Real-Time Communication System Powered by AI for Specially Abled

TEAM ID: PNT2022TMID18723

- TEAM LEADER : MATHANGI.P.R
- TEAM MEMBER 1: KAVYA P
- TEAM MEMBER 2: MEKHA R
- TEAM MEMBER 3: SABITHA V

CONTENTS

1. INTRODUCTION

1.1 Project Overview

Individuals get to know each other by sharing their thoughts, contemplations, and encounters with people around them. There are various ways of achieving this, the best of which is the endowment of "Discourse." Everybody can convincingly move their contemplations and see each other through discourse. It will be treacherous assuming that we disregard the individuals who are denied this precious gift: the not too sharp. In such cases, the human hand has stayed the favored strategy for correspondence.

1.2 Purpose

The project's purpose is to create a system that translates sign language into a human understandable language so that ordinary people may understand it.

2. LITERATURE SURVEY

2.1 Existing problem

Interchanges between hard of hearing quiet and a typical individual has forever been a difficult errand. It is truly challenging for quiet individuals to pass their message on to typical individuals. Since ordinary individuals are not prepared close by communication via gestures.

Just uncommonly abled individuals are shown gesture-based communication and the normal individual is ignorant its functioning causing a correspondence hole. Under crisis circumstances, it is considerably more-hard for uniquely abled individuals to find support. Non-Crisis typical conditions can likewise be difficult for them to explore requiring exceptional help.

2.2 References

1. Upendran, S., and Thamizharasi, A., "American Sign Language interpreter system for deaf and dumb individuals", In the Proceedings of the International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1477-1481, 2014

2. Rajamohan, A., Hemavathy, R., and Dhanalakshmi, M., "Deaf-Mute Communication Interpreter", International Journal of Scientific Engineering and Technology, Vol.2, No.5, pp.336-341, 2013.

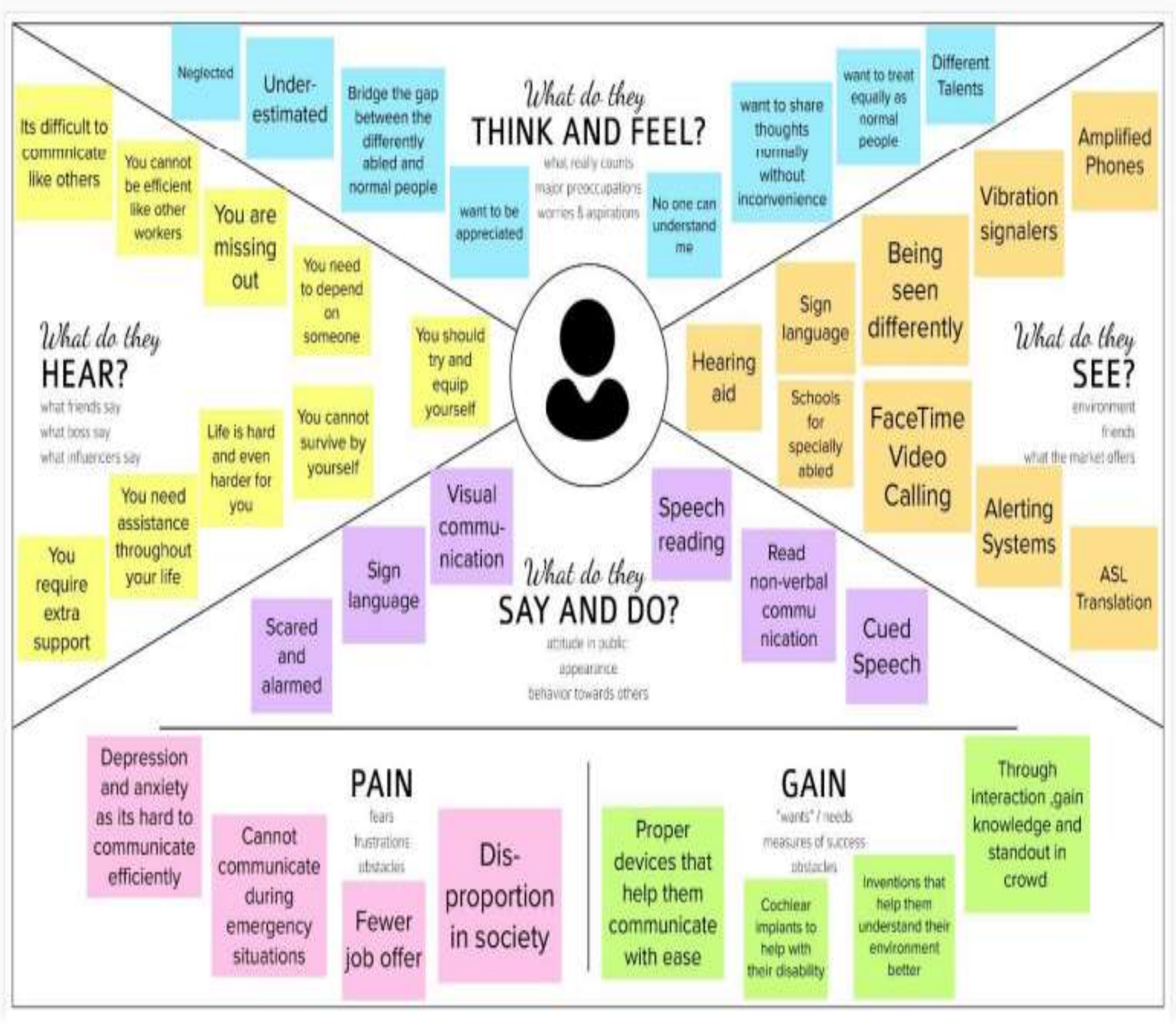
3. Verma, P., Shimi S. L. and Priyadarshani, R., "Design of Communication Interpreter for Deaf and Dumb Person", Vol.4, no.1, 2013.

2.3 Problem Statement Definition

Just uncommonly abled individuals are shown gesture-based communication and the normal individual is ignorant its functioning causing a correspondence hole. Under crisis circumstances, it is significantly more hard for uniquely abled individuals to find support. 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. Non-Crisis typical conditions can likewise be difficult for them to explore requiring unique help. Interchanges between hard of hearing quiet and a typical individual has forever been a difficult errand. It is truly challenging for quiet individuals to pass their message on to typical individuals. Since typical individuals are not prepared close by gesture-based communication.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy map canvas


What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Neglected
Under-estimated
Bridge the gap between the differently abled and normal people
want to be appreciated
No one can understand me
want to share thoughts normally without inconvenience
want to treat equally as normal people
Different Talents
What do they HEAR?
what friends say
what boss say
what influencers say
Its difficult to commnicate like others
You cannot be efficient like other workers
You are missing out
You need to depend on someone
You should try and equip yourself
Life is hard and even harder for you
You cannot survive by yourself
You need assistance throughout your life
You require extra support
What do they SEE?
environment
friends
what the market offers
Amplified Phones
Vibration signalers
Being seen differently
Sign language
Hearing aid
Schools for specially abled
FaceTime Video Calling
Alerting Systems
ASL Translation
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Visual commu-nication
Sign language
Scared and alarmed
Speech reading
Read non-verbal commu nication
Cued Speech
PAIN
fears
frustrations
obstacles
Depression and anxiety as its hard to communicate efficiently
Cannot communicate during emergency situations
Fewer job offer
Dis-proportion in society
GAIN
"wants" / needs
measures of success
obstacles
Proper devices that help them communicate with ease
Cochlear implants to help with their disability
Inventions that help them understand their environment better
Through interaction ,gain knowledge and standout in crowd

3.2 Ideation and Brainstorming

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

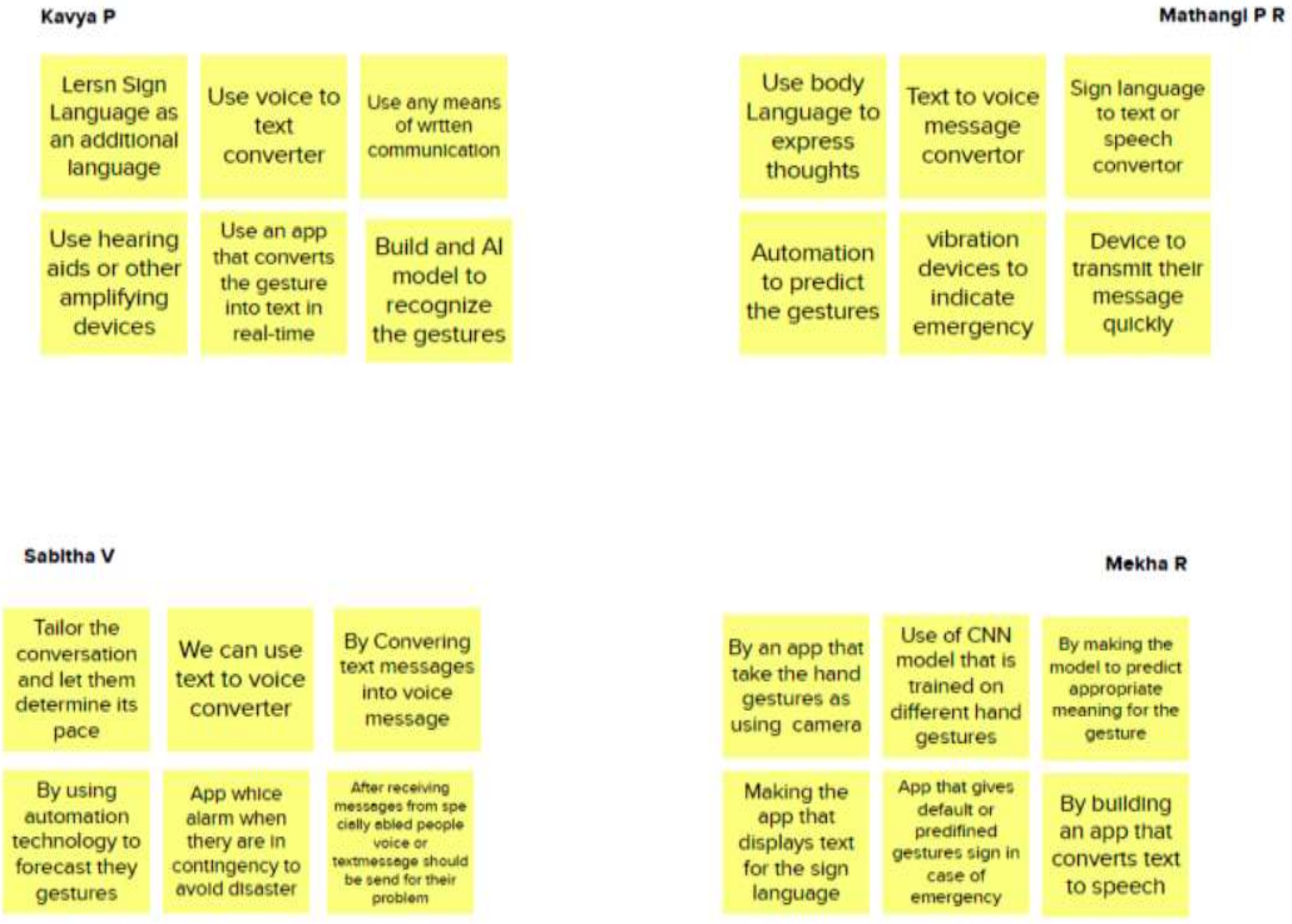
Encourage wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.



Kavya P

Lersn Sign Language as an additional language

Use voice to text converter

Use any means of wrtten communication

Use hearing aids or other amplifying devices

Use an app that converts the gesture into text in real-time

Build and AI model to recognize the gestures

Mathangi P R

Use body Language to express thoughts

Text to voice message convertor

Sign language to text or speech convertor

Automation to predict the gestures

vibration devices to indicate emergency

Device to transmit their message quickly

Sabitha V

Tailor the conversation and let them determine its pace

We can use text to voice converter

By Convering text messages into voice message

By using automation technology to forecast they gestures

App whice alarm when they are in contingency to avoid disaster

After receiving messages from specielly abled people voice or textmessage should be send for their problem

Mekha R

By an app that take the hand gestures as using camera

Use of CNN model that is trained on different hand gestures

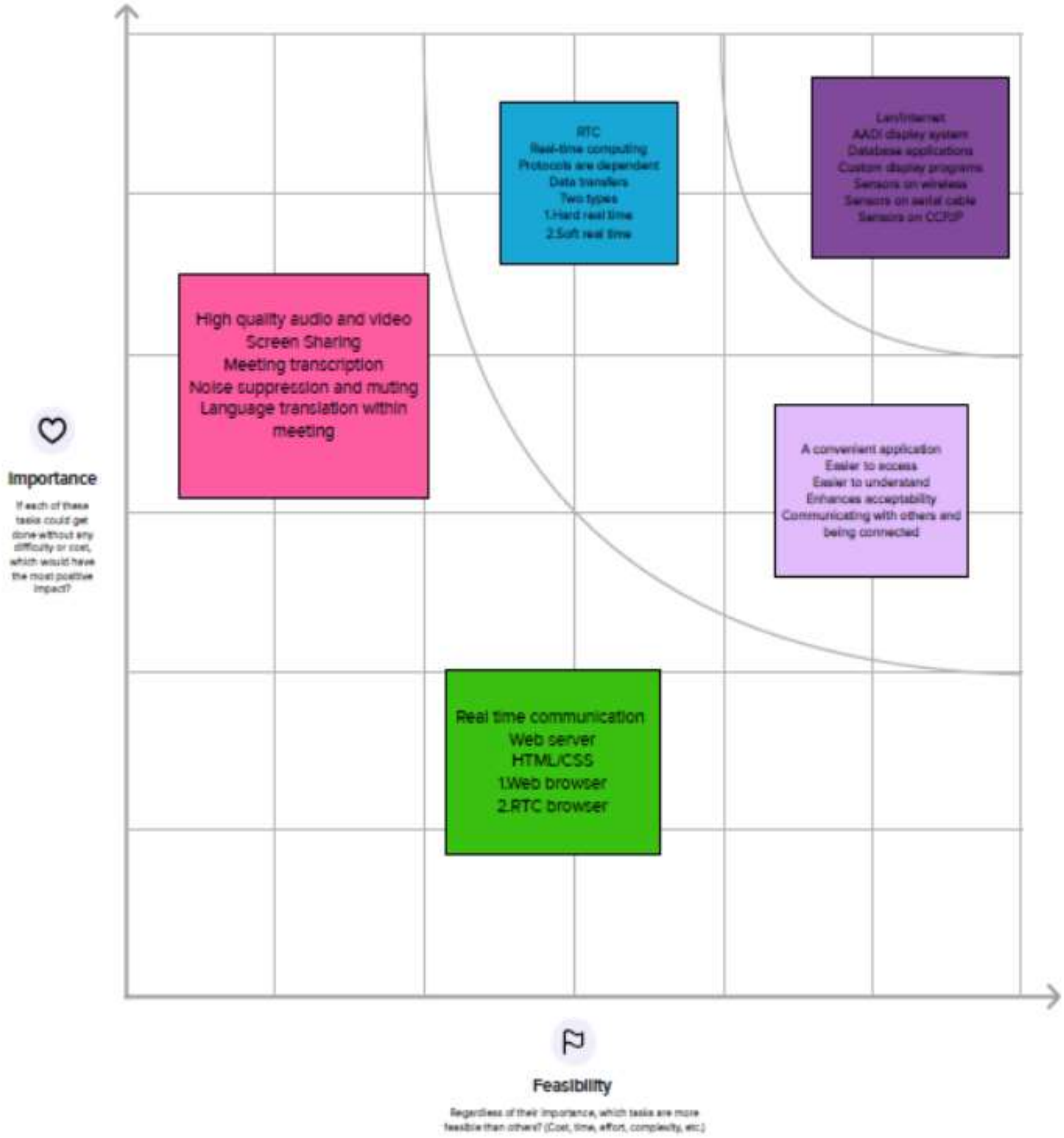
By making the model to predict appropriate meaning for the gesture

Making the app that displays text for the sign language

App that gives default or predifined gestures sign in case of emergency

By building an app that converts text to speech

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.



If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

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

RTC
Real-time computing
Protocols are dependent
Data transfers
Two types
1.Hard real time
2.Soft real time

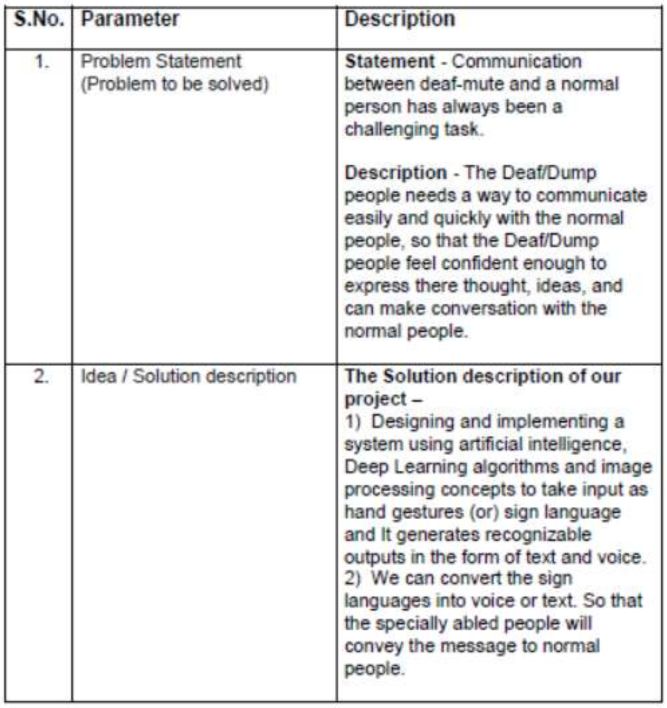
Lan/Internet
AADI display system
Database applications
Custom display programs
Sensors on wireless
Sensors on serial cable
Sensors on CCP/IP

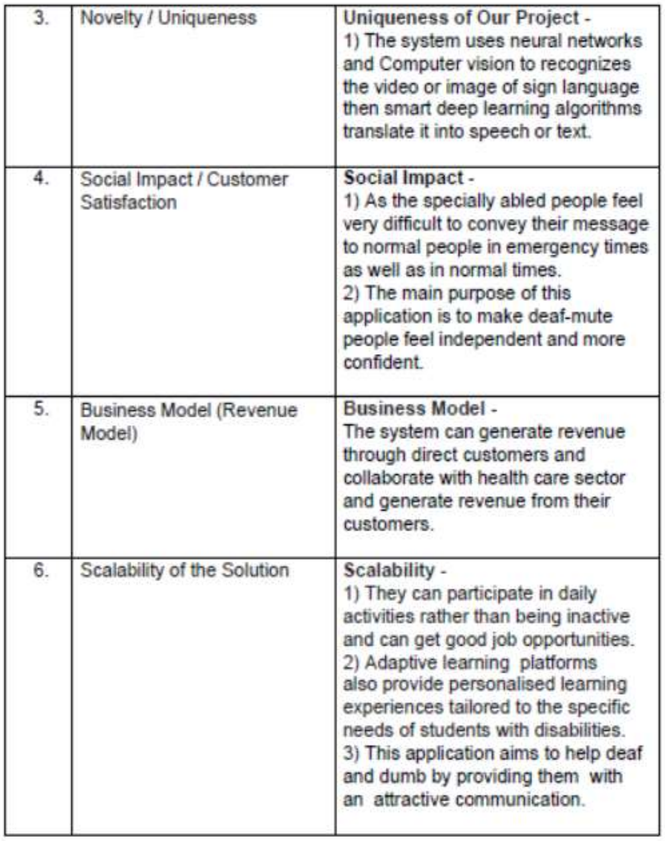
High quality audio and video
Screen Sharing
Meeting transcription
Noise suppression and muting
Language translation within meeting

A convenient application
Easier to access
Easier to understand
Enhances acceptability
Communicating with others and being connected

Real time communication
Web server
HTML/CSS
1.Web browser
2.RTC browser

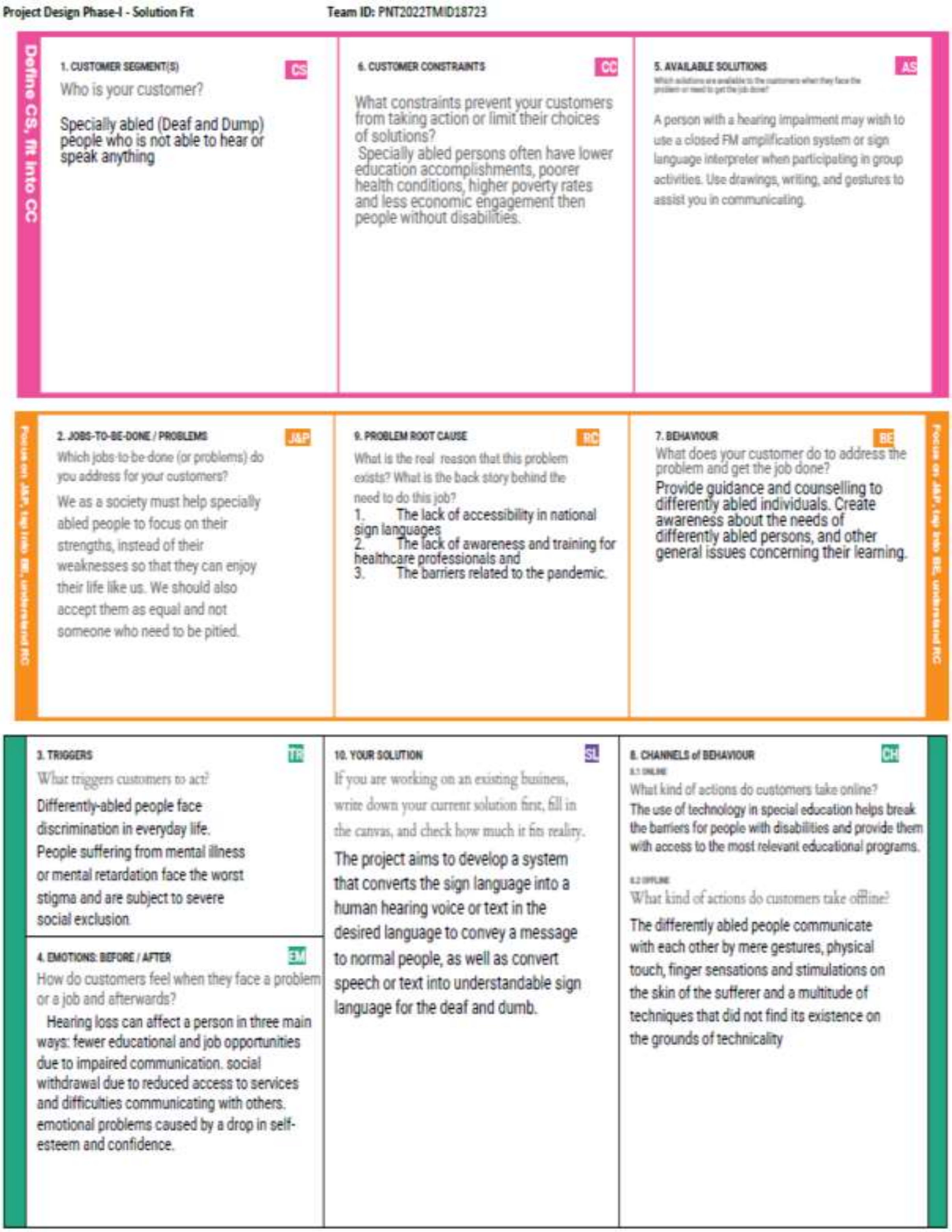
3.3 Proposed solution





S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Statement** - Communication between deaf-mute and a normal person has always been a challenging task. **Description** - The Deaf/Dump people needs a way to communicate easily and quickly with the normal people, so that the Deaf/Dump people feel confident enough to express there thought, ideas, and can make conversation with the normal people.
2.	Idea / Solution description	**The Solution description of our project –** 1) Designing and implementing a system using artificial intelligence, Deep Learning algorithms and image processing concepts to take input as hand gestures (or) sign language and It generates recognizable outputs in the form of text and voice. 2) We can convert the sign languages into voice or text. So that the specially abled people will convey the message to normal people.
3.	Novelty / Uniqueness	**Uniqueness of Our Project** - 1) The system uses neural networks and Computer vision to recognizes the video or image of sign language then smart deep learning algorithms translate it into speech or text.
4.	Social Impact / Customer Satisfaction	**Social Impact** - 1) As the specially abled people feel very difficult to convey their message to normal people in emergency times as well as in normal times. 2) The main purpose of this application is to make deaf-mute people feel independent and more confident.
5.	Business Model (Revenue Model)	**Business Model** - The system can generate revenue through direct customers and collaborate with health care sector and generate revenue from their customers.
6.	Scalability of the Solution	**Scalability** - 1) They can participate in daily activities rather than being inactive and can get good job opportunities. 2) Adaptive learning platforms also provide personalised learning experiences tailored to the specific needs of students with disabilities. 3) This application aims to help deaf and dumb by providing them with an attractive communication.

3.4 Problem Solution Fit



Project Design Phase-I - Solution Fit

Team ID: PNT2022TMID18723

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Specially abled (Deaf and Dump) people who is not able to hear or speak anything

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

Specially abled persons often have lower education accomplishments, poorer health conditions, higher poverty rates and less economic engagement then people without disabilities.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done?

A person with a hearing impairment may wish to use a closed FM amplification system or sign language interpreter when participating in group activities. Use drawings, writing, and gestures to assist you in communicating.

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?

We as a society must help specially abled people to focus on their strengths, instead of their weaknesses so that they can enjoy their life like us. We should also accept them as equal and not someone who need to be pitied.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

1. The lack of accessibility in national sign languages
2. The lack of awareness and training for healthcare professionals and
3. The barriers related to the pandemic.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

Provide guidance and counselling to differently abled individuals. Create awareness about the needs of differently abled persons, and other general issues concerning their learning.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

What triggers customers to act?

Differently-abled people face discrimination in everyday life. People suffering from mental illness or mental retardation face the worst stigma and are subject to severe social exclusion.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

Hearing loss can affect a person in three main ways: fewer educational and job opportunities due to impaired communication. social withdrawal due to reduced access to services and difficulties communicating with others. emotional problems caused by a drop in self-esteem and confidence.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.

The project aims to develop a system that converts the sign language into a human hearing voice or text in the desired language to convey a message to normal people, as well as convert speech or text into understandable sign language for the deaf and dumb.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online?

The use of technology in special education helps break the barriers for people with disabilities and provide them with access to the most relevant educational programs.

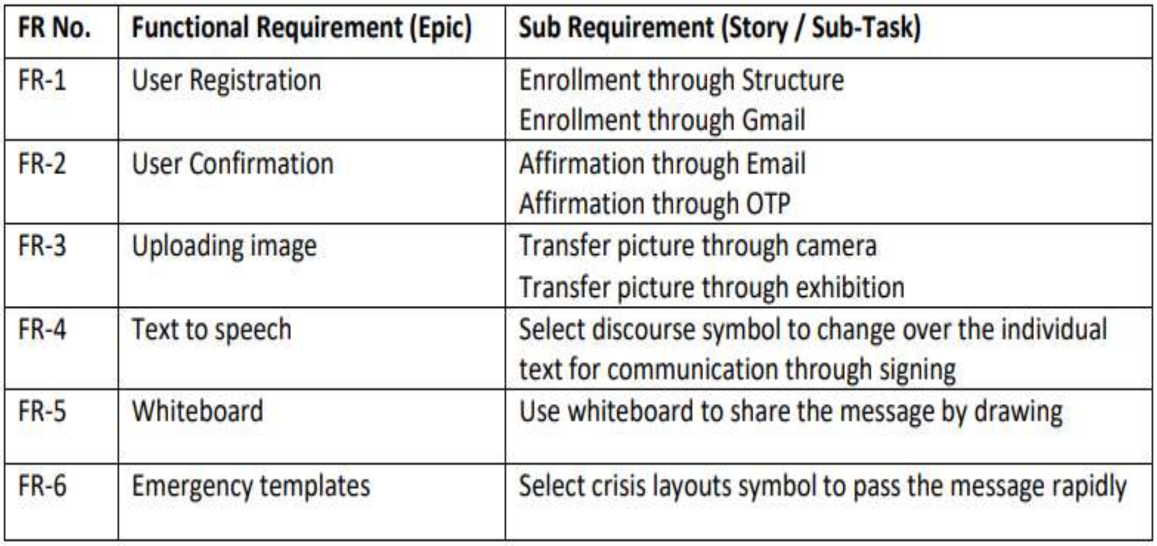
8.2 OFFLINE

What kind of actions do customers take offline?

The differently abled people communicate with each other by mere gestures, physical touch, finger sensations and stimulations on the skin of the sufferer and a multitude of techniques that did not find its existence on the grounds of technicality

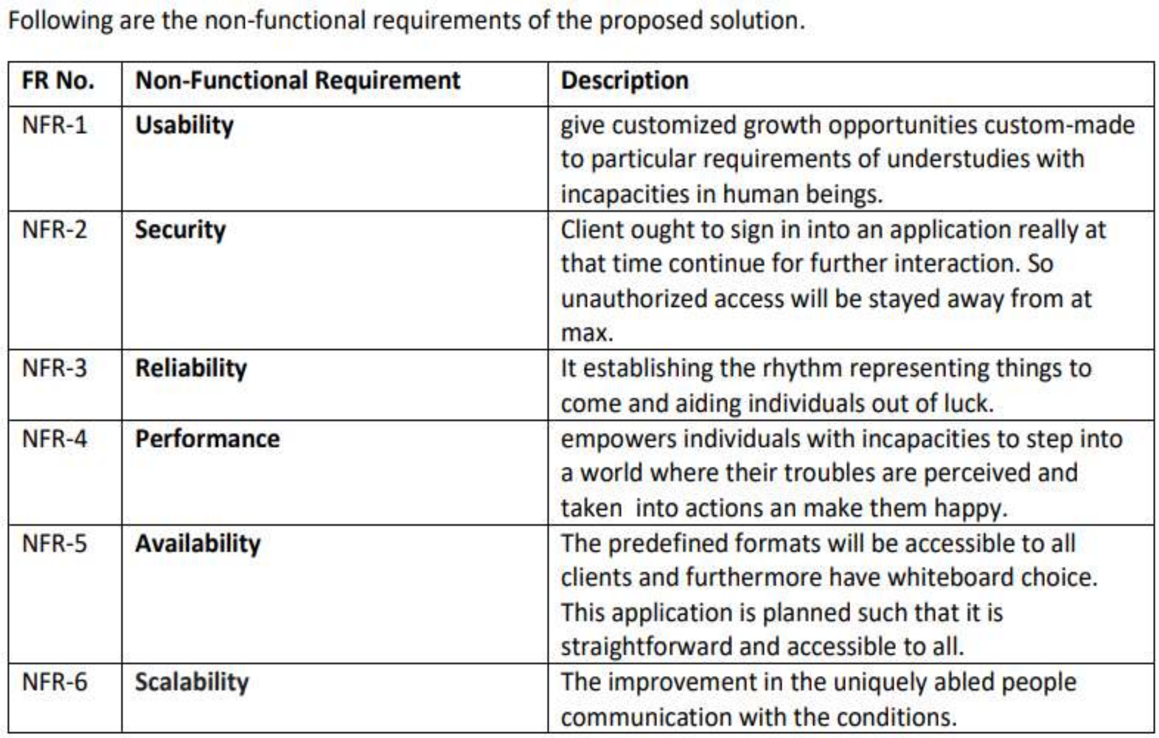
4. REQUIREMENT ANALYSIS

4.1 Functional Requirement



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Enrollment through Structure Enrollment through Gmail
FR-2	User Confirmation	Affirmation through Email Affirmation through OTP
FR-3	Uploading image	Transfer picture through camera Transfer picture through exhibition
FR-4	Text to speech	Select discourse symbol to change over the individual text for communication through signing
FR-5	Whiteboard	Use whiteboard to share the message by drawing
FR-6	Emergency templates	Select crisis layouts symbol to pass the message rapidly

4.2 Non-Functional Requirements

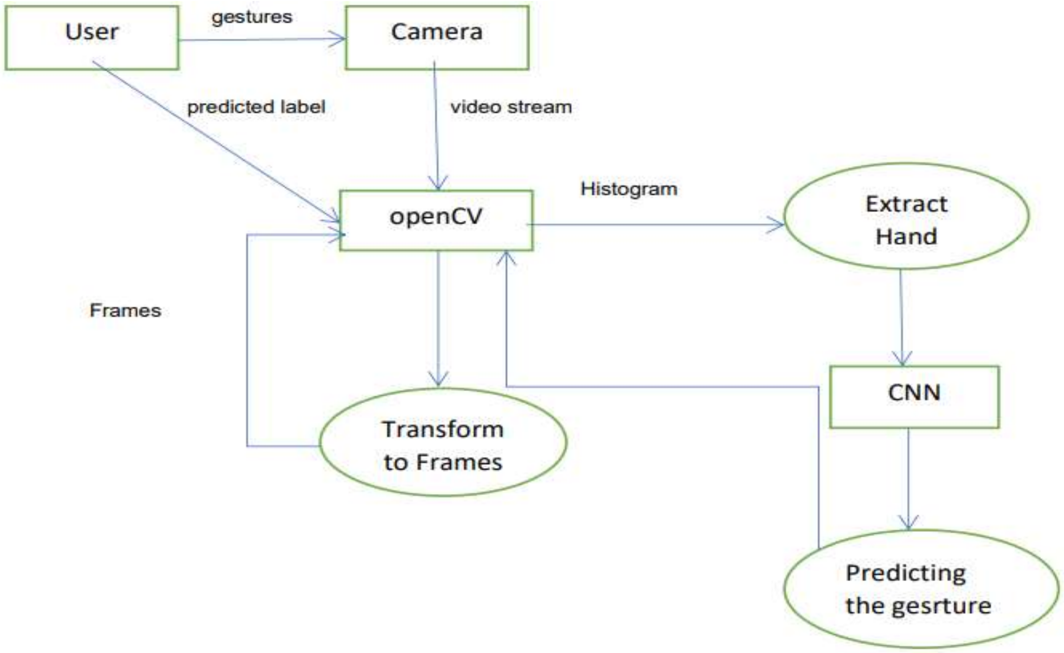


Following are the non-functional requirements of the proposed solution.

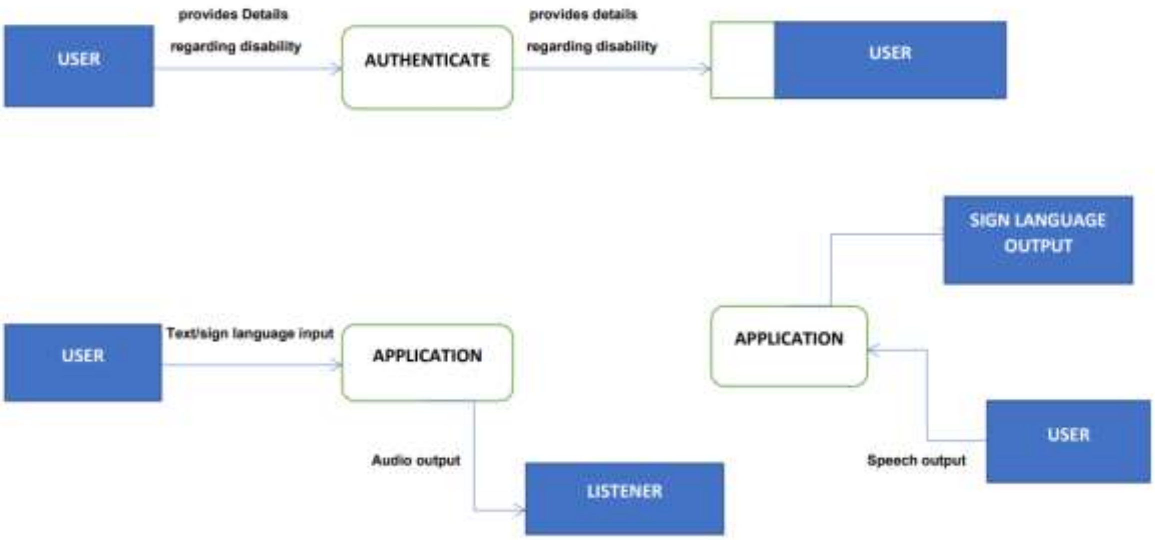
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	give customized growth opportunities custom-made to particular requirements of understudies with incapacities in human beings.
NFR-2	**Security**	Client ought to sign in into an application really at that time continue for further interaction. So unauthorized access will be stayed away from at max.
NFR-3	**Reliability**	It establishing the rhythm representing things to come and aiding individuals out of luck.
NFR-4	**Performance**	empowers individuals with incapacities to step into a world where their troubles are perceived and taken into actions an make them happy.
NFR-5	**Availability**	The predefined formats will be accessible to all clients and furthermore have whiteboard choice. This application is planned such that it is straightforward and accessible to all.
NFR-6	**Scalability**	The improvement in the uniquely abled people communication with the conditions.

5. Project Design

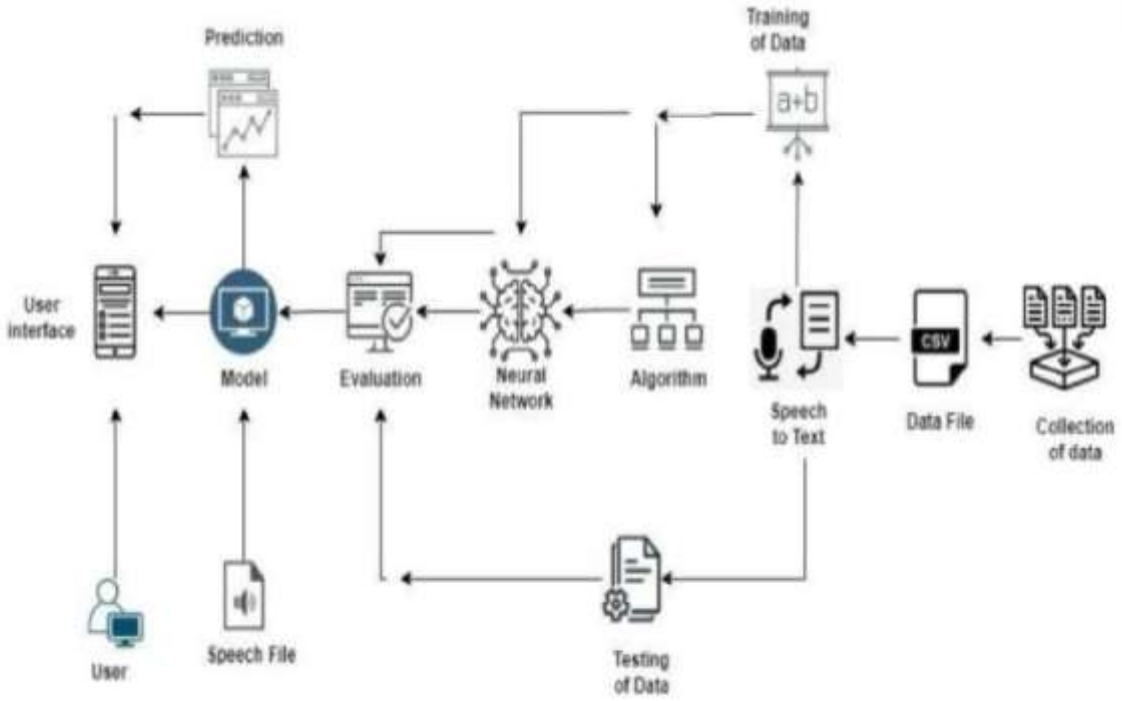
5.1 Data Flow Diagram



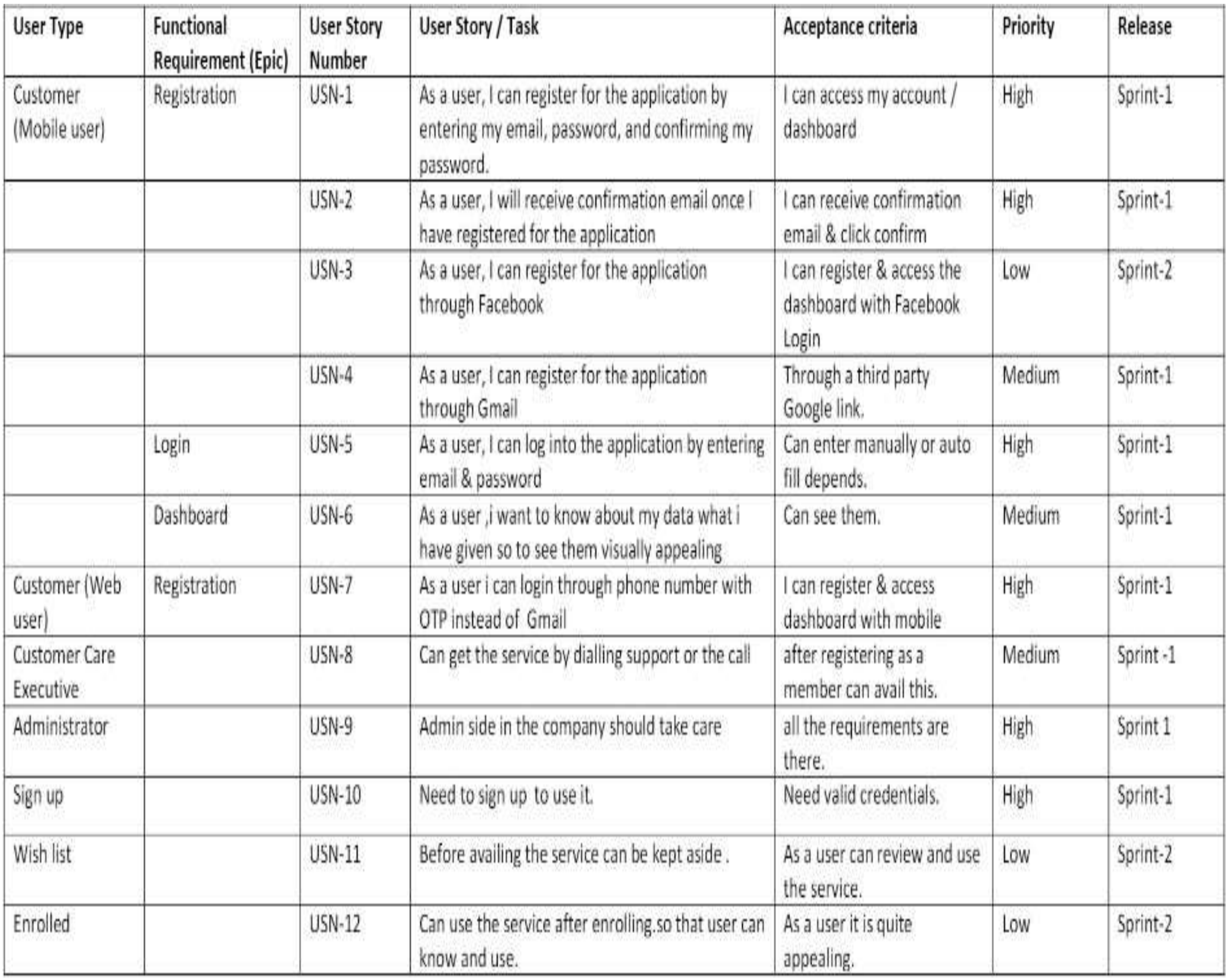
DATA FLOW DIAGRAM



5.2 Solution & Technical Architecture



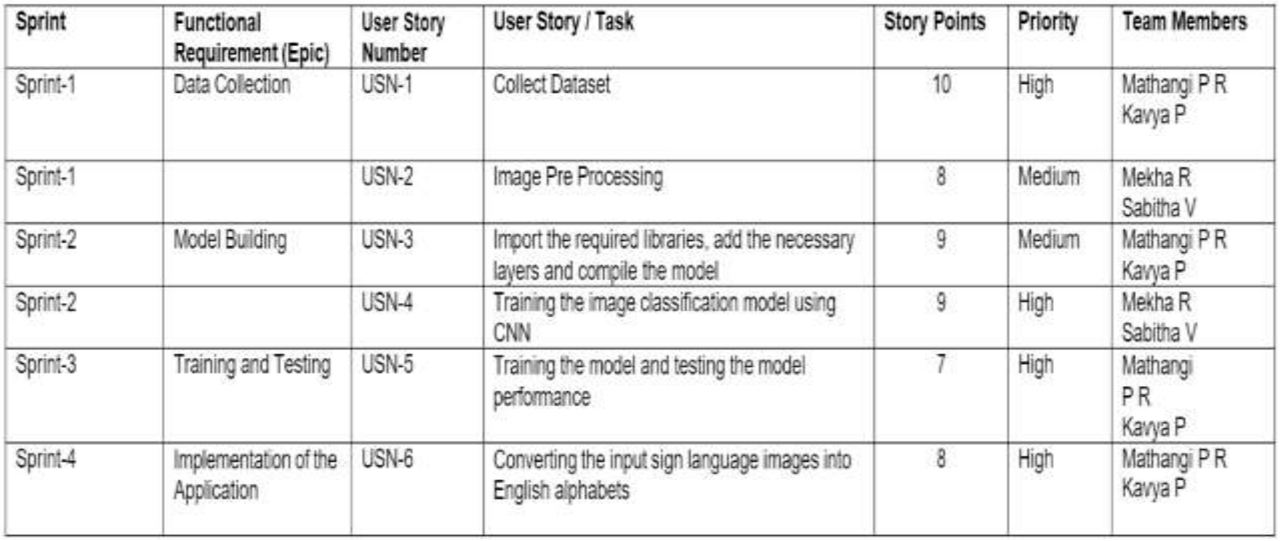
5.3 User stories



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	Sprint-1
		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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	Through a third party Google link.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	Can enter manually or auto fill depends.	High	Sprint-1
	Dashboard	USN-6	As a user ,i want to know about my data what i have given so to see them visually appealing	Can see them.	Medium	Sprint-1
Customer (Web user)	Registration	USN-7	As a user i can login through phone number with OTP instead of Gmail	I can register & access dashboard with mobile	High	Sprint-1
Customer Care Executive		USN-8	Can get the service by dialling support or the call	after registering as a member can avail this.	Medium	Sprint -1
Administrator		USN-9	Admin side in the company should take care	all the requirements are there.	High	Sprint 1
Sign up		USN-10	Need to sign up to use it.	Need valid credentials.	High	Sprint-1
Wish list		USN-11	Before availing the service can be kept aside .	As a user can review and use the service.	Low	Sprint-2
Enrolled		USN-12	Can use the service after enrolling.so that user can know and use.	As a user it is quite appealing.	Low	Sprint-2

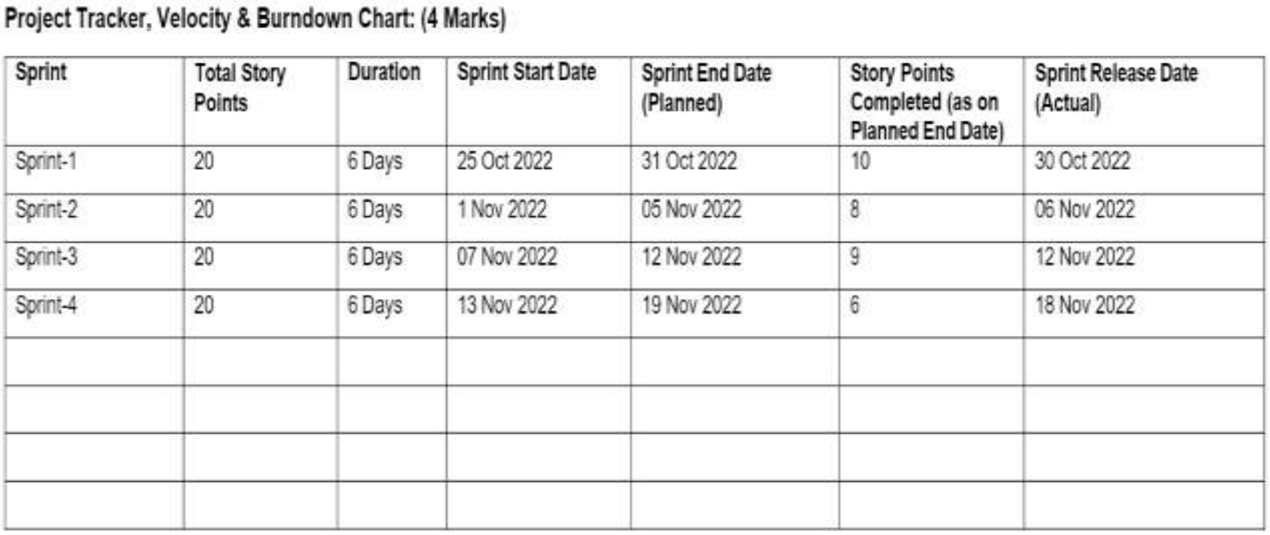
6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint planning and Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect Dataset	10	High	Mathangi P R Kavya P
Sprint-1		USN-2	Image Pre Processing	8	Medium	Mekha R Sabitha V
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	9	Medium	Mathangi P R Kavya P
Sprint-2		USN-4	Training the image classification model using CNN	9	High	Mekha R Sabitha V
Sprint-3	Training and Testing	USN-5	Training the model and testing the model performance	7	High	Mathangi P R Kavya P
Sprint-4	Implementation of the Application	USN-6	Converting the input sign language images into English alphabets	8	High	Mathangi P R Kavya P

6.2 Sprint Delivery Schedule

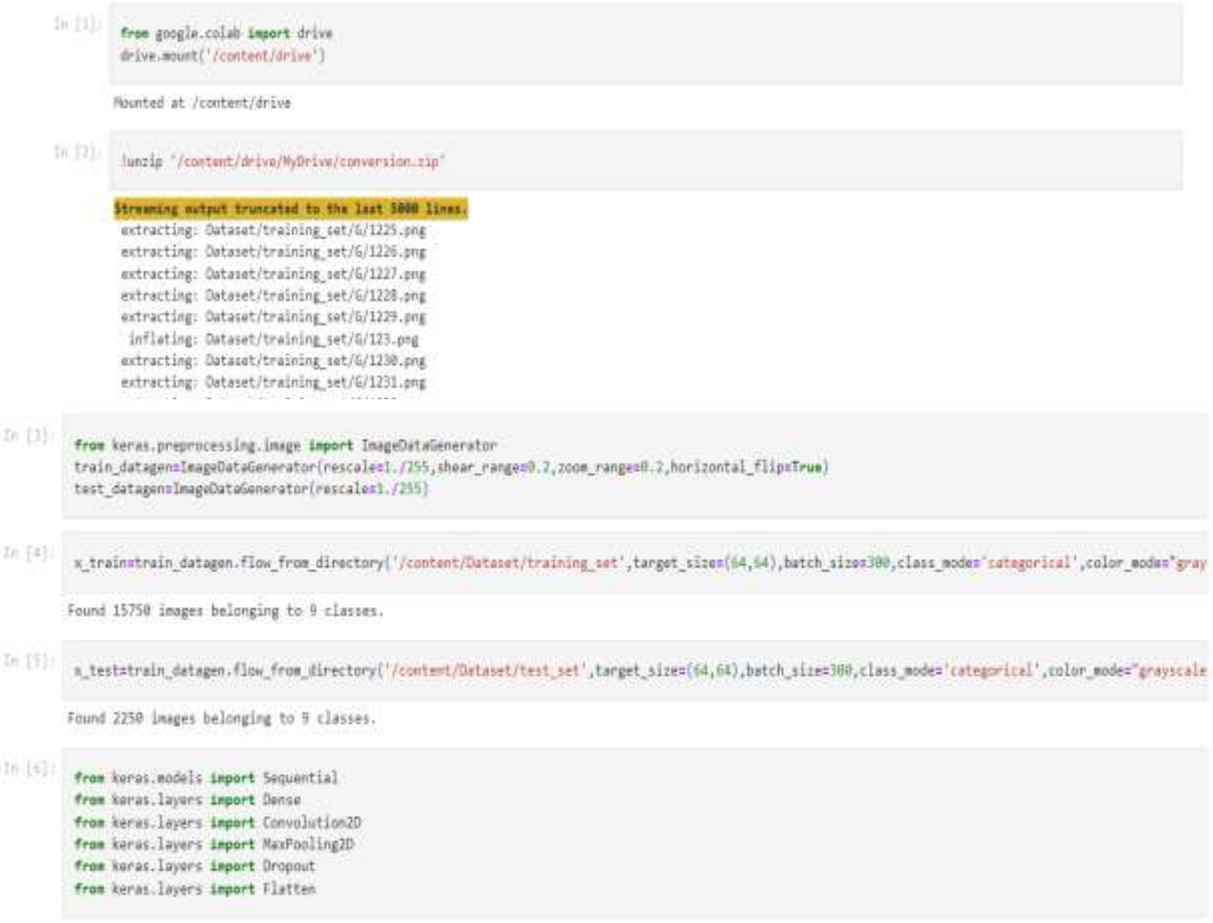


Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	20	6 Days	25 Oct 2022	31 Oct 2022	10	30 Oct 2022
Sprint-2	20	6 Days	1 Nov 2022	05 Nov 2022	8	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	9	12 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	19 Nov 2022	6	18 Nov 2022

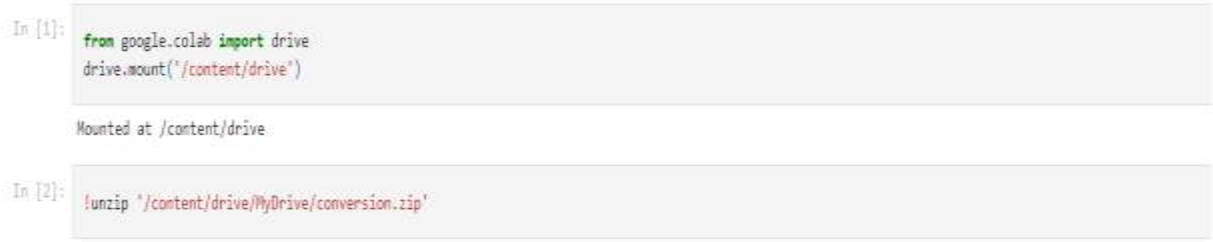
7. CODING & SOLUTIONING

7.1 Model Building

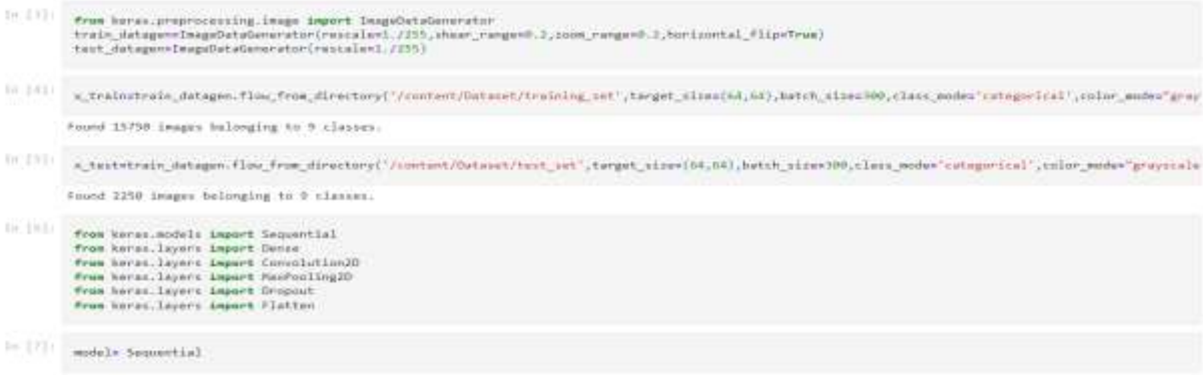


```
In [1]: from google.colab import drive
        drive.mount('/content/drive')

Mounted at /content/drive

In [2]: !unzip '/content/drive/MyDrive/conversion.zip'

Streaming output truncated to the last 5000 lines.
  extracting: Dataset/training_set/G/1225.png
  extracting: Dataset/training_set/G/1226.png
  extracting: Dataset/training_set/G/1227.png
  extracting: Dataset/training_set/G/1228.png
  extracting: Dataset/training_set/G/1229.png
   inflating: Dataset/training_set/G/123.png
  extracting: Dataset/training_set/G/1230.png
  extracting: Dataset/training_set/G/1231.png

In [3]: from keras.preprocessing.image import ImageDataGenerator
        train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
        test_datagen=ImageDataGenerator(rescale=1./255)

In [4]: x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

Found 15750 images belonging to 9 classes.

In [5]: x_test=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscale

Found 2250 images belonging to 9 classes.

In [6]: from keras.models import Sequential
        from keras.layers import Dense
        from keras.layers import Convolution2D
        from keras.layers import MaxPooling2D
        from keras.layers import Dropout
        from keras.layers import Flatten
```

INITIALIZING THE MODEL

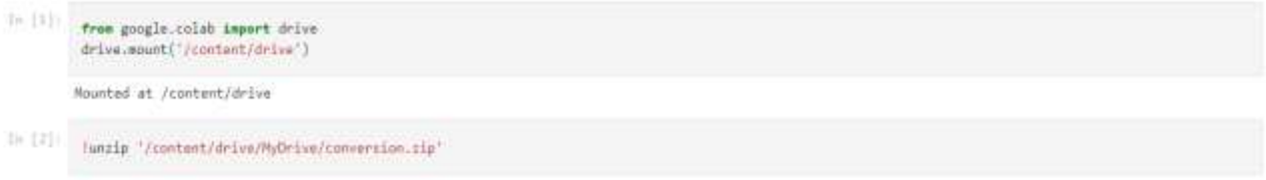


```
In [1]:
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

In [2]:
!unzip '/content/drive/MyDrive/conversion.zip'
```

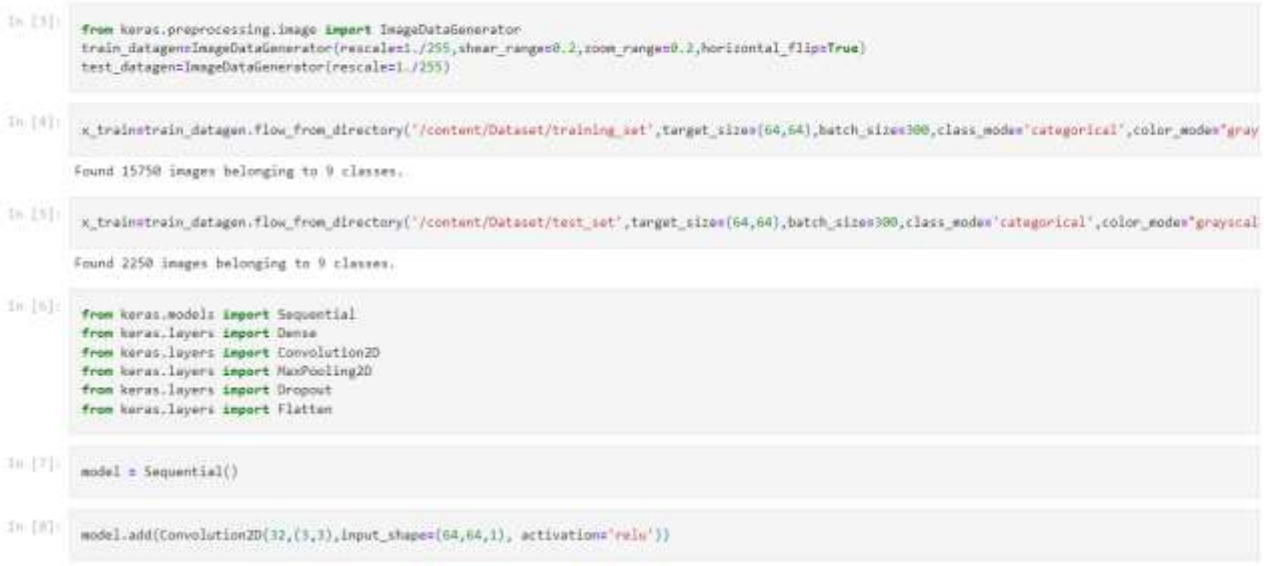


```
In [3]:
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

In [4]:
x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

Found 15750 images belonging to 9 classes.

In [5]:
x_test=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscale

Found 2250 images belonging to 9 classes.

In [6]:
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten

In [7]:
model= Sequential
```

ADDING THE CONVOLUTION LAYER

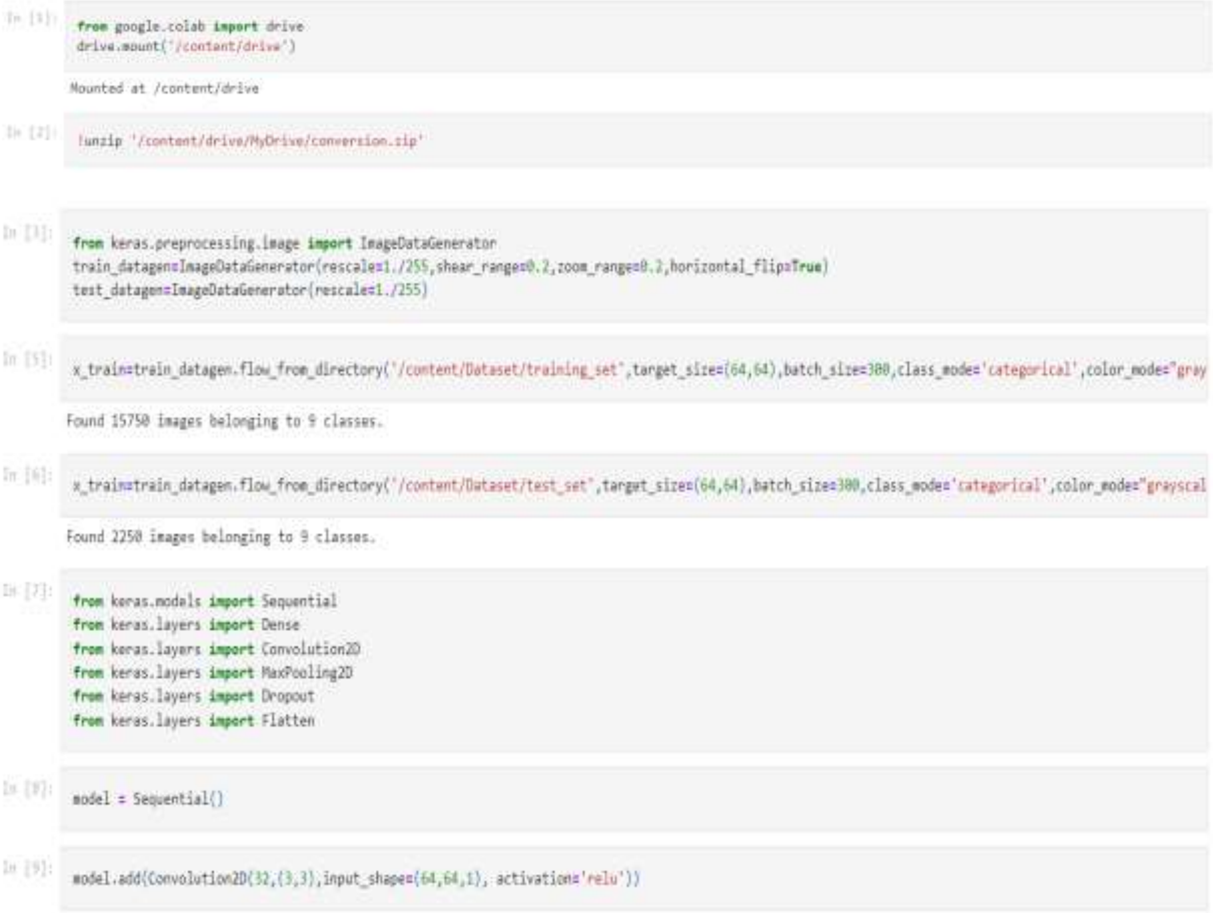


```
In [1]:
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

In [2]:
!unzip '/content/drive/MyDrive/conversion.zip'
```



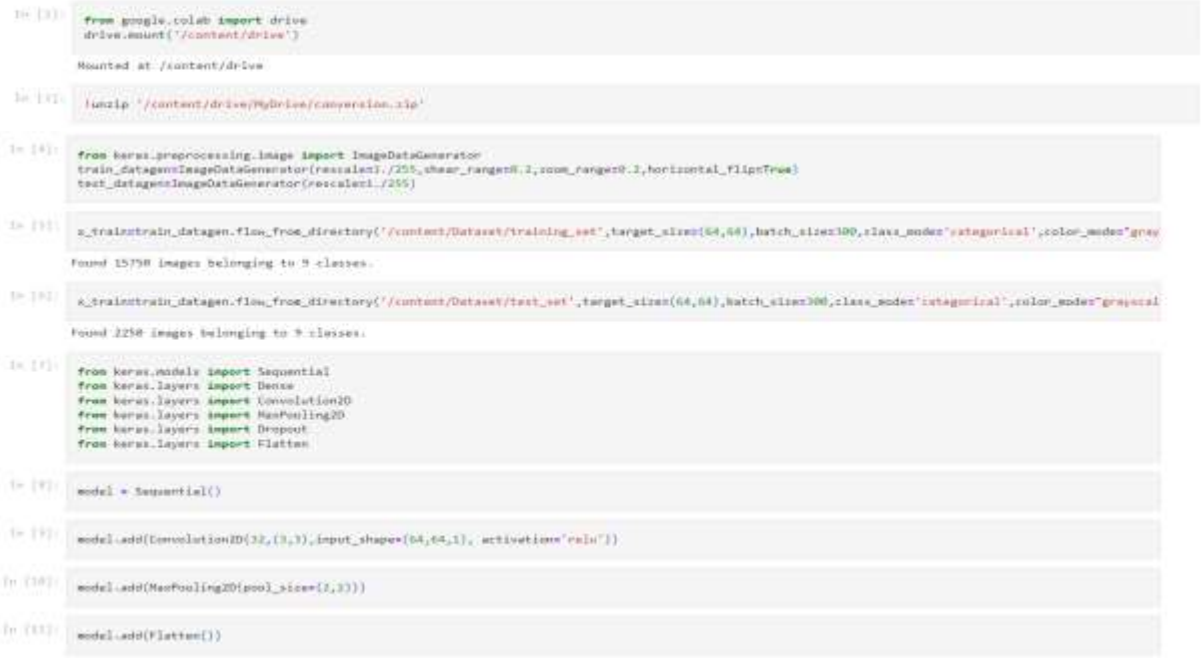
```
In [3]:
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

In [4]:
x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

Found 15750 images belonging to 9 classes.

In [5]:
x_train=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscal

Found 2250 images belonging to 9 classes.

In [6]:
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten

In [7]:
model = Sequential()

In [8]:
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1), activation='relu'))
```

ADDING THE POOLING LAYER



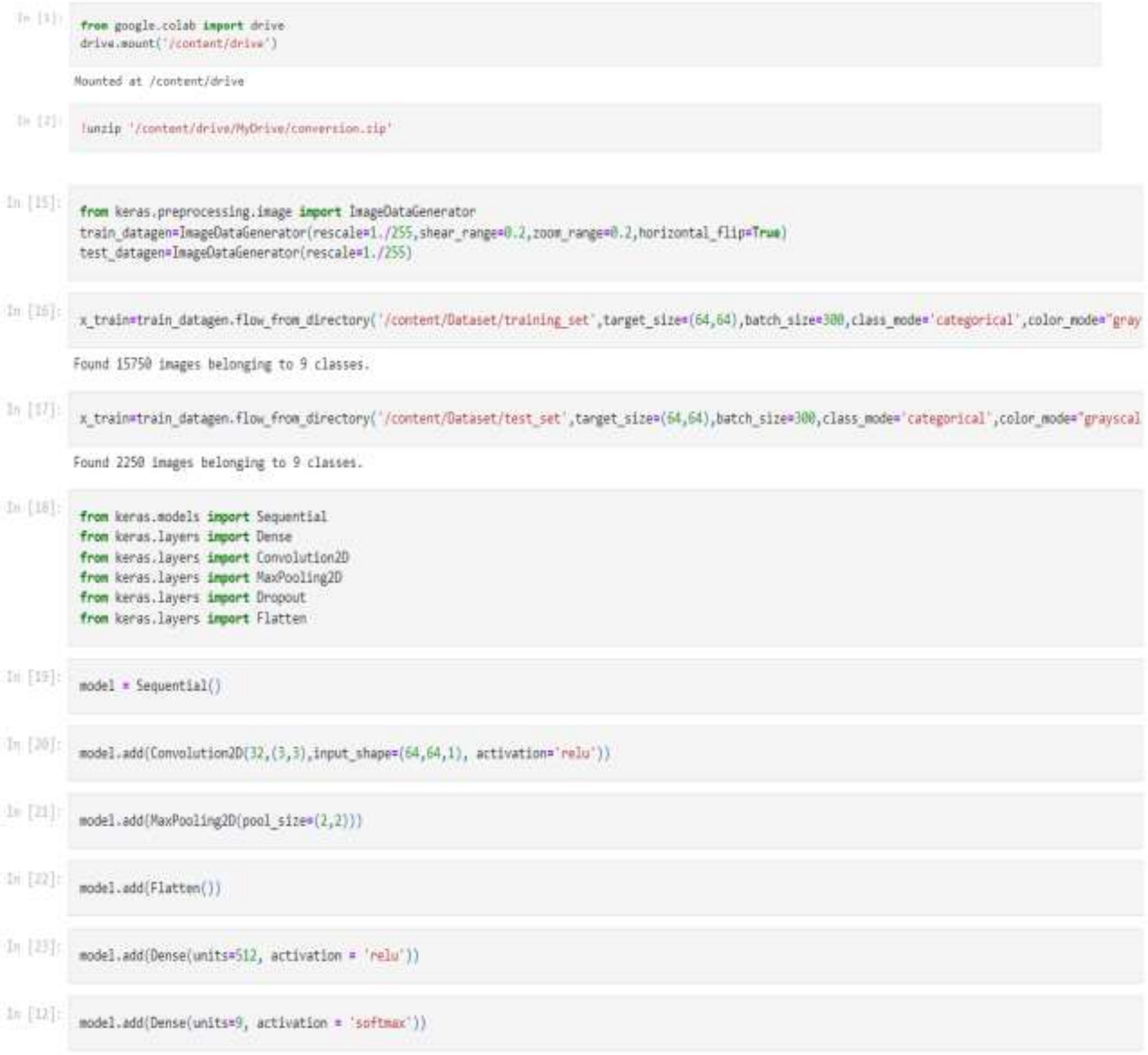
```
In [1]: from google.colab import drive
        drive.mount('/content/drive')

        Mounted at /content/drive

In [2]: !unzip '/content/drive/MyDrive/conversion.zip'

In [3]: from keras.preprocessing.image import ImageDataGenerator
        train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
        test_datagen=ImageDataGenerator(rescale=1./255)

In [5]: x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

        Found 15750 images belonging to 9 classes.

In [6]: x_train=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscal

        Found 2250 images belonging to 9 classes.

In [7]: from keras.models import Sequential
        from keras.layers import Dense
        from keras.layers import Convolution2D
        from keras.layers import MaxPooling2D
        from keras.layers import Dropout
        from keras.layers import Flatten

In [8]: model = Sequential()

In [9]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,1), activation='relu'))
```

ADDING THE FLATTEN LAYER



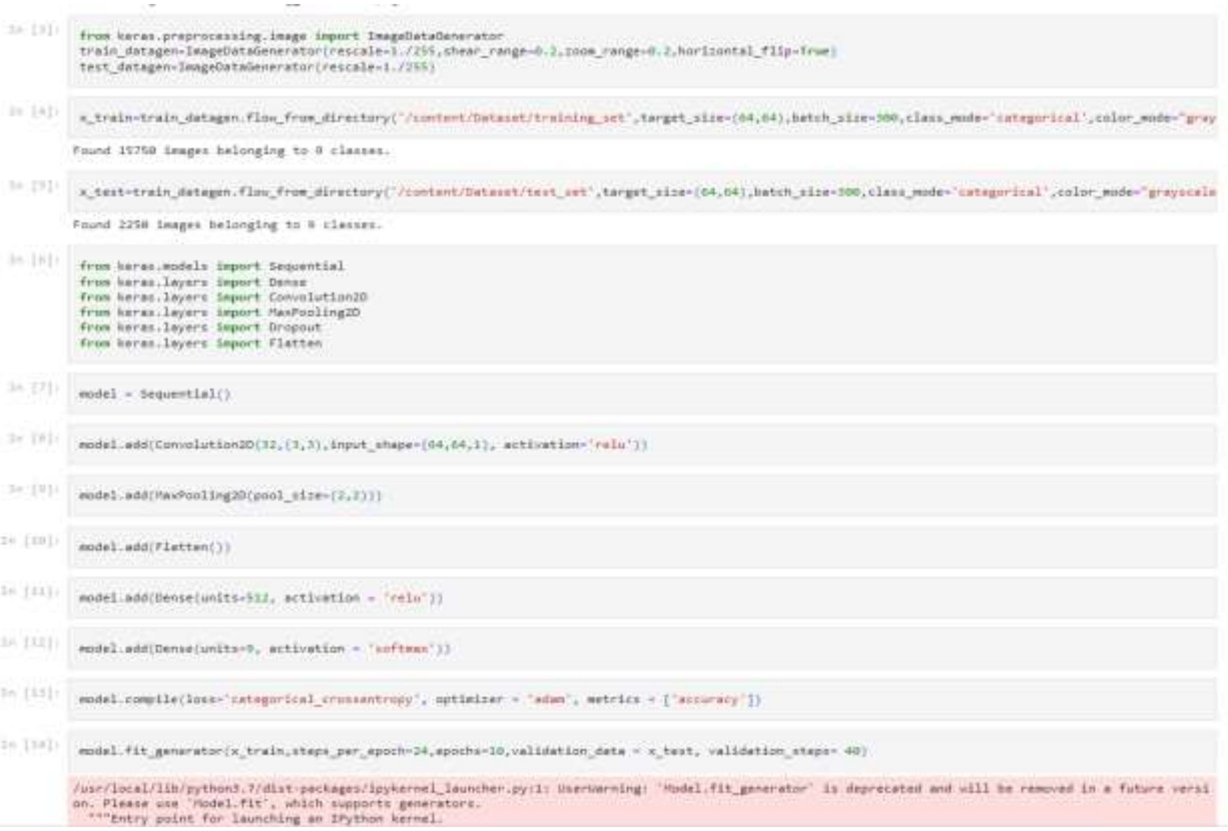
```
In [1]: from google.colab import drive
        drive.mount('/content/drive')

        Mounted at /content/drive

In [2]: !unzip '/content/drive/MyDrive/conversion.zip'

In [4]: from keras.preprocessing.image import ImageDataGenerator
        train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
        test_datagen=ImageDataGenerator(rescale=1./255)

In [5]: x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

        Found 15750 images belonging to 9 classes.

In [6]: x_train=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscal

        Found 2250 images belonging to 9 classes.

In [7]: from keras.models import Sequential
        from keras.layers import Dense
        from keras.layers import Convolution2D
        from keras.layers import MaxPooling2D
        from keras.layers import Dropout
        from keras.layers import Flatten

In [8]: model = Sequential()

In [9]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,1), activation='relu'))

In [10]: model.add(MaxPooling2D(pool_size=(2,2)))

In [11]: model.add(Flatten())
```

ADDING DENSE LAYER

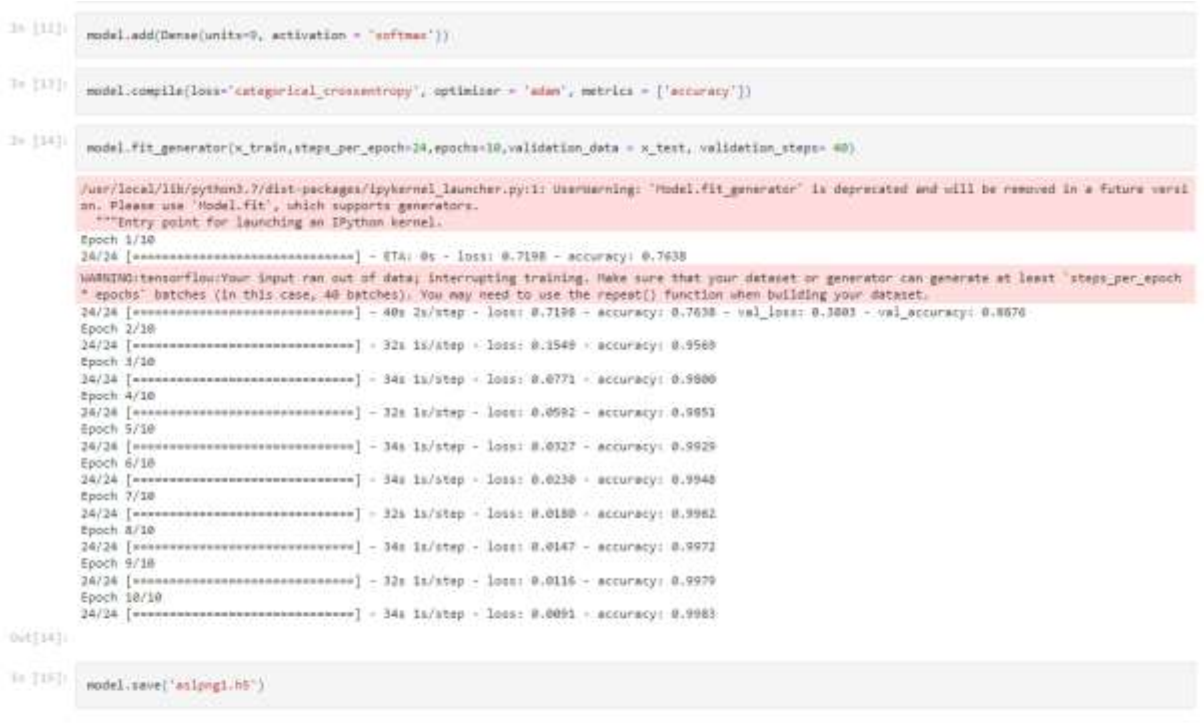


```
In [1]:  from google.colab import drive
         drive.mount('/content/drive')

         Mounted at /content/drive

In [2]:  !unzip '/content/drive/MyDrive/conversion.zip'

In [15]: from keras.preprocessing.image import ImageDataGenerator
         train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
         test_datagen=ImageDataGenerator(rescale=1./255)

In [16]: x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

         Found 15750 images belonging to 9 classes.

In [17]: x_train=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscal

         Found 2250 images belonging to 9 classes.

In [18]: from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import Convolution2D
         from keras.layers import MaxPooling2D
         from keras.layers import Dropout
         from keras.layers import Flatten

In [19]: model = Sequential()

In [20]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,1), activation='relu'))

In [21]: model.add(MaxPooling2D(pool_size=(2,2)))

In [22]: model.add(Flatten())

In [23]: model.add(Dense(units=512, activation = 'relu'))

In [12]: model.add(Dense(units=9, activation = 'softmax'))
```

FIT AND SAVING THE MODEL

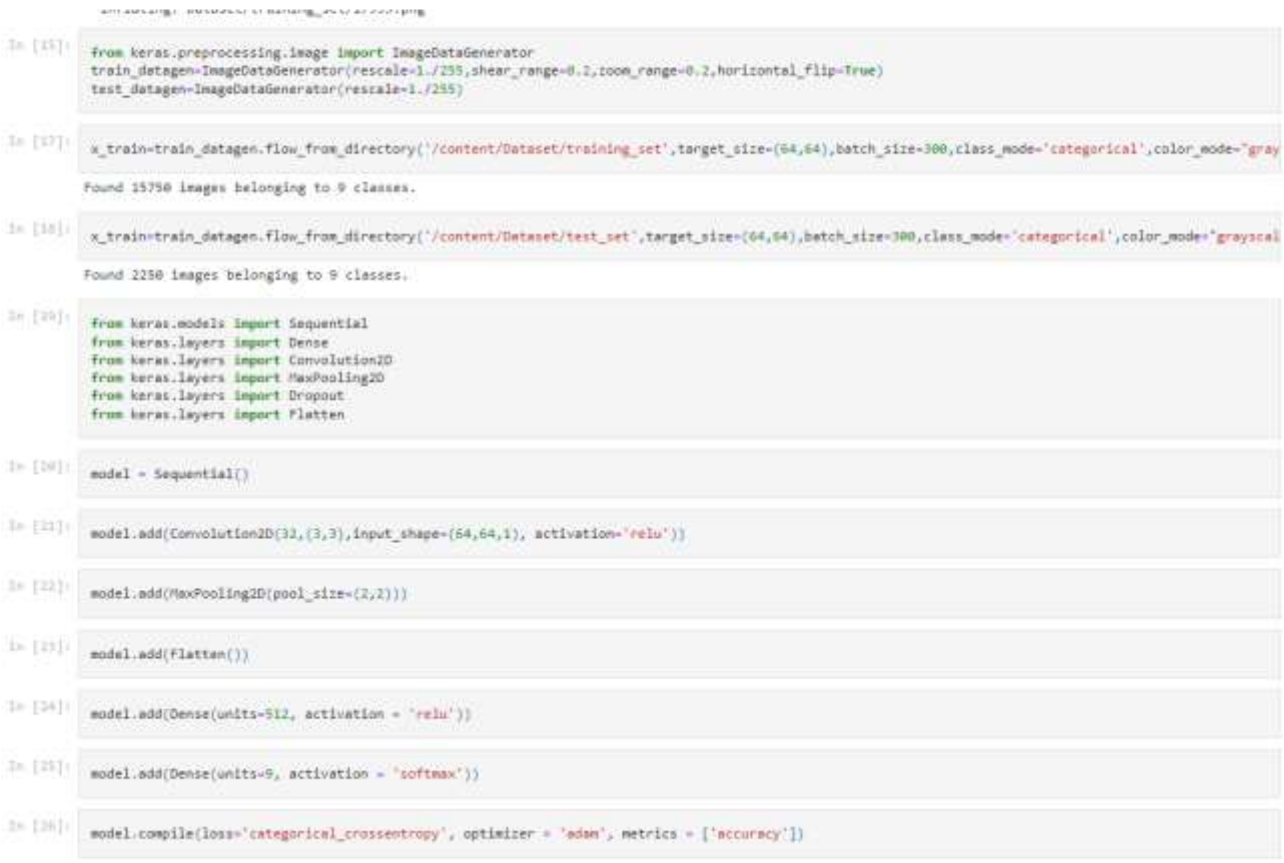


```
In [3]: from keras.preprocessing.image import ImageDataGenerator
        train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
        test_datagen=ImageDataGenerator(rescale=1./255)

In [4]: x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

        Found 15750 images belonging to 9 classes.

In [5]: x_test=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscale

        Found 2250 images belonging to 9 classes.

In [6]: from keras.models import Sequential
        from keras.layers import Dense
        from keras.layers import Convolution2D
        from keras.layers import MaxPooling2D
        from keras.layers import Dropout
        from keras.layers import Flatten

In [7]: model = Sequential()

In [8]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,1), activation='relu'))

In [9]: model.add(MaxPooling2D(pool_size=(2,2)))

In [10]: model.add(Flatten())

In [11]: model.add(Dense(units=512, activation = 'relu'))

In [12]: model.add(Dense(units=9, activation = 'softmax'))

In [13]: model.compile(loss='categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

In [14]: model.fit_generator(x_train,steps_per_epoch=24,epochs=10,validation_data = x_test, validation_steps= 40)

        /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versi
        on. Please use `Model.fit`, which supports generators.
          """Entry point for launching an IPython kernel.
```



```
In [12]: model.add(Dense(units=9, activation = 'softmax'))

In [13]: model.compile(loss='categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

In [14]: model.fit_generator(x_train,steps_per_epoch=24,epochs=10,validation_data = x_test, validation_steps= 40)

         /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versi
         on. Please use `Model.fit`, which supports generators.
           """Entry point for launching an IPython kernel.
         Epoch 1/10
         24/24 [==============================] - ETA: 0s - loss: 0.7198 - accuracy: 0.7638
         WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch
         * epochs` batches (in this case, 40 batches). You may need to use the repeat() function when building your dataset.
         24/24 [==============================] - 40s 2s/step - loss: 0.7198 - accuracy: 0.7638 - val_loss: 0.3803 - val_accuracy: 0.8676
         Epoch 2/10
         24/24 [==============================] - 32s 1s/step - loss: 0.1540 - accuracy: 0.9560
         Epoch 3/10
         24/24 [==============================] - 34s 1s/step - loss: 0.0771 - accuracy: 0.9800
         Epoch 4/10
         24/24 [==============================] - 32s 1s/step - loss: 0.0592 - accuracy: 0.9851
         Epoch 5/10
         24/24 [==============================] - 34s 1s/step - loss: 0.0327 - accuracy: 0.9929
         Epoch 6/10
         24/24 [==============================] - 34s 1s/step - loss: 0.0230 - accuracy: 0.9948
         Epoch 7/10
         24/24 [==============================] - 32s 1s/step - loss: 0.0180 - accuracy: 0.9962
         Epoch 8/10
         24/24 [==============================] - 34s 1s/step - loss: 0.0147 - accuracy: 0.9972
         Epoch 9/10
         24/24 [==============================] - 32s 1s/step - loss: 0.0116 - accuracy: 0.9979
         Epoch 10/10
         24/24 [==============================] - 34s 1s/step - loss: 0.0091 - accuracy: 0.9983
Out[14]:

In [15]: model.save('aslpng1.h5')
```

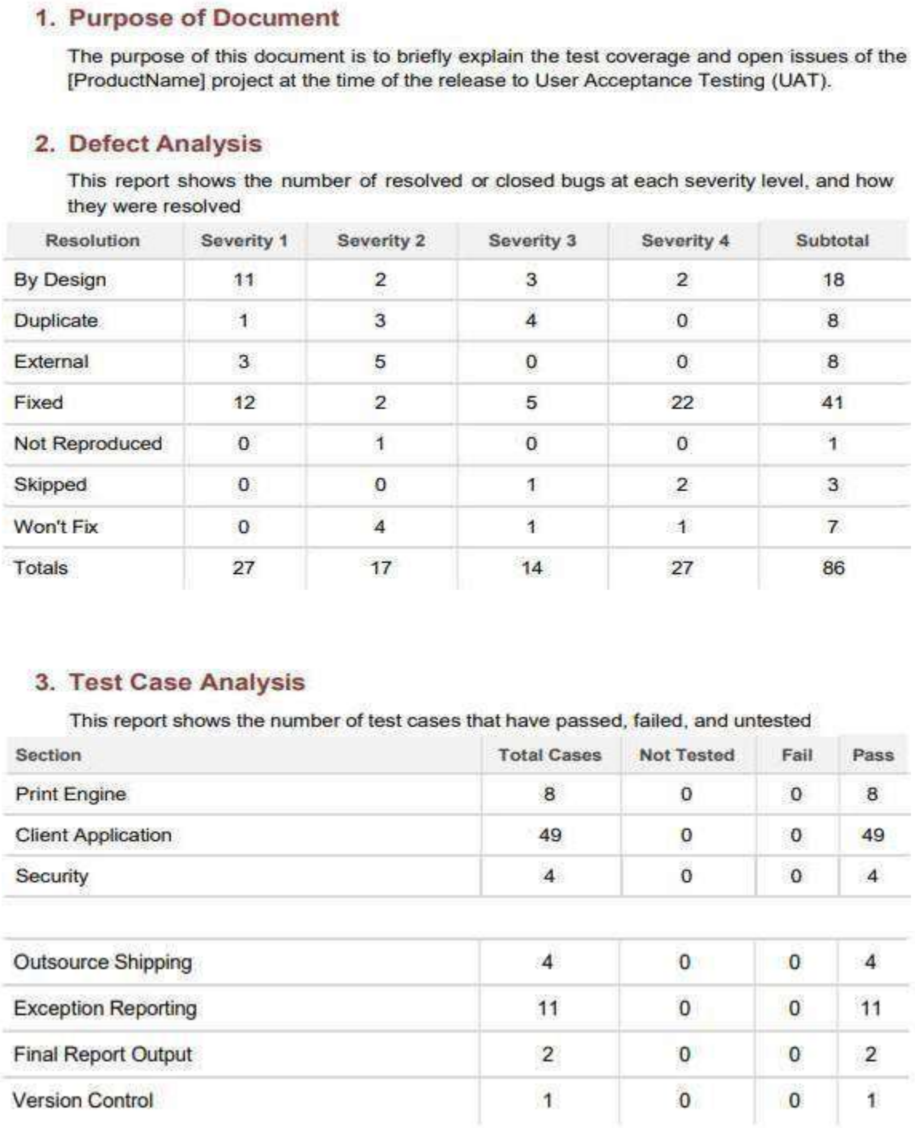
COMPILE THE MODEL



```
  inflating: Dataset/training_set/[illegible].png

In [15]:  from keras.preprocessing.image import ImageDataGenerator
          train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
          test_datagen=ImageDataGenerator(rescale=1./255)

In [17]:  x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="gray

          Found 15750 images belonging to 9 classes.

In [18]:  x_train=train_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscal

          Found 2250 images belonging to 9 classes.

In [19]:  from keras.models import Sequential
          from keras.layers import Dense
          from keras.layers import Convolution2D
          from keras.layers import MaxPooling2D
          from keras.layers import Dropout
          from keras.layers import Flatten

In [20]:  model = Sequential()

In [21]:  model.add(Convolution2D(32,(3,3),input_shape=(64,64,1), activation='relu'))

In [22]:  model.add(MaxPooling2D(pool_size=(2,2)))

In [23]:  model.add(Flatten())

In [24]:  model.add(Dense(units=512, activation = 'relu'))

In [25]:  model.add(Dense(units=9, activation = 'softmax'))

In [26]:  model.compile(loss='categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
```

8. TESTING

8.1 User Acceptance Testing



1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	2	3	2	18
Duplicate	1	3	4	0	8
External	3	5	0	0	8
Fixed	12	2	5	22	41
Not Reproduced	0	1	0	0	1
Skipped	0	0	1	2	3
Won't Fix	0	4	1	1	7
Totals	27	17	14	27	86

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

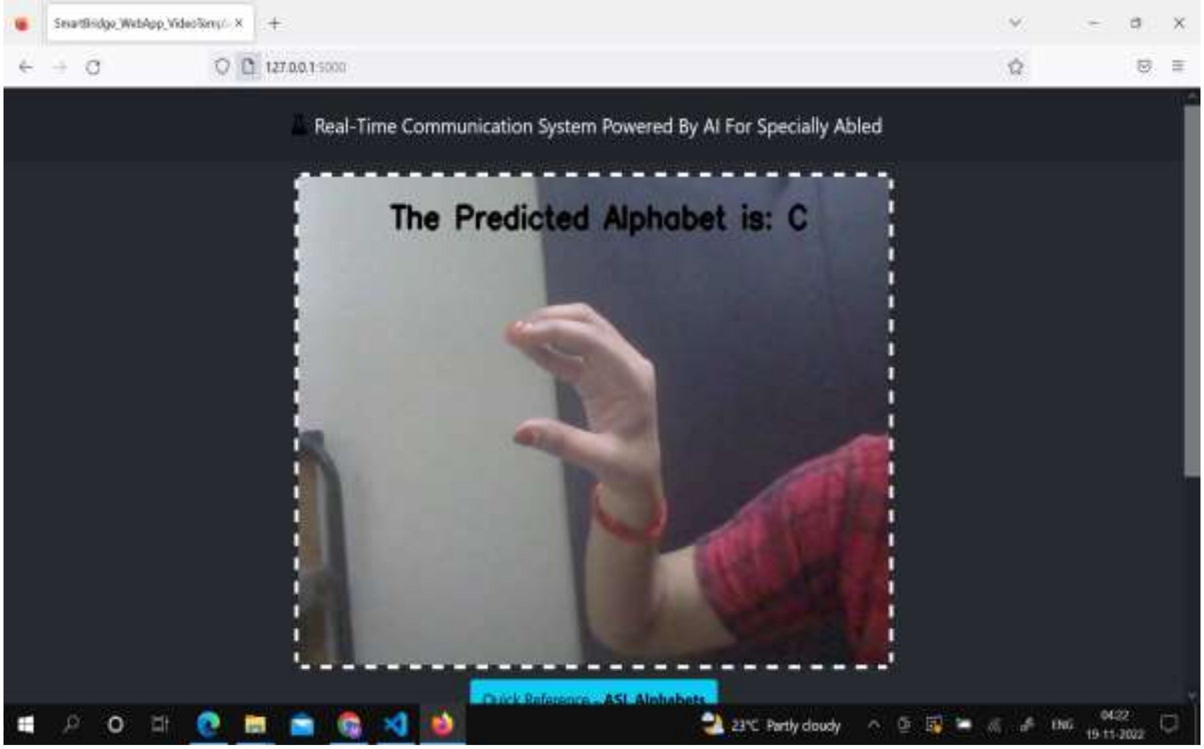
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	49	0	0	49
Security	4	0	0	4
Outsource Shipping	4	0	0	4
Exception Reporting	11	0	0	11
Final Report Output	2	0	0	2
Version Control	1	0	0	1

9. RESULTS

9.1 Performance Metrics

The proposed procedure was implemented and tested with set of images.The set of 15750 images of Alphabets from “A” to “I” are used for training database and a set of 2250 images of Alphabets from “A” to “I”are used for testing database. Once the gesture is recognise the equivalent Alphabet is shown on the screen.

Some sample images of the output are provided below:



Real-Time Communication System Powered By AI For Specially Abled

The Predicted Alphabet is: C

SmartBridge_WebApp_VideoTemp...
127.0.0.1:5000
Quick Reference - ASL Alphabets
23°C Partly cloudy
ENG
04:22
19-11-2022

10.ADVANTAGES AND DISADVANTAGES

- It is possible to create a Android application to bridge the conversation hole between deaf and dumb persons and most of the people in surroundings.
- As one of a kind signal language requirements exist, their dataset can be added, and the user can pick out which sign language to read.
- As the amount/exceptional of photos within the dataset is low, the accuracy is not incredible, however that could without difficulty be progressed by means of improved in dataset.
- Also accuracy depends upon distance between camera and object.
- It takes a some bit lot of time to listen, talk, examine, or write to a person.

11. CONCLUSION

Sign language is a useful device for facilitating conversation among deaf and listening to people. As it permits for two-way conversation, the device goals to bridge the communication gap between deaf humans and the relaxation of society. The proposed method interprets language which are understandable to humans. The system sends hand gestures to the version who recognises them and presentations the equivalent output on the display. Deaf-mute humans can use their hands to carry out signal languages that allows you to then be converted and corresponding end result would be produced.

12 . FUTURE SCOPE

Having the generation that may translate hand sign languages correspondingly is a recreation changer within the subject of verbal exchange and AI for mainly specially abled humans including deaf and dumb. With advance of gesture recognition, the web app can without difficulty be explained to apprehend all the activities that the gesture recognition can also permit controlling of software/hardware interfaces among all circumferences.