REAL - TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

SUBMITTED BY

J SAFRIN NIHAR - TEAM LEAD

K KARPAGAVALLI - TEAM MEMBER 1

L BHUVANESWAR - TEAM MEMBER 2

P PODHUMANI - TEAM MEMBER 3

TEAMID: PNT2022TMID48097

SHANMUGANATHAN ENGNEERING COLLEGE

INDEX

CHAPTER 1

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Real-time communications (RTC) is any mode of telecommunications in which all users can exchange information instantly or with negligible latency or transmission delays. IN RTC, there is always a direct path between the source and the destination. Although the link might contain several intermediate nodes, the data goes from source to destination without being stored in between them. In contrast, asynchronous or time shifting communications, such as email and voicemail, always involve some form of data storage between the source and the destination. In these cases, there is an anticipated delay between the transmission and receipt of the information.

1.2 PURPOSE

Real-time communication (RTC) refers to any communication that happens between two (or more) Individuals in real-time – with minimal latency and without transmission delays. Some examples of Real-time communication include landline phones, mobile calls, instant messaging, VoIP, and video Conferencing. This enables deaf and dumb people to convey their information using sign language and text is given as output.

The main purpose of sign language recognition is to provide an efficient and accurate sign language into text or voice has aids for the hearing impaired for example or enabling very young children to interact with computers.

CHAPTER 2

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

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. Communication 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.

2.2 REFERENCES

1. Koufos, K., EL Haloui, K., Dianati, M., Higgins, M., Elmirghani, J., Imran, M. A., &Tafazolli, R.(2021). Trends in Intelligent Communication Systems: Review of Standards, Major Research Projects, and Identification of Research Gaps. Journal of Sensor and Actuator Networks, 10(4), 60.

2. Panda, G., Upadhyay, A. K., & Khandelwal, K. (2019). Artificial intelligence: A strategic disruption in public relations. Journal of Creative Communications, 14(3), 196-213

3. Xu, G., Mu, Y., & Liu, J. (2017). Inclusion of artificial intelligence in communication networks and services. ITU J. ICT Discov. Spec, 1, 1-6.

2.3 PROBLEM STATEMENT DEFINITION

- 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 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 website is built which uses this model.
- This website enables deaf and dumb people to convey their information using signs which get converted to human understandable language and text is given as output.

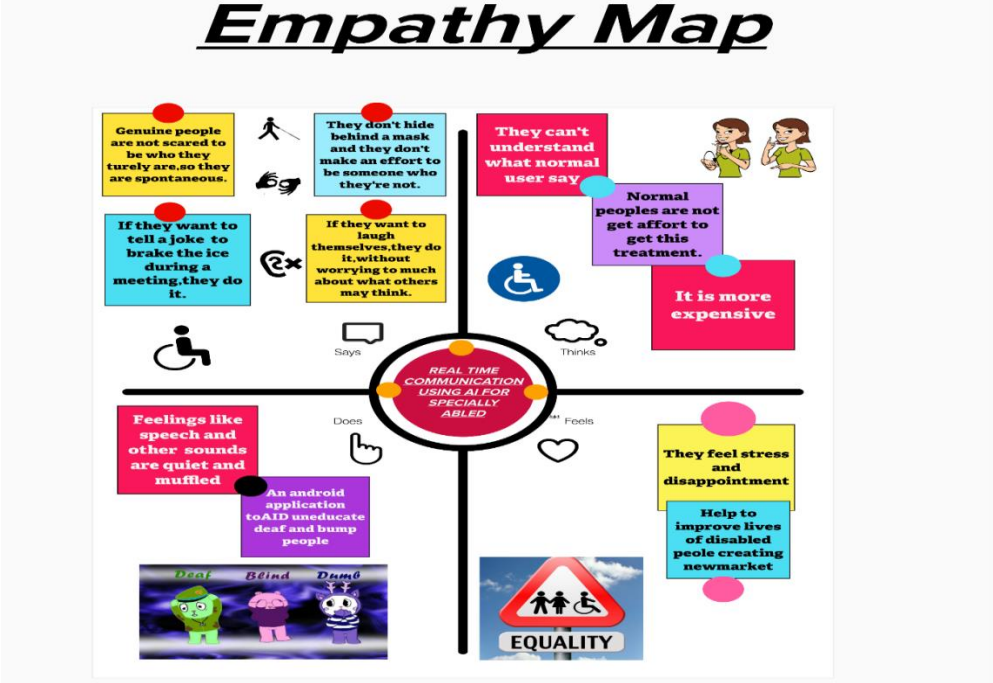
I am	Deaf and dumb person who face lots of difficulty in day to day life.
I am trying to	Speak like others and feel happy by forgetting my inabilities.
But	I feel frustrated by thinking my inability.
Because	I can’t deliver my thoughts.
Which makes me feel happy	I am feel very happy when the society recognised me and feel extremely happy when I communicate easily with others.

CHAPTER 3

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

- An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours
- It is a useful tool to helps teams better understand their users.
- Creating an effective solution requires understanding the true problem and the person who is experiencing it.
- The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



Reference:
https://app.mural.co/invitation/mural/ibm74132/1663148891949?sender=u335aaeca59037e8272905605&key=cbca06e6-bc9b-4cb3-8711-16ebcc3637f

Who are we empathizing with?

- 1)We want to understand the specially abled person
- 2)They are in a situation who finds it hard to communicate with other people on their own
- 3)Their role is about communication.

What do they need to do?

- Get familiar with new technologies
- Get familiar with sign language.

What do they see?

- App gets many downloads if it is a success
- Many people with disabilities get to communicate well

What do they say?

- Reviews and thoughts on improving the app

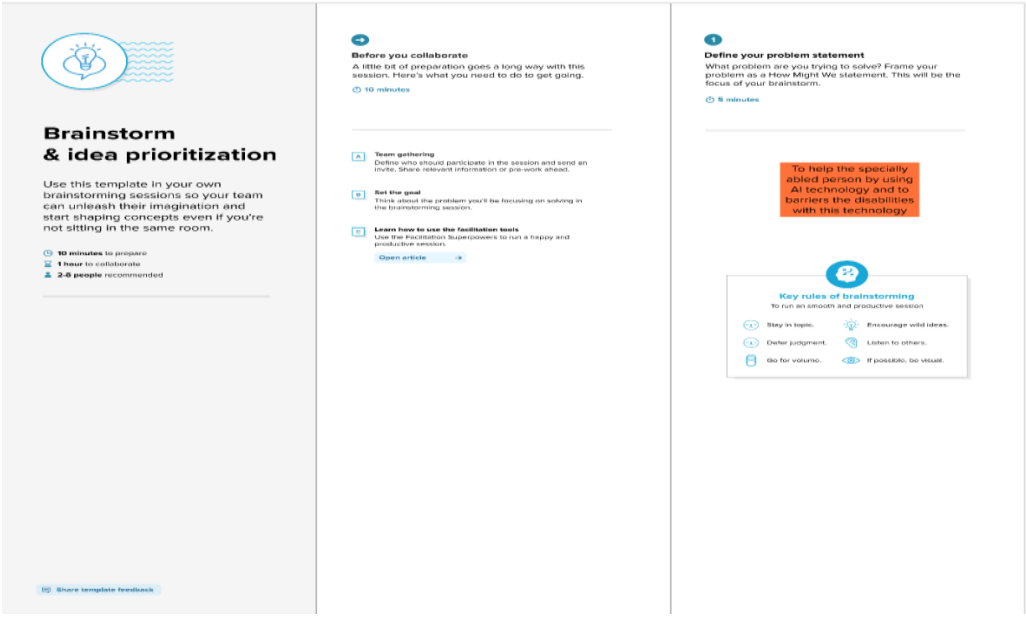
3.2 IDEATION AND BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

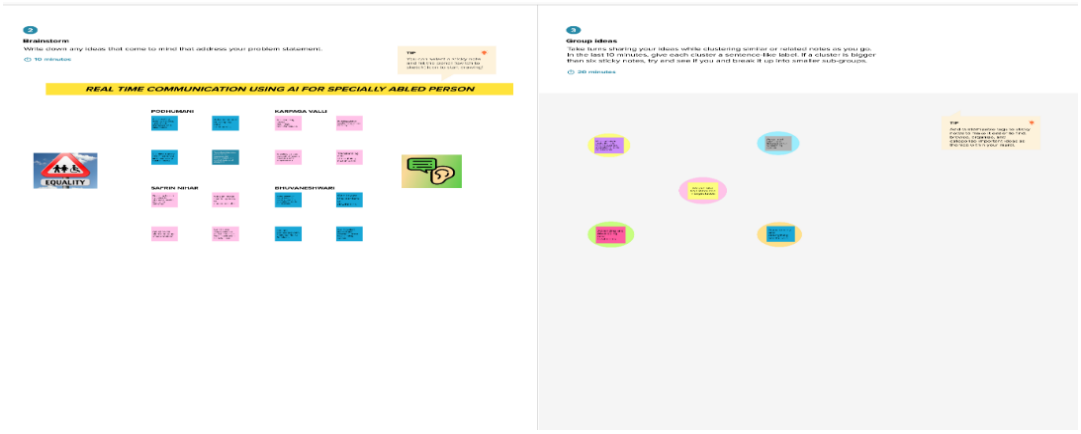
Reference:

https://app.mural.co/invitation/mural/ibm74132/1663851454750?sender=u335aaeca59037e8272905605&key=398b7a69-aa20-4f1f-91cc-bd5a20e6836d

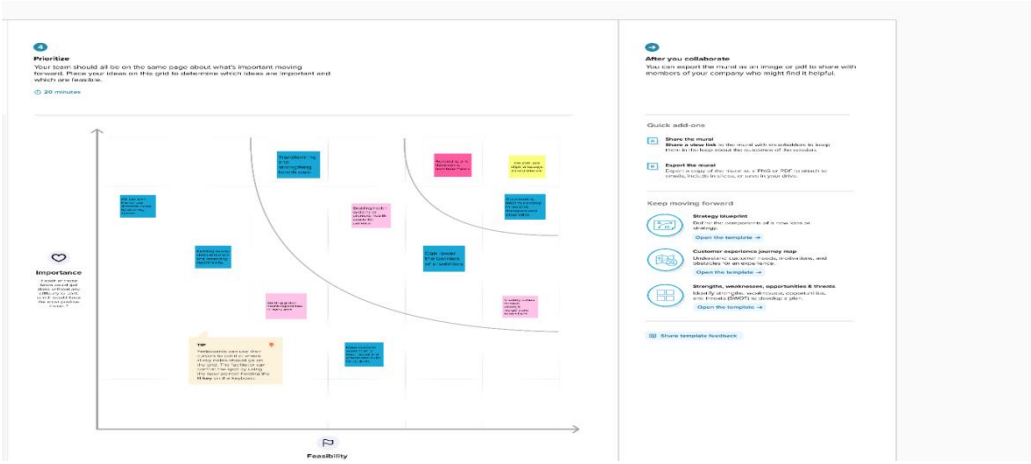
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 PROPOSED SOLUTION

1.Problem Statement (Problem to be solved)

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.

2.Idea / Solution description

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built 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.

3. Novelty / Uniqueness

Building mobile tools with data isn't as easy as importing an XML feed of your latest headlines. But if you're going to spend thousands of dollars developing a mobile app anyway, you might as well spend a little more to build a real application that helps solve problems and makes advertisers take notice.

4. Social Impact / Customer Satisfaction

These apps are using only for solve the problems. There are many different types of disabilities, and there are also many different ways in which people may use AI. The use of artificial intelligence is a boon for specially abled people. Technology had opened up new opportunities and created jobs where none had existed before, such as speech to text software that helped one woman find her voice after she was paralysed in an accident. Artificial Intelligence can help those with disabilities accomplish tasks they never thought possible; here are just a few ways we've seen AI technology impact lives: Facial recognition and predictive texting tools.

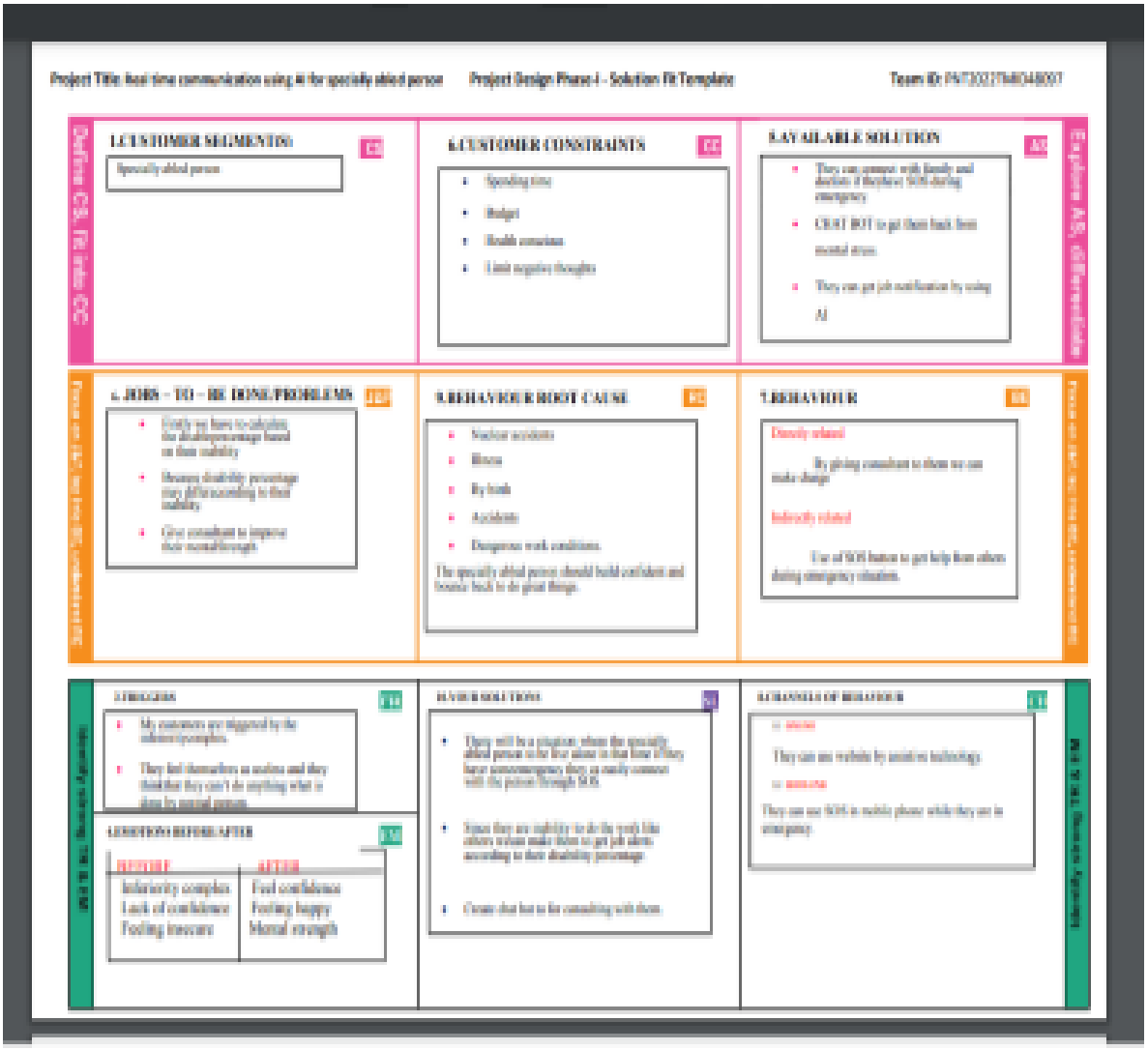
5. Business Model (Revenue Model)

Building mobile tools with data isn't as easy as importing an XML feed of your latest headlines. But if you're going to spend thousands of dollars developing a mobile app anyway, you might as well spend a little more to build a real application that helps solve problems and makes advertisers take notice.

6. Scalability of the Solution

With the help of machine tasks that usually requires human intelligence, such as voice and speech synthesis, visual perception, predictive text functionality, judgement, and a variety of other tasks, AI can assist individuals with disabilities by making a significant distinction in their ability.

3.4 PROBLEM SOLUTION FIT:



CHAPTER 4

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR-1 DATA COLLECTION

Collecting data for building our project. Creating two folders one for training and the other for testing. Images present in the training folder will be used for building the model and the testing images will be usedfor validating our model.

FR-2 MODEL BUILDING

- Initializing the model
- Adding Convolution layers
- Adding Pooling layers
- Flatten layer
- Full connection layers which include hidden layers
- Compile the model with layers we added to complete the neural network structure.

FR-3 TEST THE MODEL

- Test the model by passing an image to get predictions.
- While test the model we should make sure that the test image should meet the target size of the model,
- dimensions need to meet, and should undergo rescaling before giving it to the model.

FR-4 APPLICATION BUILDING

- Building a flask application that is used for buildingour us which in backend can be interfaced to the model to get predictions.
- Flask application requires an html page for frontendand a python file for the backend which takes care of the interface with the model.

FR-5 TRAIN CNN MODEL

- Train model
- Store y Model
- Download the Stored model to the Localsystem

FR-6 REGISTRATION

Register for IBM cloud

FR-7 TRAIN IMAGE CLASSIFICATION MODEL

- locally Train the model on IBM
- Store your Model
- Download model to local system Test model

4.2 NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements of the proposed solution

NFR-1 Usability

- Non-functional requirements are the constrains orthe requirements imposed on the system.
- They specify the quality attribute of the software.
- Non-functional requirements deal with issues likescalability, maintainability.
- Performance, portability, security, reliability, and many more.

NFR-2 Security

It provides cyber security systems with up-to-dateand relevant knowledge of Industry specific and global threats**,** which help teams make critical decisions about priorities based on what attack strategies may be used against a company

NFR-3 Reliability

- AI technology can empower people living withlimited physical mobility.
- Microsoft's AI for Accessibility program uses the potential of Artificial Intelligence to develop solutions to many physical and cognitive challengesdisabled individuals face at work and in daily life to promote social inclusion for them.

NFR-4 Performance

- AI enables people with disabilities to step into aworld where their difficulties are understood and taken into account.
- Technology adapts and helps transform the worldinto an inclusive place with artificial intelligence accessibility.

NFR-5 Availability

- Using driverless cars enables disabled people toleave the house, get around their communities, interact with people and even find jobs.
- Once autonomous vehicles are fully integrated intosociety, they could ease independent mobility, and increase accessibility adapted to each user's abilitiesand needs.

NFR-6 Scalability

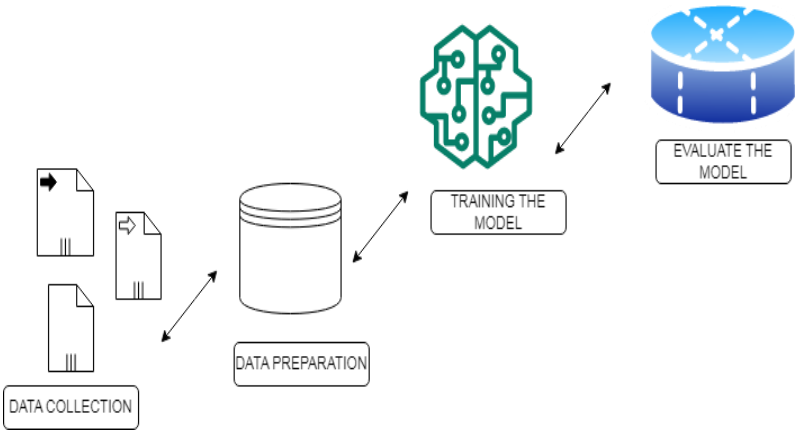
Scalability is a non-functional property of a system that describes the ability to appropriatelyhandle increasing workloads.

CHAPTER 5

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

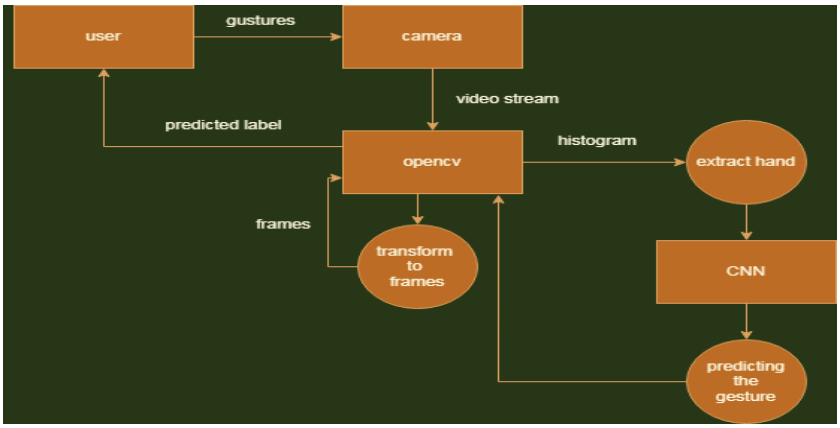
A Data flow diagram (DFD) is a traditional visual representation of the information flows within a system. A neat ad clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information.



5.2 SOLUTION AND 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, behaviour, 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.



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (web user)	Data collectionand data augmentation	USN-1	To collect the dataset it was asked to four users to wear the vision system and performevery gesture for 10 seconds while both cameras were recording in a 640x480 pixel resolution.	Artificial Neural Network	High	Sprint-1
	Data augmentation	USN-2	To improve the generalization capability of themodel it was artificially added more images with different backgrounds replacing the green Backgrounds. This way it is obtained moredata without investing too much time.	Convolution neural network	High	Sprint-1
	Creating data set	USN-3	To improve the generalization capability of themodel it was artificially added more images with different backgrounds replacing the greenbackgrounds. This way it is obtained more Data without investing too much time.	Using computer Vision OpenCv	Low	Sprint-1
	Threshold value	**USN-4**	Calculate the threshold value for every frame and determine the contours.	**Using openCV**	**Medium**	**Sprint-2**
	Training CNN	USN-5	Load the data using Image Data Generator ofkeras through which we can use the flow_from_directory function to load the train and test set data, and each of the names of the number folders will be the class names for The images loaded.	Using keras	High	Sprint-2
	Plot image function		Plotting images of data loaded.	Using Python and openCv	Low	Sprint-2
	Testing the model	USN-6	Train the model through various epochs.	It can be get through the computer vision (openCV)based on the deep learning concept	High	Sprint -3
	Visualizing the model.	USN-7	Check if everything is working as we expect itto while detecting on the live cam feed.	Understanding the concept of ANN	Medium	Sprint-3
	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Predict the Gestures	USN-8	Check if everything is working as we expect it to while detecting on the live cam feed.	Learning the concept of deep learning	Medium	Sprint-3
	Calculate weight average	USN-9	Caution of average weight of preloaded data.	Learning the concepts of opencv	High	Sprint-4
	Segmenting the hands	USN-10	Getting the max contours and the threshold image of the hand detected.	Learning the concept of deep learning	High	Sprint-4
	Detection of hand	USN-11	Running the CNN model.	Learning the concept of CNN and open cv concepts	High	Sprint-4

CHAPTER 6

6. PROJECT PLANNING AND SCHEDULING

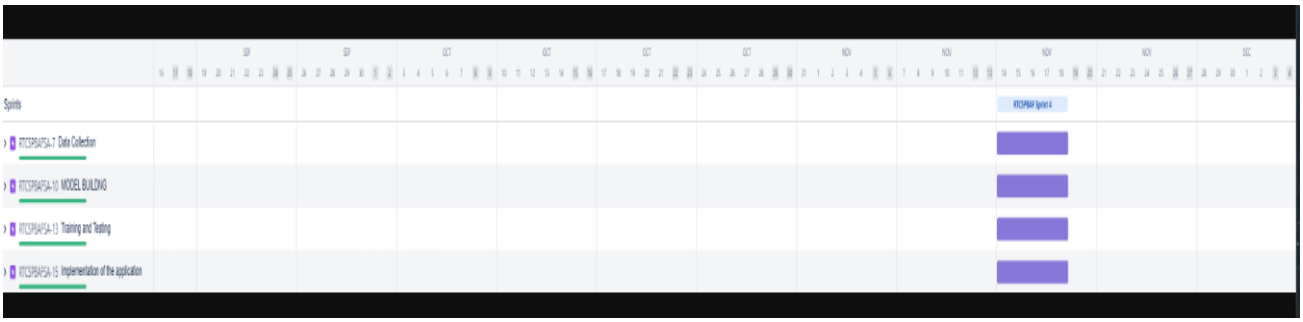
6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint- 1	Local system	USN-1	I have to collect the data for building ourproject.	2	High	J SAFRIN NIHAR K KARPAGAVALLI L BHUVANESWARI PODHUMANI
Sprint- 1	Local system	USN-2	Create train and test data folder	1	High	J SAFRIN NIHAR K KARPAGAVALLI L BHUVANESWARI P PODHUMANI
Sprint- 1	Python IDE – keras package	USN-3	We have to do Image preprocessing, shearing, rescale, zooming, etc. to make the model robust with different types of images. We will be loading all the images of thetrain and test using the flow from directory method.	2	Low	J SAFRIN NIHAR K KARPAGAVALLI L BHUVANESWARI P PODHUMANI
Sprint- 2	Neural Networks	USN-5	Model building	1	High	J SAFRIN NIHAR K KARPAGAVALLI L BHUVANESWARI PPODHUMANI
Sprint- 2	openCV	USN-5	Load The Test Image, Pre-Process ItAnd Predict	1	High	J SAFRIN NIHAR K KARPAGAVALLI L BHUVANESWARI P PODHUMANI
Sprint- 3	Python Flask	USN-6	Integrate the code with python flask.Build the HTML page. Open the navigator and run the localhost.	2	high	J SAFRIN NIHAR K KARPAGAVALLI L BHUVANESWARI P PODHUMANI
Sprint- 4	IBM cloud	USN-7	Train the model on IBMStore Model Download model to local system Deployment of the project.	4	high	J SAFRIN NIHAR K KARPAGAVALLI L BHUVANESWARI P PODHUMANI

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	29 Oct 2022	02 Oct 2022	20	02 Oct 2022
Sprint-2	20	6 Days	03 Oct 2022	07 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	08 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA



CHAPTER 7

7. CODING AND SOLUTIONING

7.1 FEATURE 1

import libraries

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D , MaxPool2D , Flatten , Dropout , BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
from tensorflow.keras.callbacks import ReduceLROnPlateau
```

To accelerate GPU growth run the below cell

```
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
```

```
/home/admin/anaconda3/envs/gpu/lib/python3.8/site-packages/tensorflow/python/client/session.py:1761: UserWarning: An interact
  warnings.warn('An interactive session is already active. This can '
```

Loading the ASL dataset

```
train_df = pd.read_csv("Dataset/sign_mnist_train.csv")
test_df = pd.read_csv("Dataset/sign_mnist_test.csv")
```

```
test = pd.read_csv("Dataset/sign_mnist_test.csv")
y = test['label']
```

```
train_df.head()
```

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778
0	3	107	118	127	134	139	143	146	150	153	...	207	207	207	207

```
train_df.head()
```

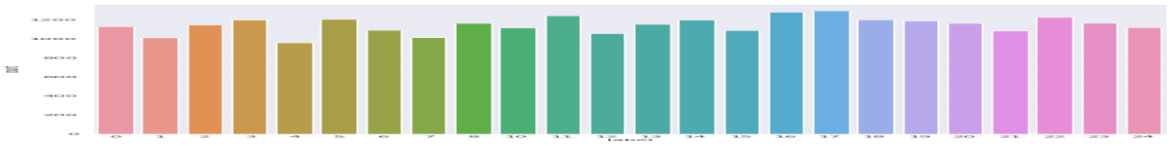
	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778
0	3	107	118	127	134	139	143	146	150	153	...	207	207	207	207
1	6	155	157	156	156	156	157	156	158	158	...	69	149	128	87
2	2	187	188	188	187	187	186	187	188	187	...	202	201	200	199
3	2	211	211	212	212	211	210	211	210	210	...	235	234	233	231
4	13	164	167	170	172	176	179	180	184	185	...	92	105	105	108

5 rows × 785 columns

Data Visualization and Preprocessing

```
plt.figure(figsize = (10,10)) # Label Count
sns.set_style("darkgrid")
sns.countplot(train_df['label'])
```

```
/home/admin/anaconda3/envs/gpu/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variab
  warnings.warn(
<AxesSubplot:xlabel='label', ylabel='count'>
```



The dataset seems balanced as for each training label , enough training examples exist

```
y_train = train_df['label']
y_test = test_df['label']
del train_df['label']
del test_df['label']
```

+ Code + Text

```
from sklearn.preprocessing import LabelBinarizer
label_binarizer = LabelBinarizer()
y_train = label_binarizer.fit_transform(y_train)
y_test = label_binarizer.fit_transform(y_test)
```

```
x_train = train_df.values
x_test = test_df.values
```

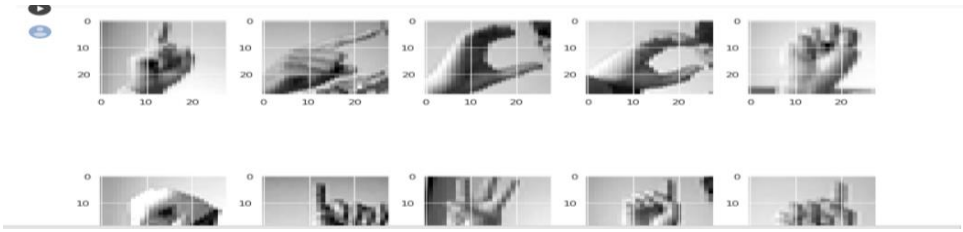
We perform a grayscale normalization to reduce the effect of illumination's differences.Moreover the CNN converges faster on [0..1] data than on [0..255].

```
[ ]  # Normalize the data
     x_train = x_train / 255
     x_test = x_test / 255
```

```
[ ]  # Reshaping the data from 1-D to 3-D as required through input by CNN's
     x_train = x_train.reshape(-1,28,28,1)
     x_test = x_test.reshape(-1,28,28,1)
```

Preview of first 10 images

```
f, ax = plt.subplots(2,5)
f.set_size_inches(10, 10)
k = 0
for i in range(2):
    for j in range(5):
        ax[i,j].imshow(x_train[k].reshape(28, 28) , cmap = "gray")
        k += 1
plt.tight_layout()
```



7.2 FEATURE 2

Data Augmentation

In order to avoid over fitting problem, we need to expand artificially our dataset. We can make your existing dataset even larger. The idea is to alter the training data with small transformations to reproduce the variation.

Approaches that alter the training data in ways that change the array representation while keeping the label the same are known as data augmentation techniques. Some popular augmentations people use are grayscales, horizontal flips, vertical flips, random crops, colour jitters, translations, rotations, and much more.

By applying just a couple of these transformations to our training data, we can easily double or triple the number of training examples and create a very robust model.

```
# With data augmentation to prevent overfitting

datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.1, # Randomly zoom image
        width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=False,  # randomly flip images
        vertical_flip=False)  # randomly flip images


datagen.fit(x_train)
```

For the data augmentation, I chose to :

- Randomly rotate some training images by 10 degrees
- Randomly Zoom by 10% some training images
- Randomly shift images horizontally by 10% of the width
- Randomly shift images vertically by 10% of the height
- I did not apply a vertical flip nor horizontal flip since it could have lead to misclassify.
- Once our model is ready, we fit the training dataset .

CONVOLUTIONAL NEURAL NETWORKS TO THE RESCUE

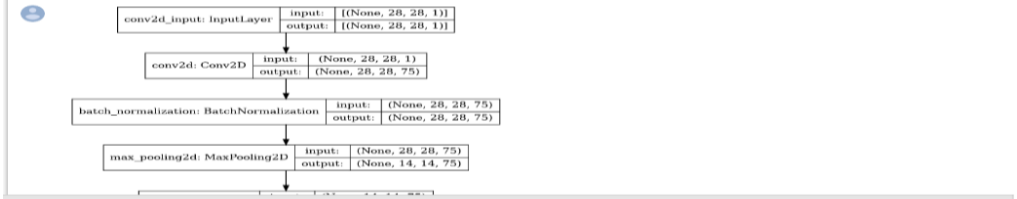
Training The Model

```
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy', patience = 2, verbose=1,factor=0.5, min_lr=0.00001)
```

```
[ ] model = Sequential()
    model.add(Conv2D(75 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input_shape = (28,28,1)))
    model.add(BatchNormalization())
    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(50 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
```

```
model = Sequential()
model.add(Conv2D(75 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input_shape = (28,28,1)))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(50 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(25 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Flatten())
model.add(Dense(units = 512 , activation = 'relu'))
model.add(Dropout(0.3))
model.add(Dense(units = 24 , activation = 'softmax'))
model.compile(optimizer = 'adam' , loss = 'categorical_crossentropy' , metrics = ['accuracy'])
```

```
plot_model(model, show_shapes=True, show_layer_names=True, to_file='model.png')
from IPython.display import Image
Image(retina=True, filename='model.png')
```



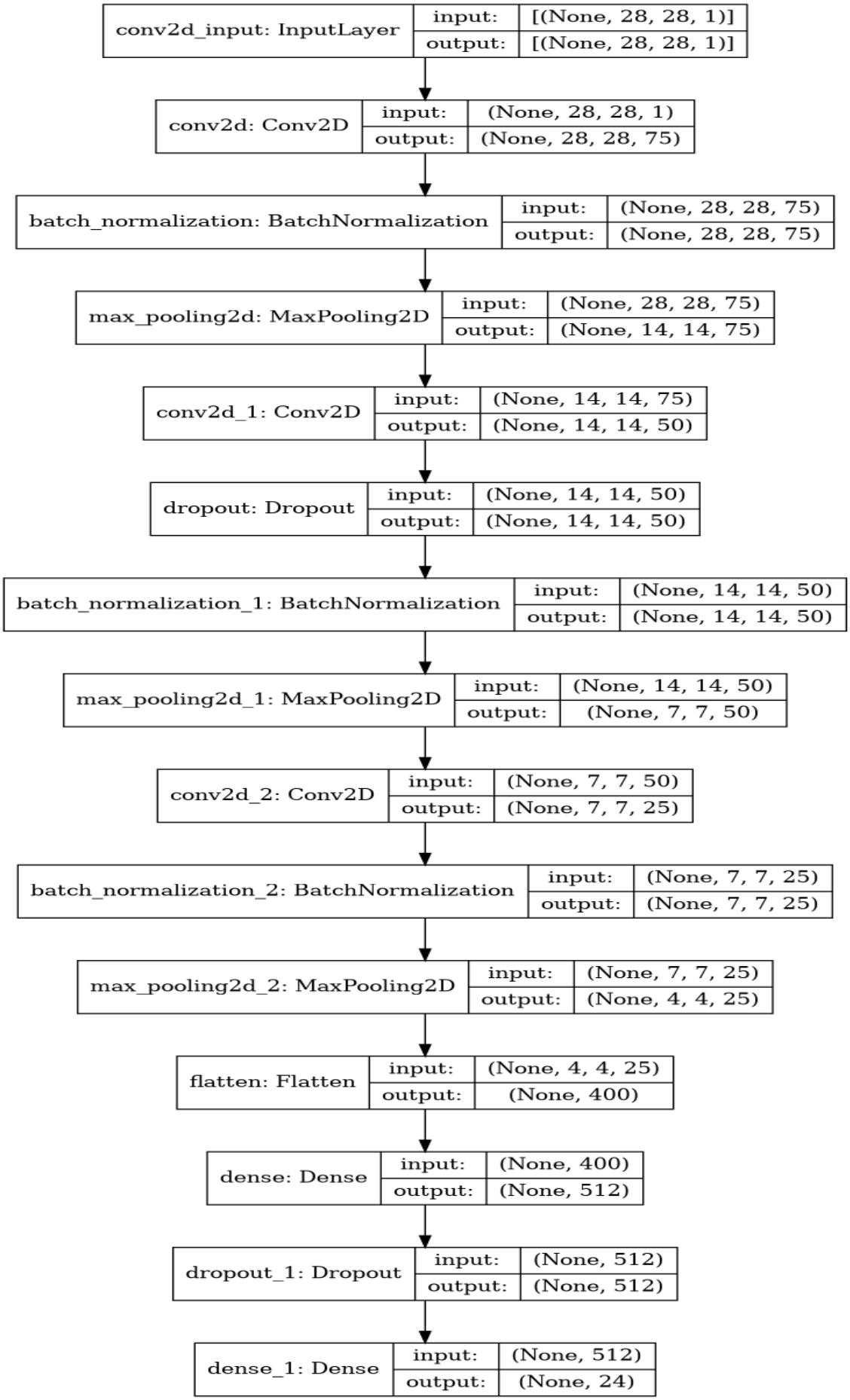
Save the model as CNNmodel.h5

```
[ ] model.save('CNNmodel.h5')
```

Analysis after Model Training

```
epochs = [i for i in range(20)]
fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
fig.set_size_inches(16,9)
```

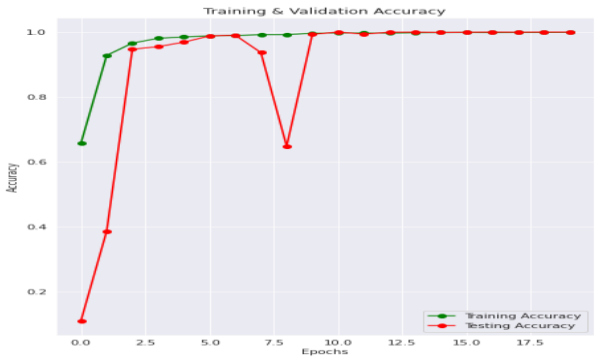
```
ax[0].set_title('Training & Validation Accuracy')
ax[0].legend()
ax[0].set_xlabel("Epochs")
ax[0].set_ylabel("Accuracy")

ax[1].plot(epochs , train_loss , 'g-o' , label = 'Training Loss')
ax[1].plot(epochs , val_loss , 'r-o' , label = 'Testing Loss')
ax[1].set_title('Testing Accuracy & Loss')
ax[1].legend()
ax[1].set_xlabel("Epochs")
ax[1].set_ylabel("Loss")
plt.show()
```

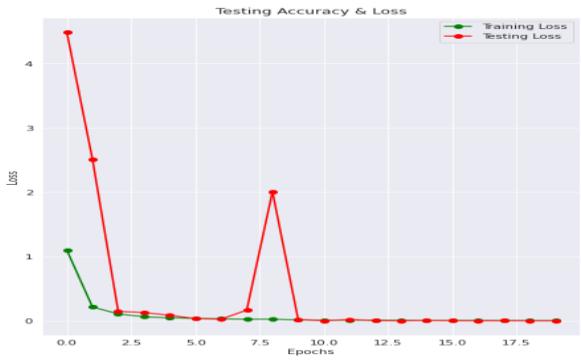

conv2d_input: InputLayer
input: [(None, 28, 28, 1)]
output: [(None, 28, 28, 1)]
conv2d: Conv2D
input: (None, 28, 28, 1)
output: (None, 28, 28, 75)
batch_normalization: BatchNormalization
input: (None, 28, 28, 75)
output: (None, 28, 28, 75)
max_pooling2d: MaxPooling2D
input: (None, 28, 28, 75)
output: (None, 14, 14, 75)
conv2d_1: Conv2D
input: (None, 14, 14, 75)
output: (None, 14, 14, 50)
dropout: Dropout
input: (None, 14, 14, 50)
output: (None, 14, 14, 50)
batch_normalization_1: BatchNormalization
input: (None, 14, 14, 50)
output: (None, 14, 14, 50)
max_pooling2d_1: MaxPooling2D
input: (None, 14, 14, 50)
output: (None, 7, 7, 50)
conv2d_2: Conv2D
input: (None, 7, 7, 50)
output: (None, 7, 7, 25)
batch_normalization_2: BatchNormalization
input: (None, 7, 7, 25)
output: (None, 7, 7, 25)
max_pooling2d_2: MaxPooling2D
input: (None, 7, 7, 25)
output: (None, 4, 4, 25)
flatten: Flatten
input: (None, 4, 4, 25)
output: (None, 400)
dense: Dense
input: (None, 400)
output: (None, 512)
dropout_1: Dropout
input: (None, 512)
output: (None, 512)
dense_1: Dense
input: (None, 512)
output: (None, 24)

CHAPTER 8

8. TESTING

8.1 TEST CASES






```
[ ]  predictions = model.predict_classes(x_test)
     for i in range(len(predictions)):
         if(predictions[i] >= 9):
             predictions[i] += 1
     predictions[:5]
```

```
/home/admin/anaconda3/envs/gpu/lib/python3.8/site-packages/tensorflow/python/keras/engine/sequential.py:450: UserWarning: `mod
  warnings.warn('`model.predict_classes()` is deprecated and '
array([ 6,  5, 10,  0,  3])
```

```
classes = ["Class " + str(i) for i in range(25) if i != 9]
print(classification_report(y, predictions, target_names = classes))
```

8.2 USER ACCEPTANCE TESTING

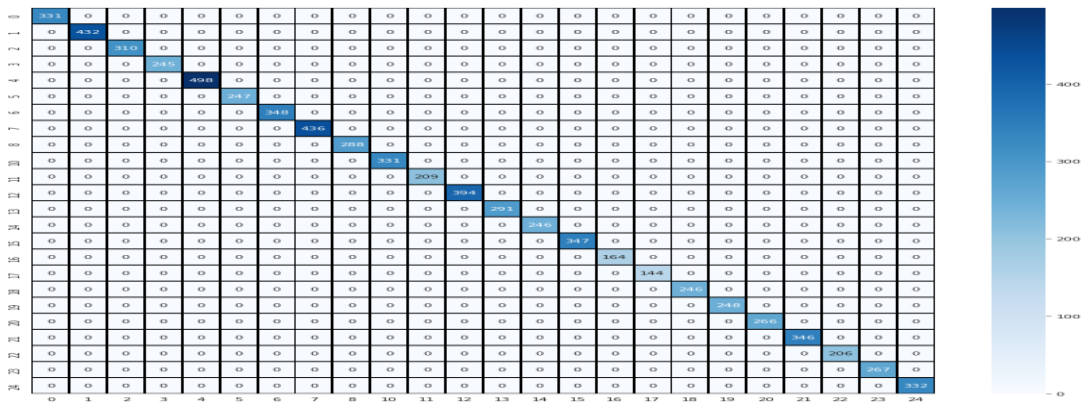
```
classes = ["Class " + str(i) for i in range(25) if i != 9]
print(classification_report(y, predictions, target_names = classes))
```

```
              precision    recall  f1-score   support

     Class 0       1.00      1.00      1.00       331
     Class 1       1.00      1.00      1.00       432
     Class 2       1.00      1.00      1.00       310
     Class 3       1.00      1.00      1.00       245
     Class 4       1.00      1.00      1.00       498
     Class 5       1.00      1.00      1.00       247
     Class 6       1.00      1.00      1.00       348
     Class 7       1.00      1.00      1.00       436
     Class 8       1.00      1.00      1.00       288
    Class 10       1.00      1.00      1.00       331
    Class 11       1.00      1.00      1.00       209
    Class 12       1.00      1.00      1.00       394
    Class 13       1.00      1.00      1.00       291
    Class 14       1.00      1.00      1.00       246
    Class 15       1.00      1.00      1.00       347
```

```
[ ]  cm = confusion_matrix(y,predictions)
```

```
[ ]  cm = pd.DataFrame(cm , index = [i for i in range(25) if i != 9] , columns = [i for i in range(25) if i != 9])
```

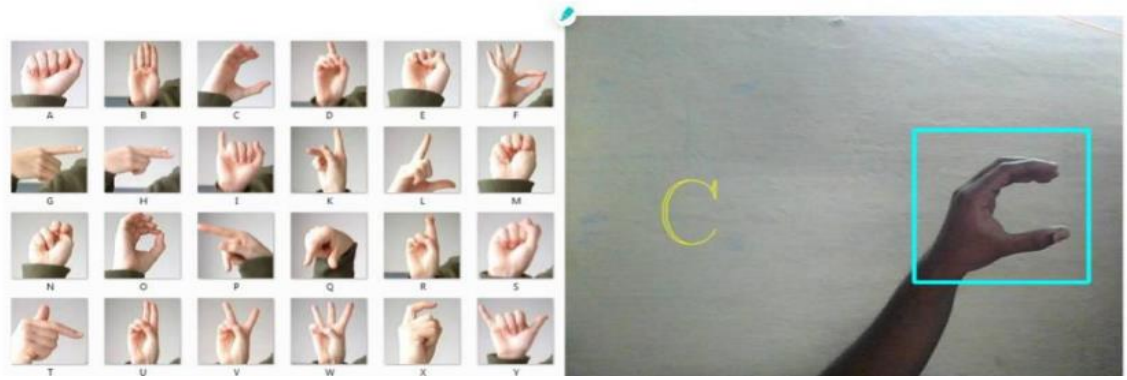
```
plt.figure(figsize = (15,15))
sns.heatmap(cm,cmap= "Blues", linecolor = 'black' , linewidth = 1 , annot = True, fmt='')
```



CHAPTER 9

9.RESULT

We are running the output in the website which shows the letters of sign language in web camera



CHAPTER 10

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Enables deaf and dumb to communicate effectively
- Decrease frustration
- Improves child – parent communication if they are disabled
- Helps child remember words
- Increase self esteem
- Provides an insight into child's world
- Builds communication easily

DISADVANTAGES

- The biggest disadvantage of communication is that it takes a lot of time to listen, speak, read, or write to someone.
- While trying to do one thing you can accidentally hurt another person's feelings by not listening or paying attention.
- Causes damage to the relationship when they not any attention.
- They are costly and difficult to be used commercially classification method are also varying from researchers.
- Problem for people who do not have full use of their hands properly.
- The normal person should feel difficulties in showing the full hand position.

CHAPTER 11

11.CONCLUSION

Real-time communication (RTC) workloads can be deployed on IBM cloud to attain scalability, elasticity, and high availability while meeting the key requirements.

Today, several customers are using cloud,its partners, and open source solutions to run RTC workloads with reduced cost and faster agility aswell as a reduced global footprint.

The reference architectures and best practices provided in this white paper can help customers successfully set up RTC workloads on IBM CLOUD and optimize the solutions to meet end user requirements while optimizing for the cloud.

In this project we finally conclude with the text as the output. The CNN model is running in backend and we get the output in local system after that we created frontend with HTML and CSS and then by using flask and by registrations through IBM we deploy the model in IBM cloud

CHAPTER 12

12. FUTURE SCOPE

- ❖ Through image recognition technology, AI understands the context objects in photos and describes photos to people.
- ❖ The speech-to-text and text-to-speech technologies helped those people who had speech impediments.
- ❖ We can develop a model for ISI word and sentence level recognition.
- ❖ This will require a system that detect changes with respect to the temporal space.
- ❖ We can develop a complete product that will help the speech and hearing impaired people and thereby reduce the communication gap.
- ❖ Through image recognition technology, AI understands the context of objects in photos and describes photos to people.
- ❖ The product in AI that narrates the entire world around them visually impaired by reading texts, describing whereabouts and the looks of the nearby people by identifying and recognizing faces andemotions.
- ❖ Autonomous vehicles are in trend and their success is due to AI technology. These vehicles can bebeneficial to people living with limited physical mobility

CHAPTER 13

13.APPENDIX

SOURCE CODE

FINAL CODE- APP.PY

https://github.com/IBM-EPBL/IBM-Project-2817-1658483653/blob/main/Project%20Development%20Phase/SPRINT%203/app.py

HTML PAGE

https://github.com/IBM-EPBL/IBM-Project-2817-1658483653/blob/main/Project%20Development%20Phase/SPRINT%203/index.html

FULL PROJECT CODE

https://github.com/IBM-EPBL/IBM-Project-2817-1658483653/tree/main/Project%20Development%20Phase/Sprint%202/Test_The_Model

GIT HUB LINK

https://github.com/IBM-EPBL/IBM-Project-2817-1658483653

PROJECT DEMO LINK

https://drive.google.com/file/d/167B6I9Pzi-HECABrQP2UdhXr7O2DtrAH/view?usp=share_link