REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

TEAM ID	**PNT2022TMID30828**
TEAM MEMBERS	**VALARMATHI.S -620119104105** **SRIHARINI.G -620119104094** **SARANYA.M -620119104087** **VINITHA.K -620119104115** **SUREKA.D -620119104098**

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.

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
2	Design of a Communication System using Sign Language aid for	Our goal is to design a human computer interface a system that can	❖ Feature Extraction ❖ Sign to text and Speech Conversion	❖ Blob Detection ❖ Skin color recognition ❖ Template Matching	Hand gestures of deaf people by normal peoples this system is proposed and it gives output in the form of sound./A

	Differently Abled Peoples.	accurately identify the language of the deaf and dumb.	❖ Image preprocessing and segmentation.		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 TO AID UNEDUCATED	In this paper, we introduce an integrated android	❖ Sign language keyboard app	❖ Eclipse ❖ SQL Lite ❖ Java	Offer a great tool for parents to teach their deaf and dumb kids And Introduce Sign language

	DEAF DUMB PEOPLE	application to blend uneducated Deaf Dumb people within society, and help them to communicate with normal people.			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.

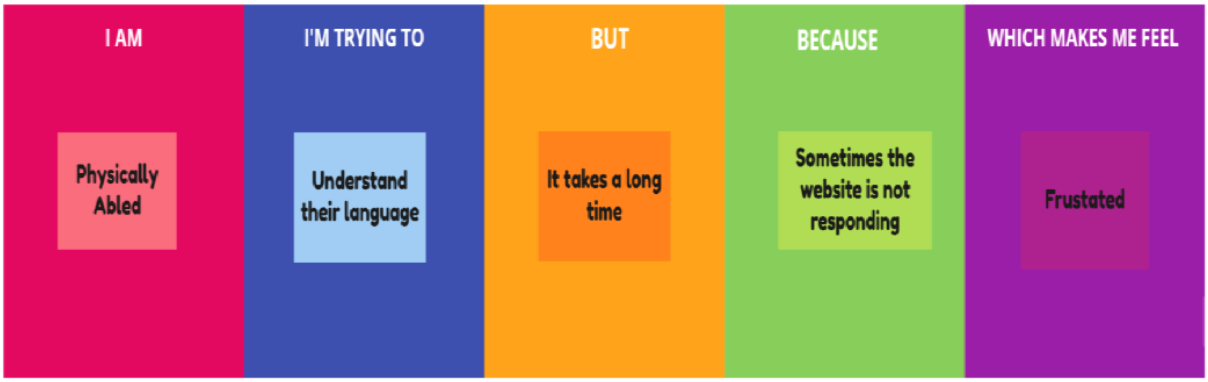
[2] Q. Ye, "A Novel FPGA Control Process with AI-Fuzzy VCR Algorithm," *2010 International Conference on Electrical and Control Engineering*, 2010, pp. 190-193, doi: 10.1109/iCECE.2010.53.

[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.

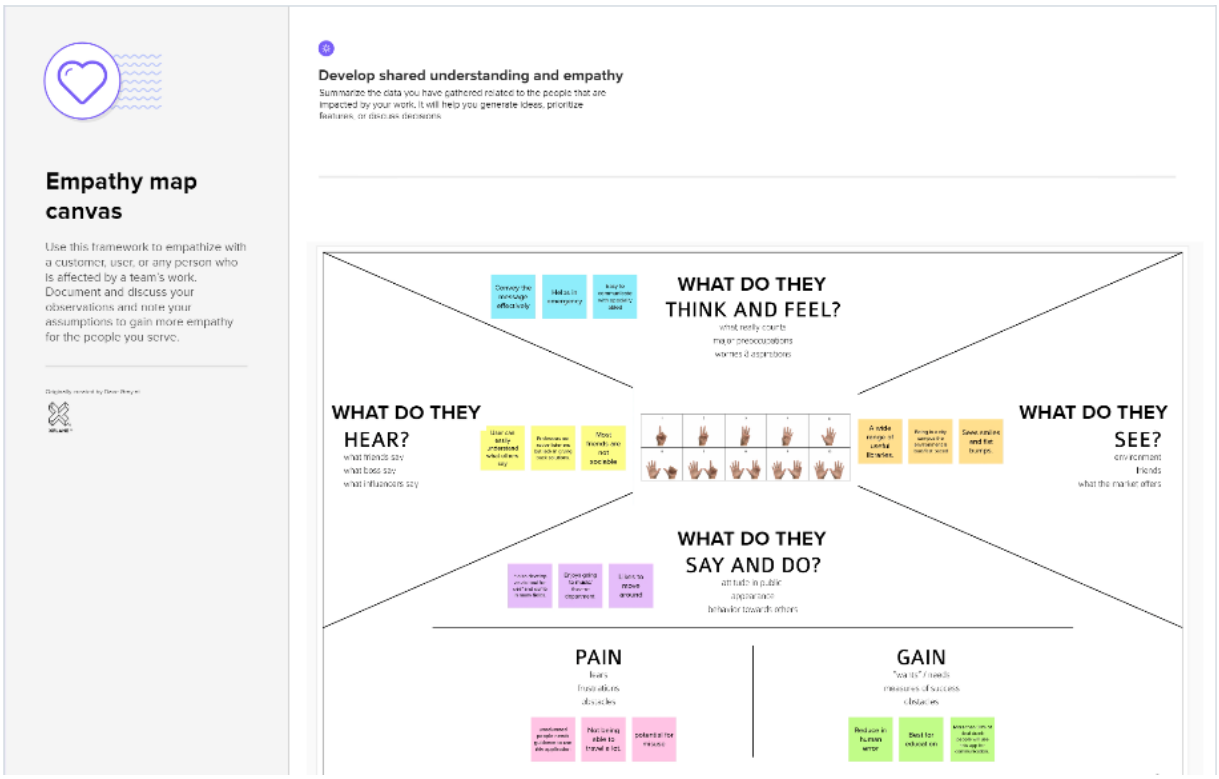
[5] R. H. Rachmadi, R. Azzahra, R. A. Darmawan, P. A. Nigo and N. N. Qomariyah, "Developing AI Bots with Minimax Algorithm for Surakarta Board Game," *2021 International Conference on ICT for Smart Society (ICISS)*, 2021, pp. 1-6, doi: 10.1109/ICISS53185.2021.9533206.

2.3 PROBLEM STATEMENT DEFINITION:

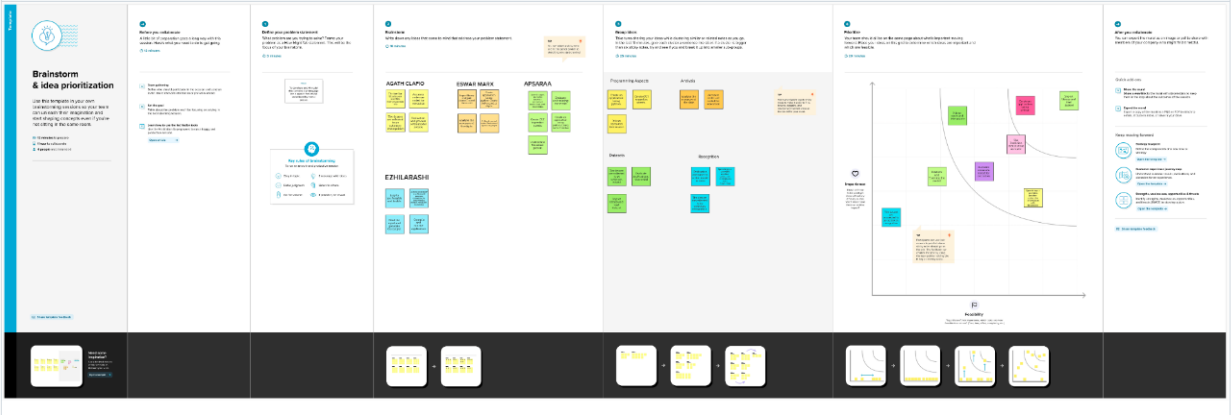


3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION & BRAINSTORMING:



PROBLEM STATEMENT

Communication is the only medium by which we can share our thoughts or convey the message but communications between deafmute and a normal person has always ben 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. Problem: Vedha has difficulty in hearing. He uses sign language to communicate with others. But he can't able to communicate with normal people who don't understand sign language. Solution: 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 ,the system enhances the user friendly experience. Problem: Ram is a dumb by birth. He uses sign language to communicate with others. But he can't able to communicate with normal people who don't understand sign language. Solution: To create a app for understanding sign language and convert into Speech signal as output for normal people

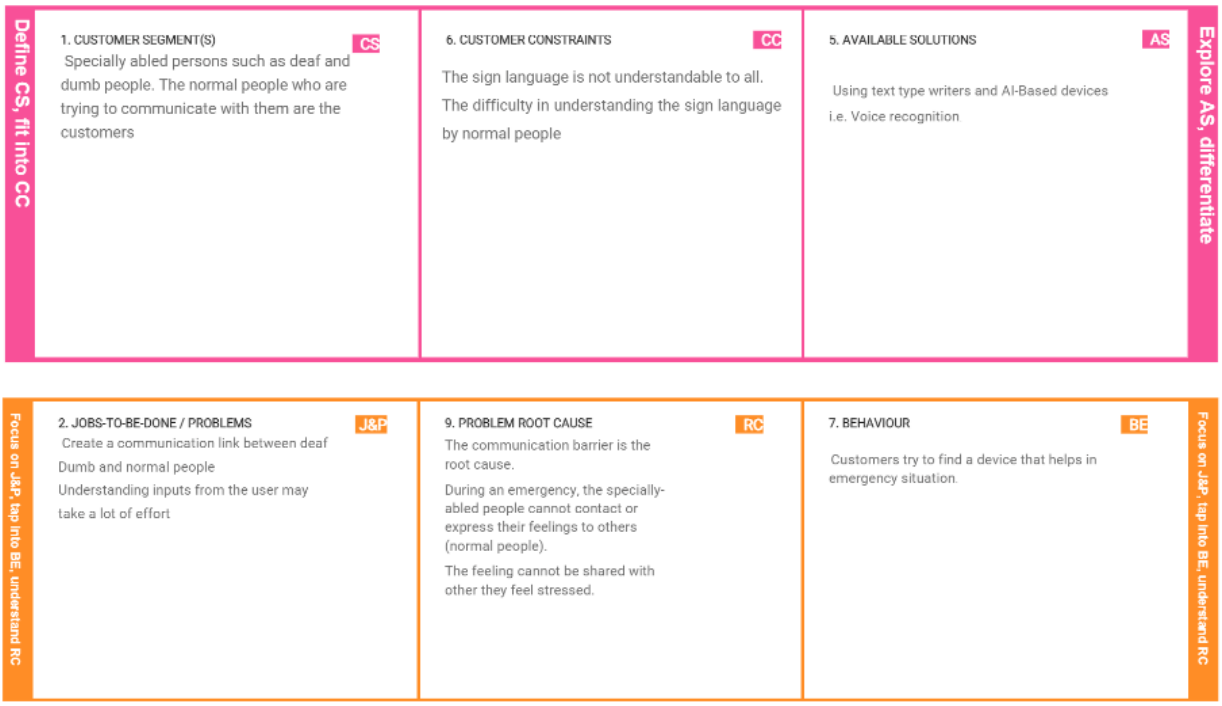
3.3 PROPOSED SOLUTION:

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:



4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENT:

● Here , the Desktop along with the 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 people especially should be able to see the translation of signs to text format.

● Dumb person should be able to understand the conversion of text into voice mode.

● Normal users should be able to understand the corresponding information conveyed by disabled through sign language

Default Operation:

● Users of the app face the camera and perform the concerned hand sign to convey information. ● System/Desktop analyses the sign made by the user.

● Once analysis gets finished, then the concerned signs together are shown as a text based and also through voice

Unexpected Operations:

● Desktop indicates that the user's hand sign is not within the frame or in Region of Interest(ROI).

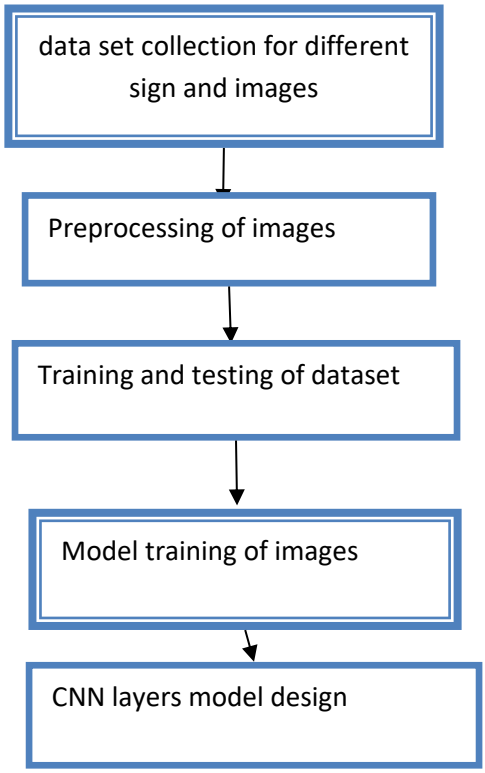
1. Users of the app show the hand sign towards the camer
2. Desktop shows that the sign is not within ROI.
3. Still User , make sure to present his/her sign within frame. 4. At last, Desktop finally detects the hand sign.

● Signs are not recognized

1. Excepts the signs that are trained and included in the dataset, the Desktop will never detect the sign rather than this.
2. User Performs the sign and sees that after 50ms, the concerned letter occupies the space of text.

5. PROJECT DESIGN:

5.1 DATA FLOW DIAGRAMS:



5.2 SOLUTION & TECHNICAL ARCHITECTURE:

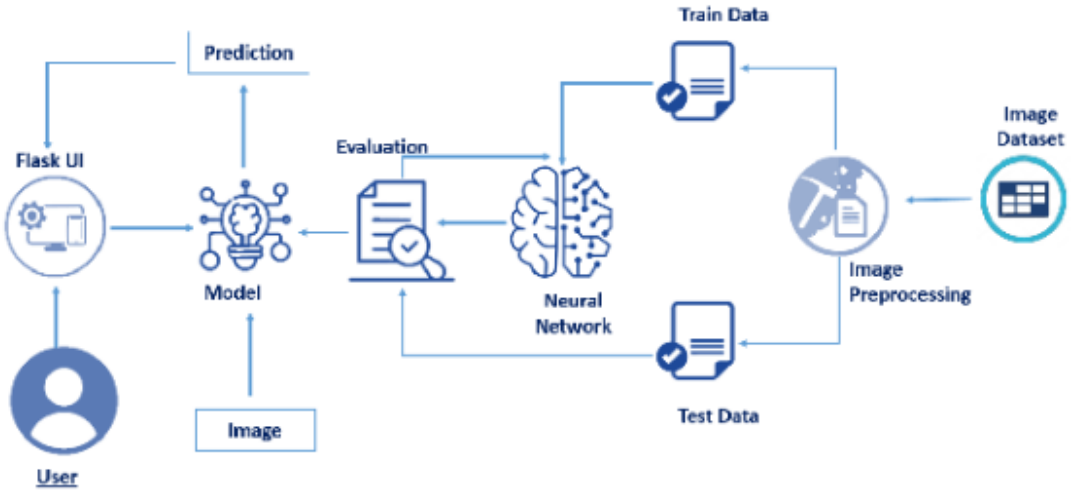


Figure: Technical Architecture design

SOLUTION ARCHITECTURE:

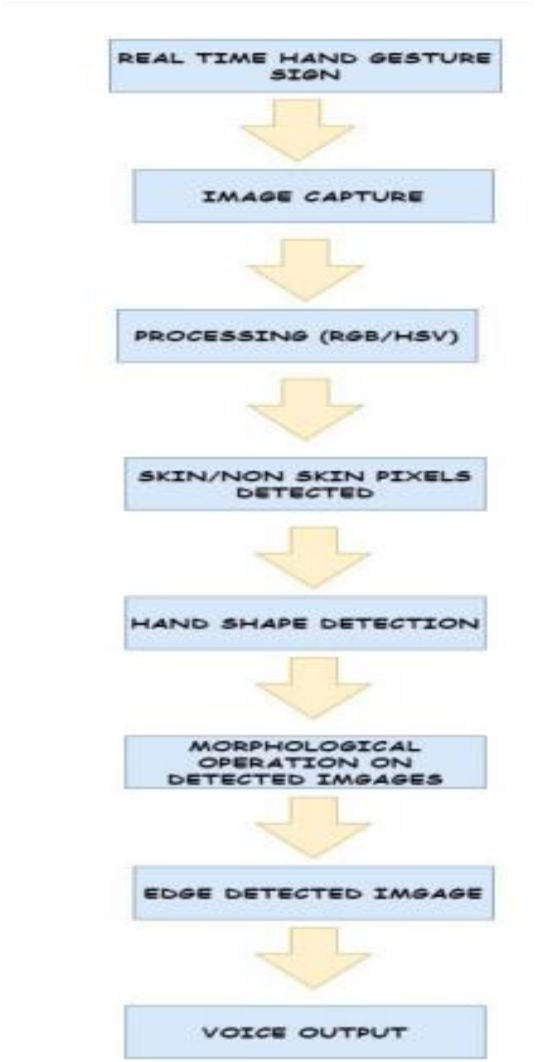
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

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

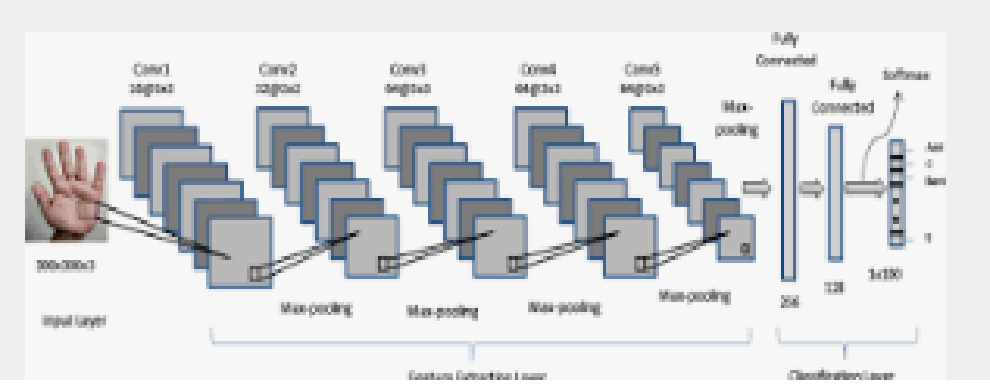
• Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

• Define features, development phases, and solution requirements.

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



TECHNICAL ARCHITECTURE:



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer *(Desktop 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	Sprint-1
	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	USN-3	As a user, I can select options in dashboard.	I can select options in dashboard.		
Customer *(Desktop user)*	**Main page**	USN-4	As a User, **I can enter the web page once clicked, which provides be the Guidelines** to use the app	I can **enter the web page** once clicked.	Medium	Sprint-1
Customer (Desktop user)	**Guidelines**	USN-5	As a User, **I can give a read through the guidelines to understand the functioning of the app.**	I can **give a read through the guidelines.**	Medium	Sprint-1
Customer *(Desktop user)*	**Convert Sign**	USN-6	As a User, **I can click the button Convert sign**, which directs me towards the Main screen	I can **click the button Convert sign and it direct me to main screen.**	Medium	Sprint-2
Customer *(Desktop user)*	**Camera (Hand movement detection)**	USN-7	As a User, **I can show my hand sign towards the camera which converts them into text manner.**	**I can show my hand sign** towards the camera accurately.	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING:

6.1 SPRINT PLANNING & ESTIMATION:

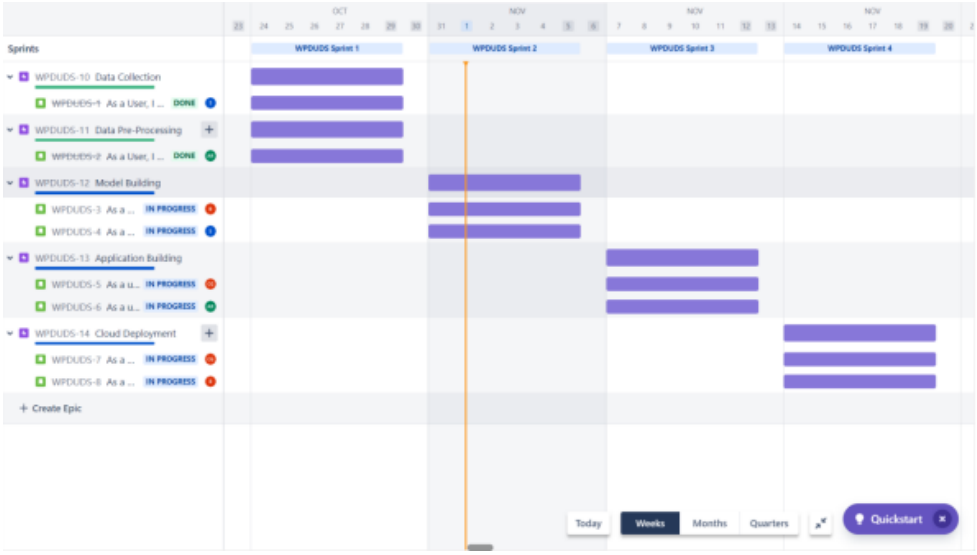
Milestone Activity Plan.

Milestone	Function (Epic)	Milestone Story Number	Story / Task
Milestone 1	Data collection	M1	we're collecting dataset for building our project and creating two folders, one for training and another one for testing.
Milestone 2	Image preprocessing	M2	Importing image data generator libraries and applying image data generator functionality to train the test set.
Milestone 3	Model building	M3	Importing the model building libraries, Initializing the model, Adding Convolution layers, Adding the Pooling layers, Adding the Flatten layers, Adding Dense layers, Compiling the model Fit and Save the model.
Milestone 4	Testing the model	M4	Import the packages first. Then we save the model and Load the test image, preprocess it and predict it.
Milestone 5	Application layer	M5	Build the flask application and the HTML pages.
Milestone 6	Train CNN model	M6	Register for IBM Cloud and train Image Classification Model.
Milestone 7	Final result	M7	To ensure all the activities and resulting the final output.

6.2 SPRINT DELIVERY SCHEDULE:

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	VALARMATHI.S VINITHA.K
Sprint-1		USN-2	Image preprocessing	8	Medium	SARANYA.M SUREKA.D
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	SRIHARINI.G, VALARMATHI.S
Sprint-2		USN-4	Training the image classification model using CNN	7	Medium	VINITHA.K SARANYA.M
Sprint-3	Training and Testing	USN-5	Training the model and testing the model's performance	9	High	SUREKA.D VALARMATHI.S
Sprint-4	Implementation of the application	USN-6	Converting the input sign language images into English alphabets	8	Medium	SRIHARINI.G, VINITHA.K

6.3 REPORTS FROM JIRA:



7. CODING & SOLUTIONING:

% 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

fruit: 11

fruit: 11

fruit: 11

fruit: 10

fruit: 5

fruit: 9

fruit: 11

fruit: 11

fruit: 11

fruit: 10

fruit: 5

fruit: 9

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 identify 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 in which the software tester has knowledge of 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, as 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 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 encountered.

9. RESULTS

10 ADVANTAGES & DISADVANTAGES:

- Its user friendly
- Good accuracy
- High predicting result

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 high performance computing environment. The designed system has achieved an accuracy of 99.52% on the test dataset with an 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 the hand gesture recognition in dark light conditions Flask based framework is implemented for sign language recognition

12 FUTURE SCOPES:

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

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
```