in the mind regarding usability of data model we have saved our model as json file.

Gesture Classification:

Our approach uses two layers of algorithm to predict the finalsymbol of the user.

Algorithm Layer 1:

- Apply gaussian blur filter and threshold to the frame taken with Open CV to go the processed image after feature extraction.
- This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.
- Space between the words is considered using the blank symbol.

Algorithm Layer 2:

- We detect various sets of symbols which show similar results on getting detected.
- We then classify between those sets using classifiers made for those sets only.

Activation Layer :

We have used ReLu (Rectified Linear Unit) in each of the layers (convolutional as well as fully connected neurons). ReLu calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time.

Pooling Layer :

We apply Max pooling to the input image with a pool size of (2, 2) with relu activation function. This reduces the amount of parameters thus lessening the computation cost and reduces over fitting.

Dropout Layer:

The problem of over fitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn't perform well when given new examples. This layer "drops out" a random set of activations in that layer by setting them to zero. The network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out.

Considering the above features , the following main application code is generated.

% packages

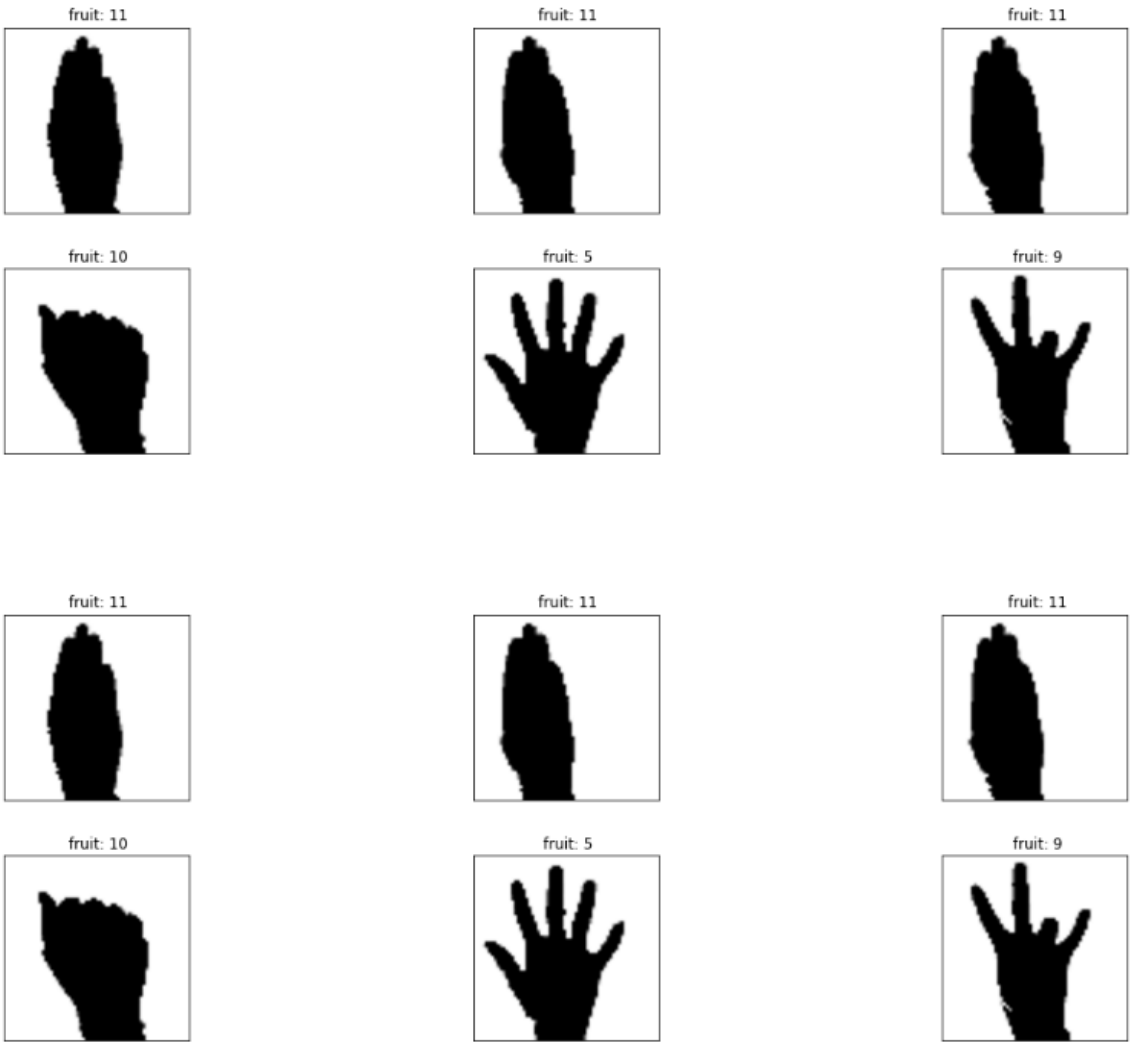
```
import os
import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

from tensorflow import keras
from keras.models import Sequential
from keras.layers import Conv2D,MaxPooling2D,Dense,Flatten,Dropout
from tensorflow.keras.layers import BatchNormalization
print("Loaded all libraries")
```

%data preprocessing

```
fpath = '/content/drive/My Drive/test_set'
random_seed = 42

categories = os.listdir(fpath)
categories = categories[:20]
print("List of categories = ",categories,"\n\nNo. of categories = ", len(c
ategories))
def load_images_and_labels(categories):
    img_lst=[]
    labels=[]
    for index, category in enumerate(categories):
        for image_name in os.listdir(fpath+"/"+category):
            img = cv2.imread(fpath+"/"+category+"/"+image_name)
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

            img_array = Image.fromarray(img, 'RGB')

            #resize image to 227 x 227 because the input image resolution
for AlexNet is 227 x 227
            resized_img = img_array.resize((227, 227))

            img_lst.append(np.array(resized_img))

            labels.append(index)
    return img_lst, labels

images, labels = load_images_and_labels(categories)
```

```
print("No. of images loaded = ",len(images),"\nNo. of labels loaded = ",le
n(labels))
print(type(images),type(labels))
```

CNN model design



7.2 FEATURE 2

Implementation:

- Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold, we print the letter and add it to the current string.
- Otherwise, we clear the current dictionary which has the count of detections of present symbol to avoid the probability of a wrong letter getting predicted.
- Whenever the count of a blank (plain background) detected exceeds a specific value and if the current buffer is empty no spaces are detected.
- In other case it predicts the end of word by printing a space and the current gets appended to the sentence below.

Autocorrective Feature:

A python library Hunspell_suggest is used to suggest correct alternatives for each(incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence. This helps in reducing mistakes committed in spellings and assists in predicting complex words.

Training and Testing:

We convert our input images (RGB) into grayscale and apply gaussian blur to remove unnecessary noise. Then we have applied adaptive threshold to extract our hand from the background and resize our images to 128 x 128. We feed the input images after pre-processing to our model for training and testing after applying all the operations mentioned above. The prediction layer estimates how likely the image will fall under one of the classes. So, the output is normalized between 0 and 1 and such that the sum of each values in each class equals to 1. We have achieved this using SoftMax function..

User Interface:

UI stands for user interface and is the point of communication between human and the computer. The function of our UI is to provide users with an intuitive interaction and support that manages to provide a solution for the desired task. We have provided user input control's which are the interactive component of interface. It basically includes space for input gesture, various other spaces for character, word, sentence and buttons for audio output, backspace, reset and also for the suggestions to be picked up from the screen. We are providing both text and audio output on the same platform which makes a user to understand the input gestures more effectively.

As you can see the UI window below. At the top there is a title shown and exactly below it there is a space provided for gestures to displayed. And the on the right-hand side of it three buttons are given with the distinctive functionality which are as follow i.e., pressing the audio button one can have an audio output; backspace button to correct the wrong word being predicted; reset button to have a blank screen for new gestures to be classified. After that downside in the UI screen there are spaces given for result to be obtained i.e., for character, word, sentence. And below it few more spaces are provided for the suggestions to presented before a use to help me picking up the correct word inorder to communicate effectively.

CHAPTER 8

TESTING

8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.1 TYPES OF TESTS

8.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.1.5 White Box Testing

White Box Testing is a testing in which the software tester knows the inner workings, structure, and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure, or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

8.2 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

8.2.1 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

8.2.2 Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages, and responses must not be delayed.

8.2.3 Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects were encountered.

8.4 Acceptance Testing

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.

Test Results: All the test cases mentioned above passed successfully. No defects were encountered.

9. RESULTS

9.1 PERFORMANCE METRICS

Evaluating the performance of a CNN model is one of the important steps while building an effective CNN model. To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics. These performance metrics help us understand how well our model has performed for the given data. In this way, we can improve the model's performance by tuning the hyperparameters. Each CNN model aims to generalize well on unseen/new data, and performance metrics help determine how well the model generalizes on the new dataset.

In a classification problem, the category or classes of data is identified based on training data. The model learns from the given dataset and then classifies the new data or groups based on the training. It predicts class labels as the output, such as Yes or No, 0 or 1, Spam or Not Spam, etc. To evaluate the performance of a classification model, different metrics are used, and some of them are as follows:

- Accuracy
- Confusion Matrix
- Precision
- Recall
- F-Score
- AUC(Area Under the Curve)-ROC

CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

I. REAL-TIMEIME FUNCTIONING

The output of the sign language will be displayed in text form in real-time. This makes the system more efficient and hence communication the hearing and speech impaired people easier. The images captured through the webcam are compared and the result of the comparison is displayed at the same time. Thus this feature of the system makes communication very simple and delay-free.

II. PORTABLE

When this entire project is implemented on a Raspberry Pie computer, which is a very small yet powerful computer, the entire system becomes portable and can be taken anywhere. This feature facilitates the user to take the system anywhere and everywhere and overcomes the barrier of restricting him/herself to communicate without a desktop or laptop.

III. NO NEED FOR CALIBRATION.

DISADVANTAGES:

- Sign language requires the use of hands to make gestures. This can be a problem for people who do not have full use of their hands. Even seemingly manageable disabilities such as Parkinson's or arthritis can be a major problem for people who must communicate using sign language.
- Cost and is difficult to be used commercially.

CHAPTER 11
CONCLUSIONS

This work presented the design of a complete end-to-end embedded system, which can accurately recognize the hand gestures of the low-resolution thermal images of 32×32 pixels. A thermal dataset of 3200 images was curated and each sign language digit has 320 hand gestures of thermal images. We have also developed a lightweight convolutional neural network to provide high accuracy and the need for having ing high-performance computing environment. The designed system has achieved an accuracy of 99.52% on the test dataset with the added advantage of accuracy being invariable to background lighting conditions as it is based on thermal imaging. The developed system has shown that thermal imaging is well suited for hand gesture recognition in dark light conditions Flask based framework is implemented for sign language recognition.

CHAPTER 12

FUTURE SCOPES

In future work we designed the real rasperpi based real time implementation is implemented forget the product

CHAPTER 13

APPENDIXES

SOURCE CODE:

Model design

```
model=Sequential()

#1 conv layer
model.add(Conv2D(filters=96,kernel_size=(11,11),strides=(4,4),padding="val
id",activation="relu",input_shape=(227,227,3)))

#1 max pool layer
model.add(MaxPooling2D(pool_size=(3,3),strides=(2,2)))

model.add(BatchNormalization())

#2 conv layer
model.add(Conv2D(filters=256,kernel_size=(5,5),strides=(1,1),padding="vali
d",activation="relu"))

#2 max pool layer
model.add(MaxPooling2D(pool_size=(3,3),strides=(2,2)))

model.add(BatchNormalization())

#3 conv layer
model.add(Conv2D(filters=384,kernel_size=(3,3),strides=(1,1),padding="vali
d",activation="relu"))

#4 Conv layer
model.add(Conv2D(filters=384,kernel_size=(3,3),strides=(1,1),padding="vali
d",activation="relu"))

#5 conv layer
model.add(Conv2D(filters=256,kernel_size=(3,3),strides=(1,1),padding="vali
d",activation="relu"))

#3 max pool layer
model.add(MaxPooling2D(pool_size=(3,3),strides=(2,2)))

model.add(BatchNormalization())
```

```
model.add(Flatten())

#1 dense layer
model.add(Dense(4096,input_shape=(227,227,3),activation="relu"))

model.add(Dropout(0.4))

model.add(BatchNormalization())

#2 dense layer
model.add(Dense(4096,activation="relu"))

model.add(Dropout(0.4))

model.add(BatchNormalization())

#3 dense layer
model.add(Dense(1000,activation="relu"))

model.add(Dropout(0.4))

model.add(BatchNormalization())

#output layer
model.add(Dense(20,activation="softmax"))

model.summary()



model.fit(x_train, y_train, epochs=15)



%predict results
pred = model.predict(x_test)

pred.shape
```

GITHUB :

https://github.com/IBM-EPBL/IBM-Project-47016-1660795866.git

PROJECT DEMO LINK :

https://youtu.be/ypIlAsGrFrs