ANNAI COLLEGE OF ENGINEERING AND TECHNOLOGY

Kovilacheri in kumbokonam

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

PROJECT REPORT

Submitted by

TEAM ID : PNT2022TMID52462

TEAM MEMBERS:

SIVARANJANI T
SUGITHA M
BHUVANESHWARI M
PRIVITHA K

ABSTRACT

Deaf and mute people use sign language to communicate. Unlike acoustically conveyed sound patterns, sign language uses hand gestures, facial expressions, body language and manual communication to convey thoughts. Due to the considerable time required in learning Sign Language, people find it difficult to communicate with specially-abled people, creating a communication gap. Hence conventionally, people face problems in recognizing sign language. Moreover, different countries have their respective form of sign gesture communication which results in non-uniformity. The ISL (Indian Sign Language) used in India is largely different from the American Sign Language used in the US, mostly because of the difference in culture, geographical and historical context. Somewhere between 138 and 300 different types of sign language are currently being used throughout the world. Sign language structure varies spatially and temporally. We have identified these as a major barrier in communication with a significant part of society. And hence, we propose to design a system that recognizes different signs and conveys the information to people. The component of any sign language consists of hand shape, motion, and place of articulation. When combined, these three components (together with palm orientation) uniquely determine the meaning of the manual sign. For sign language identification, sensorbased and vision-based methods are used In vision-based gesture recognition technology, a camera reads the movements of the human body, typically hand movements and uses these gestures to interpret sign language, whereas in sensor-based methods, real- time hand and finger movements can be monitored using the leap motion sensor. We aim at developing a scalable project where we will be considering different hand gestures to recognize the letters and words. We plan to use different deep learning models to predict the sign. This may be developed as a desktop or mobile application to enable specially abled people to communicate easily and effectively with others. However, this project can later be extended to capture the whole vocabulary of ASL (American Sign Language) through manual and non-manual signs.

Keywords: Sign language, ASL, ISL, Dynamic hand gesture recognition

1.INTRODUCTION

1.1Project 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 PROBLEM STATEMENT

The Deaf and mute community can only communicate using sign language. Sign language involves simultaneously combining hand shapes, orientations, gestures and movement of the hands, arms, or body to express the speaker's thoughts. Because of cultural, geographic and historical differences, there exists over 300 different types of sign languages around the world. The ISL (Indian Sign Language) used in India is very different from the American Sign Language used in the United States. This causes inconsistency of sign languages around the world. Moreover, learning sign language requires significant amount of time and effort. This makes it difficult for the conventional world to learn and hence interact with the deaf and mute community. According to a recent study, out of every thousand kids born, 2 to 3 of them are deaf or hard-of-hearing, and, as degrees of hearing loss go, there are 16 to 30 times more children who are identified as Deaf (having a Profound 91+dB hearing loss) than hard-ofhearing. For those deaf or hard of hearing children, only 10% of parents & family learn sign language to communicate with them. We identify this as a major barrier in communicating with a significant part of the society. 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.

OBJECTIVE AND MOTIVATION

The objective of our project is to bridge the gap and ensure the inclusion of deaf and mute community into the conventional society meanwhile ensuring an easy and effective mode of communication. We aim at designing a real time system that recognizes the sign language and expresses the same in an easy language, like English. Currently, extensive work has been done on American sign language recognition, but Indian sign language differs significantly from American sign language. ISL uses two hands for communicating (20 out of 26) whereas ASL uses single hand for communicating. Using both hands often lead to obscurity of features due to overlapping of hands. In addition to this, lack of datasets and variance in sign language with locality has resulted in restrained efforts in ISL gesture detection. Our project aims at taking the basic step in bridging the communication gap between normal people and deaf and dumb people using Indian sign language. Effective extension of this project to words and common expressions may not only make the deaf and mute people communicate faster and easier with outer world, but also provide a boost in developing autonomous systems for understanding and aiding them.

Communication between Deaf and Mute People and Normal People Chat applications have become a powerful media that assist people to communicate in different languages with each other. There are lots of chat applications that are used different people in different languages but there is not such a chat application that has facilitated to communicate with sign languages. The developed system is based on Sinhala Sign language. The system has included four main components as text messages are converted to sign messages, voice messages are converted to sign messages, sign messages are converted to text messages and sign messages are converted to voice messages. Google voice recognition API has used to develop speech character recognition for voice messages. The system has been trained for the speech and text patterns by using some text parameters and signs of Sinhala Sign language is displayed by emojis. Those emojis and signs that are included in this system will bring the normal people closer to the disabled people. This is a 2-way communication system, but it uses pattern of gesture recognition which is not very reliable in getting appropriate output.

Intelligent Sign Language Recognition

Using Image Processing Computer recognition of sign language is an important research problem for enabling communication with hearing impaired people. This project introduces an efficient and fast algorithm for identification of the number of fingers opened in a gesture representing an alphabet of the Binary Sign Language. The system does not require the hand to be perfectly aligned to the camera. The project uses image processing system to identify, especially English alphabetic sign language used by the deaf people to communicate. The basic objective of this project is to develop a computer based intelligent system that will enable dumb people significantly to communicate with all other people using their natural hand gestures. The idea consisted of designing and building up an intelligent system using image processing, machine learning and artificial intelligence concepts to take visual inputs of sign language's hand gestures a generate easily recognizable form of outputs. Hence the objective of this project is to develop an intelligent system which can act as a translator between the sign language and the spoken language dynamically and can make the communication between people with hearing impairment and normal people both effective and efficient. The system is we are implementing for Binary sign language, but it can detect any sign language with prior image processing.

Sign Language Recognition Using Image Processing One of the major drawbacks of our society is the barrier that is created between disabled or handicapped persons and the normal person. Communication is the only medium by which we can share our thoughts or convey the message but for a person with disability (deaf and mute) faces difficulty in communication with normal person. For many deaf and dumb people, sign language is the basic means of communication. Sign language recognition (SLR) aims to interpret sign languages automatically by a computer in order to help the deaf communicate with hearing society conveniently. Our aim is to design a system to help the person who trained the hearing impaired to communicate with the rest of the world using sign language or hand gesture recognition techniques. In this system, feature detection and feature extraction of hand 23 gesture is done with the help of SURF algorithm using image processing. All this work is done using MATLAB software. With the help of this algorithm, a person can easily train a deaf and mute

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.

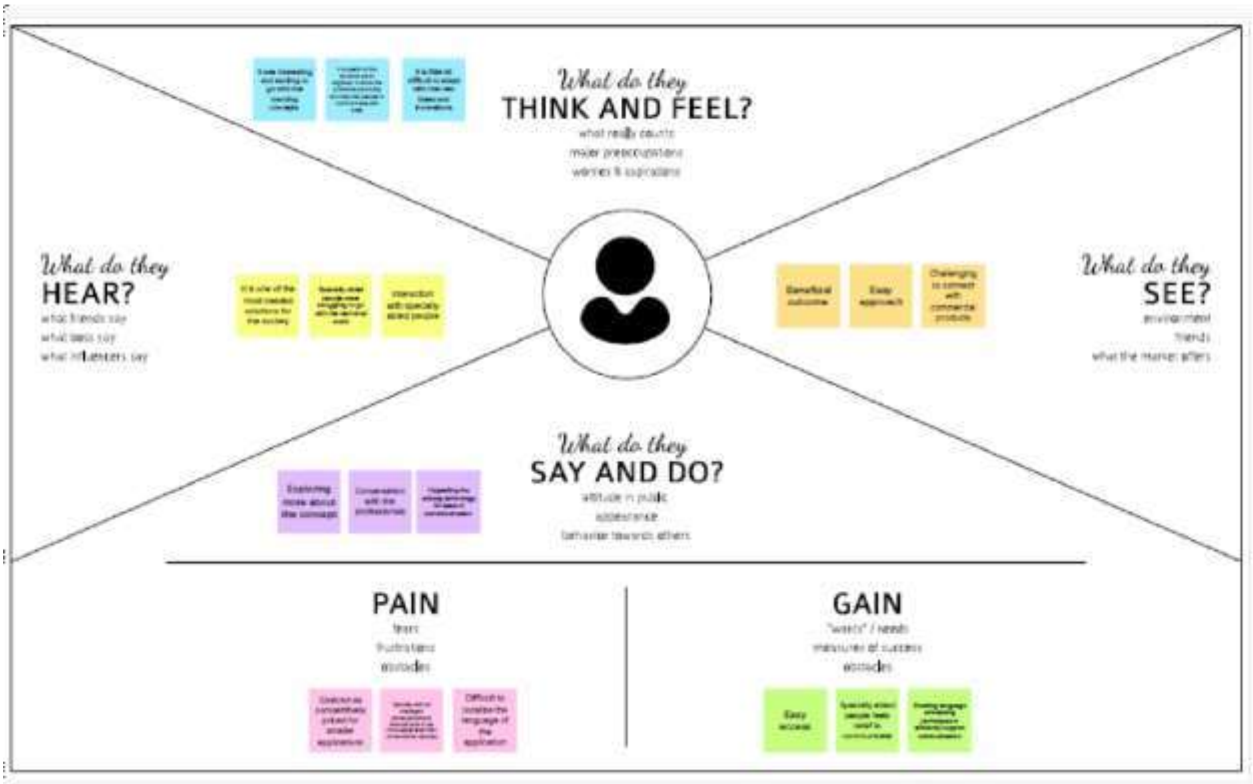
3.Xu, G., Mu, Y., & Liu, J. (2017). Inclusion of artificial intelligence in communication networks and services. ITU J. ICT Discov. Spec, 1, 1-6.

2.3 Problem Statement Definition

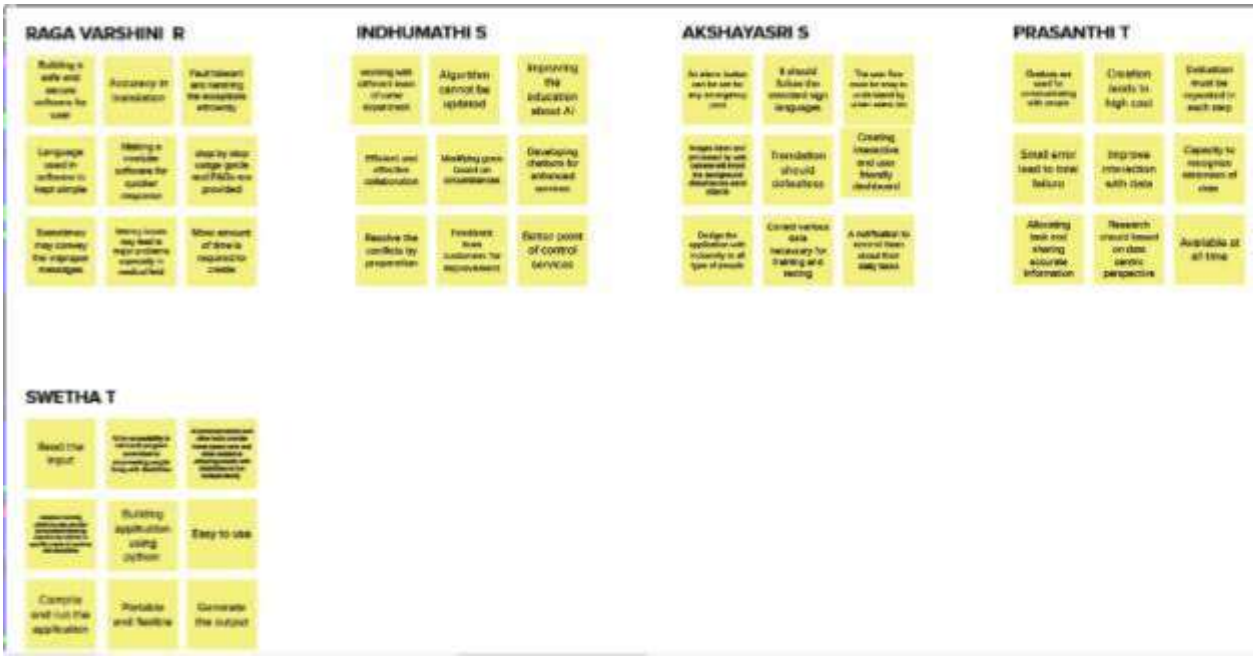
In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deafmute and a normal person has always been a challenging task. It is very difficult for mute people to 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 isgivenasoutput.

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & brainstorming

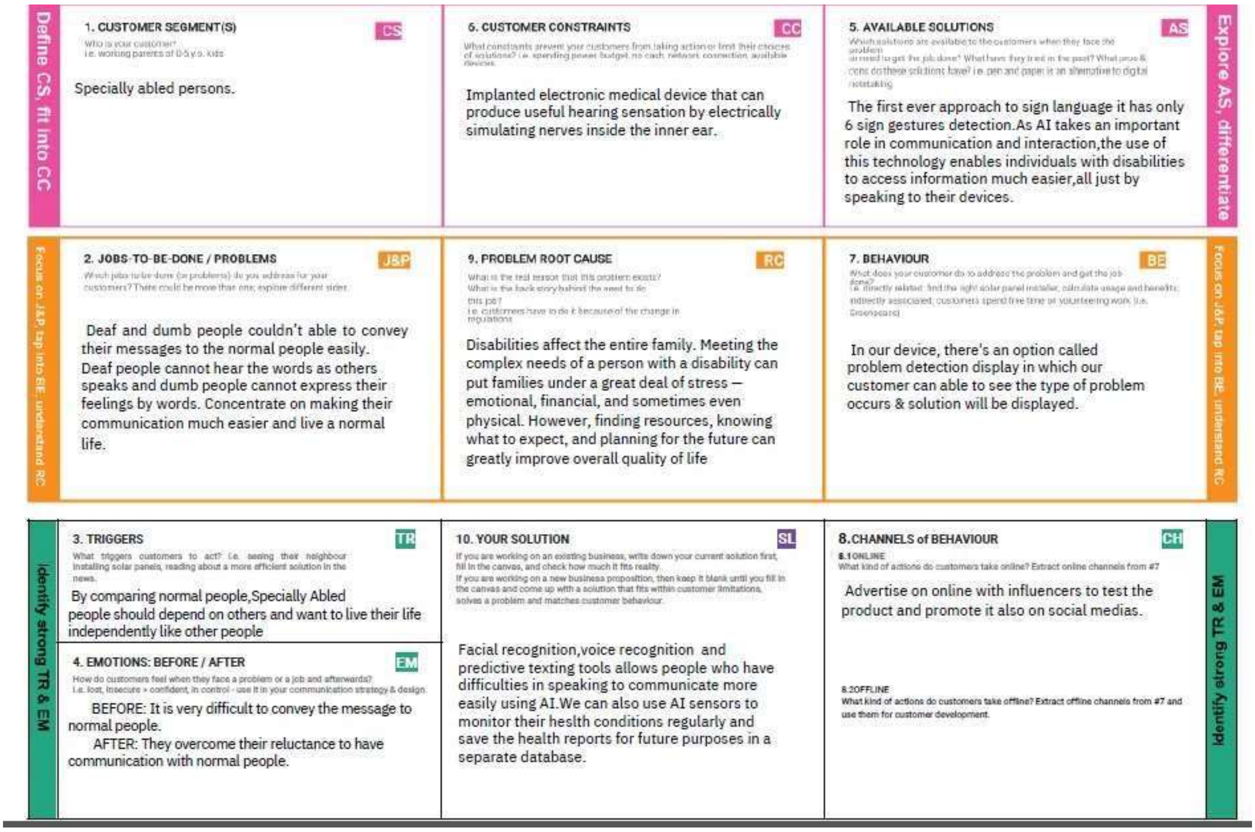

RAGA VARSHINI R
Accuracy in translation
INDHUMATHI S
Algorithm cannot be updated
Improving the education about AI
Developing chatbots for enhanced services
Resolve the conflicts by prevention
Feedback from customers for improvement
Better point of control services
AKSHAYASRI S
It should follow the standard sign language
Translation should be flawless
Creating interactive and user friendly dashboard
Design the application with inclusivity to all type of people
Collect various data necessary for training and testing
A notification to remind them about their daily tasks
PRASANTHI T
Creation leads to high cost
Evaluation must be repeated in each step
Small error lead to total failure
Improve interaction with data
Allocating task and sharing accurate information
Research should based on data centric perspective
Available at all time
SWETHA T
Read the input
Building application using python
Easy to use
Compile and run the application
Portable and flexible
Generate the output

3.3 PROPOSED SOLUTION

Themotiveofourapplicationistomakedeaf-dumbpeoplecommunicateeasilywiththepeoplebythehelp of real-time system.

S.No	Parameter	Description
1	Problem statement (problem to be solved)	□ To solve the issues of deaf-dumb people to communicate with the people to make them feel confident
2	Idea/Solution description	□ Converting sign language into voice and text in the desired language (two-way communication) using Convolutional Neural Network technology.
3	Novelty/Uniqueness	□ Upgrading our solution by implementing an alert system using Big Panda algorithm for improvement
4	Social Impact/Customer Satisfaction	• To reduce the risk of losing their lives. • It increases the scope for career development. • It will smash all the barriers and will help to enhance their skills in a positive manner.
5	Business Model(financial Benefit)	• We will collaborate with multi deaf-dumb organizations to out spread the application. • Here we give most of the basic features at free cost but they have to pay if they need more advanced features.
6	Scalability of Solution	• It has very less complexity for the user • Encoding the errors and decoding with better accuracy.

3.3PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Specially abled persons.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Implanted electronic medical device that can produce useful hearing sensation by electrically simulating nerves inside the inner ear.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

The first ever approach to sign language it has only 6 sign gestures detection.As AI takes an important role in communication and interaction,the use of this technology enables individuals with disabilities to access information much easier,all just by speaking to their devices.

Explore AS, differentiate

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? There could be more than one; explore different sides.

Deaf and dumb people couldn't able to convey their messages to the normal people easily. Deaf people cannot hear the words as others speaks and dumb people cannot express their feelings by words. Concentrate on making their communication much easier and live a normal life.

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?
i.e. customers have to do it because of the change in regulations.

Disabilities affect the entire family. Meeting the complex needs of a person with a disability can put families under a great deal of stress — emotional, financial, and sometimes even physical. However, finding resources, knowing what to expect, and planning for the future can greatly improve overall quality of life

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

In our device, there's an option called problem detection display in which our customer can able to see the type of problem occurs & solution will be displayed.

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

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

By comparing normal people,Specially Abled people should depend on others and want to live their life independently like other people

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

BEFORE: It is very difficult to convey the message to normal people.

AFTER: They overcome their reluctance to have communication with normal people.

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.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Facial recognition,voice recognition and predictive texting tools allows people who have difficulties in speaking to communicate more easily using AI.We can also use AI sensors to monitor their heslth conditions regularly and save the health reports for future purposes in a separate database.

8.CHANNELS of BEHAVIOUR CH

8.1ONLINE
What kind of actions do customers take online? Extract online channels from #7

Advertise on online with influencers to test the product and promote it also on social medias.

8.2OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Identify strong TR & EM

4 REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

- System is presented as black box
- Hearing impaired is the person that performs the signs
- Normal hearing is the passive user of the system The System Requirements Can Be Specified

1. Hearing impaired person should be able to perform sign that represent digit number
2. Hearing impaired person should be able to perform sign that represent alphabet letter 29
3. Hearing impaired person should be able to perform sign that represent word
4. Hearing impaired person should be able to perform sign that represent sentence
5. Hearing impaired person should be able to see the translation of sign to text
6. Hearing impaired person should be able to change the component (number/alphabet or word/sentence) for which translation to speech is provided

NORMAL FLOW

1. User comes in front of camera and performs the alphabet letter
2. System analyzes the performed sign
3. System shows the sign meaning as text and speech

ALTERNATIVE FLOWS

□ System indicates that user is not within field of view of Kinect

1. System shows that user is not detected
2. User enters the field of view
3. System shows that user is detected □ Sign not recognized

1. System does not react to indicate thatsign was not recognized 2. User performs again the alphabet letter until it is recognized

□ Enabling speech for this component:

1. Enable speech component

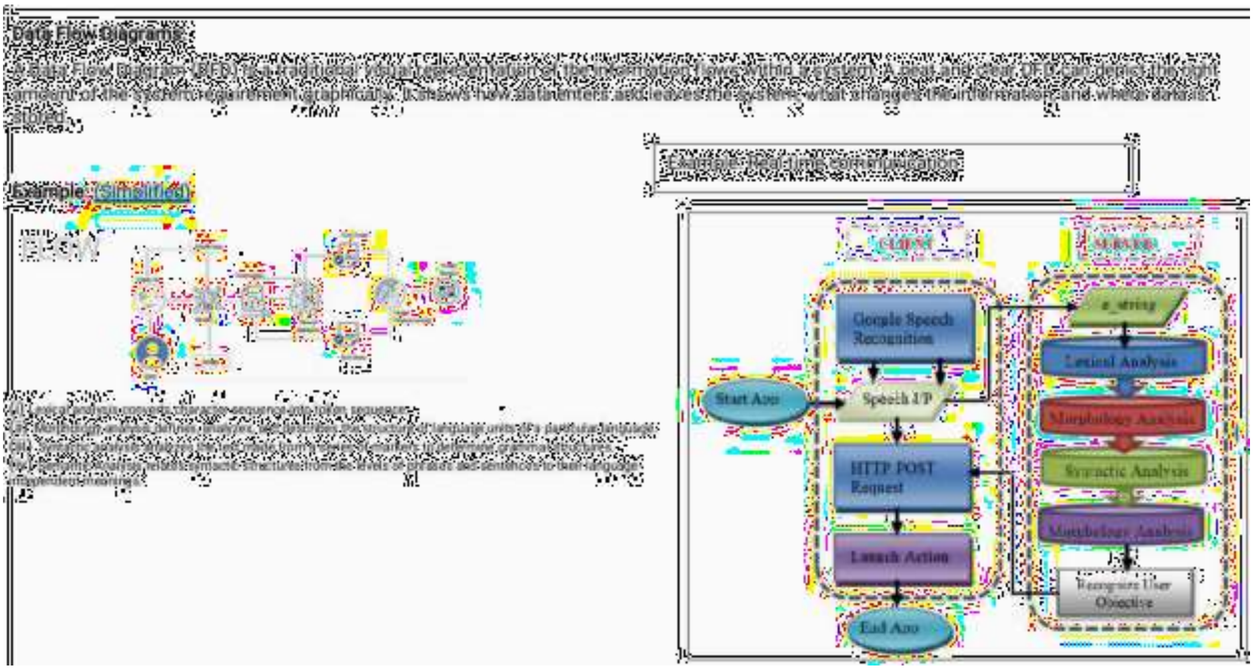
4.2 NON FUNCTIONAL REQUIREMENT

FRNo.	Non-FunctionalRequirement	Description
NFR-1	**Usability**	The designed system is easy to use for speciallyabledpersonsasitisportableand platformindependent.
NFR-2	**Security**	Convertedinformationusingsignsintospeechis accessed only by the user.
NFR-3	**Reliability**	Systemistestedwithlargenumberofdataand Providesinsight into issues.
NFR-4	**Performance**	QuickLaunchtimeofapplicationandfasterinconverting signs into speech
NFR-5	**Availability**	Providesautomaticrecoveryand Useraccess.
NFR-6	**Scalability**	Standard network condition the device shouldconvertinformationwithinsecond.

5 PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

Adataflowdiagramisatraditionalvisualrepresentationoftheinformationflowwithina system. It shows how data enters and leaves the system. It uses defined symbols like rectangles, circles and arrows, plus short text labels,to show data inputs, outputs, storage points and the routes between each destination.



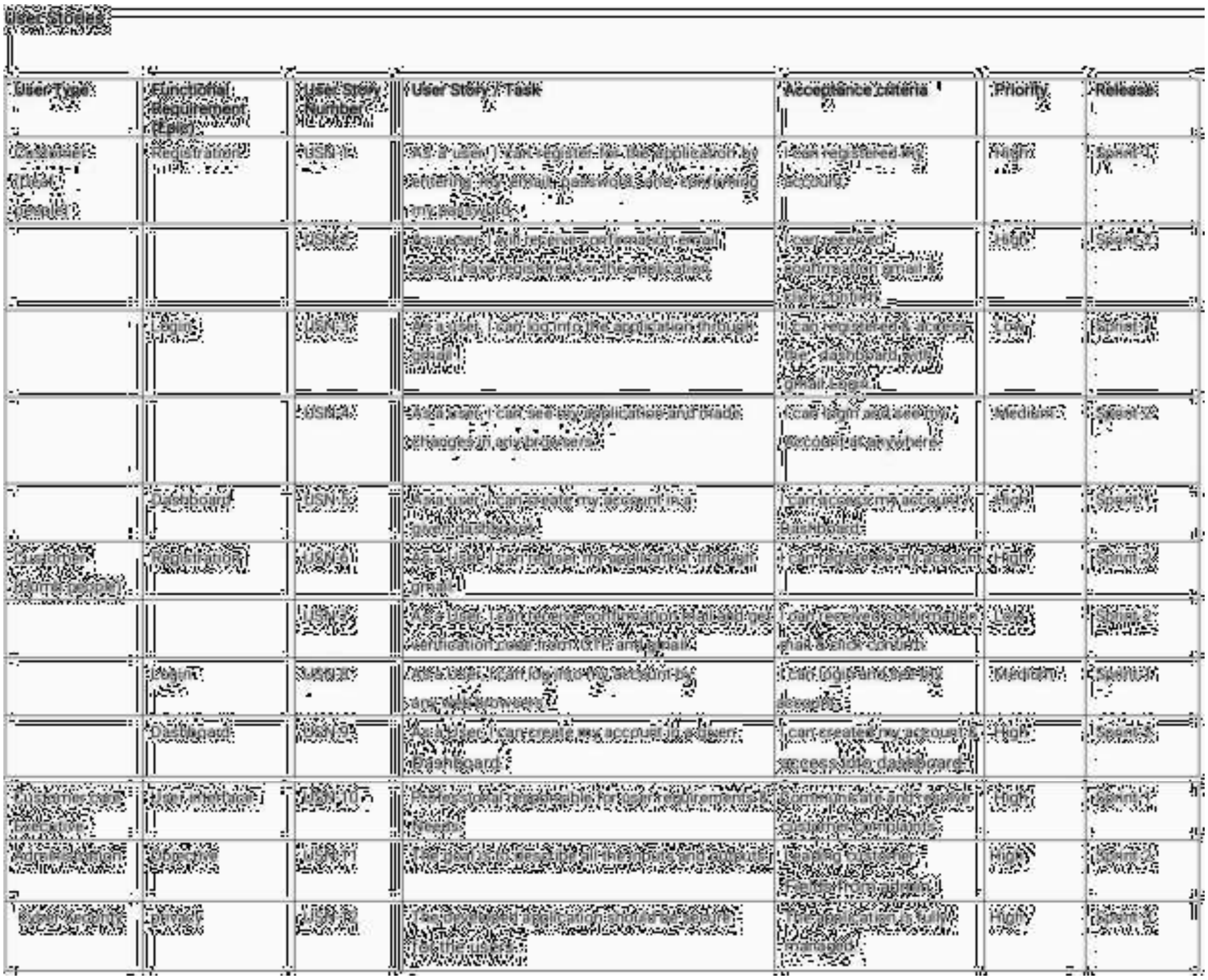
Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and 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, and where data is stored.

Example: (Simplified)

Example: Real-time communication

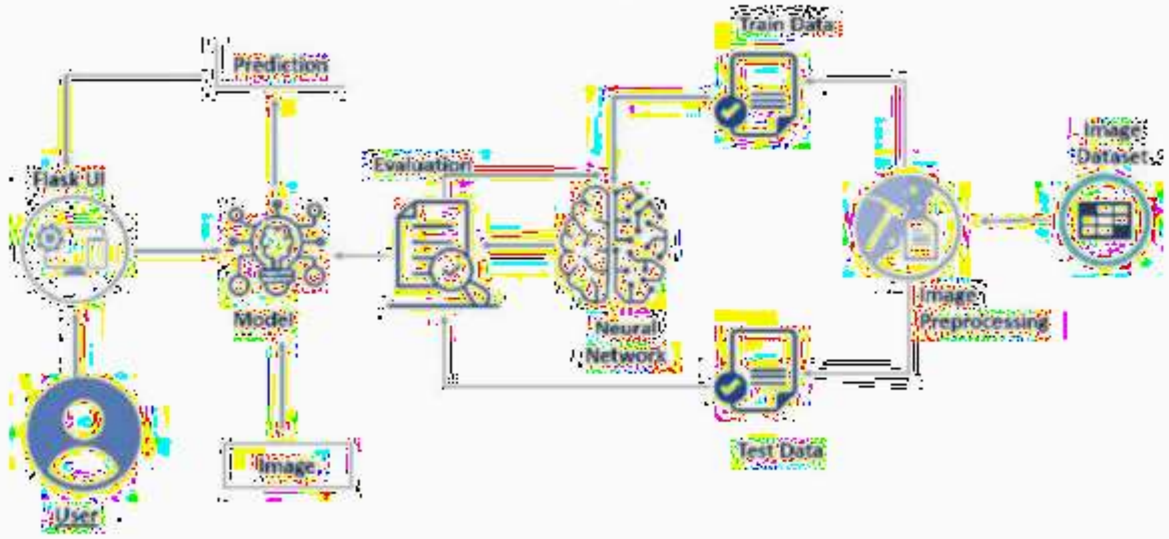
(i) Lexical analysis converts character sequence into token sequence.
(ii) Morphology analysis defines, analyzes, and describes the structure of language units of a particular language.
(iii) Syntactic analysis analyzes the text made from a sequence of markers to determine grammatical structures.
(iv) Semantic Analysis relates syntactic structures from the levels of phrases and sentences to their language independent meanings.



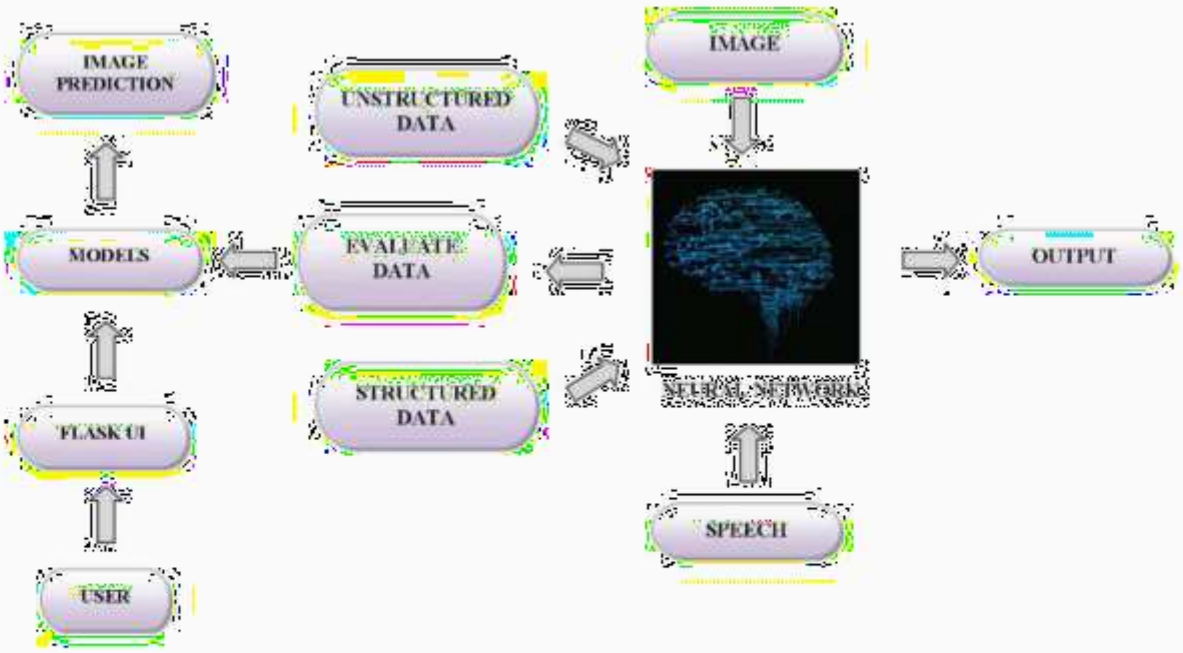
User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Deaf people)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can registered my account	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can received confirmation email & click confirm	High	Sprint-2
	Login	USN-3	As a user, I can log into the application through Gmail	I can registered & access the dashboard with gmail Login	Low	Sprint-1
		USN-4	As a user, I can see my application and trade changes in any browsers	I can login and see my account at anywhere	Medium	Sprint-2
	Dashboard	USN-5	As a user, I can create my account in a given dashboard	I can access my account / dashboard	High	Sprint-1
Customer (Dumb people)	Registration	USN-6	As a user, I can register my application through gmail	I can registered my account	High	Sprint-2
		USN-7	As a User, I can receive confirmation mail and get verification code from OTP and gmail	I can received confirmation mail & click confirm	Low	Sprint-2
	Login	USN-8	As a user, I can log into my account by any web browsers	I can login and see my account	Medium	Sprint-1
	Dashboard	USN-9	As a user, I can create my account in a given Dashboard	I can created my account & accessible dashboard	High	Sprint-3
Customer care Executive	User interface	USN-10	Professional responsible for user requirements & Needs	Communicate and resolve customer complaints	High	Sprint-1
Administration	Objective	USN-11	The goal is to describe all the inputs and outputs	Reading customer Feedback from admin	High	Sprint-2
Cyber security	privacy	USN-12	The developed application should be secure for the users	The application is fully managed	High	Sprint-3

5.2 SOLUTION AND TECHINICAL ARCHITECHTURE



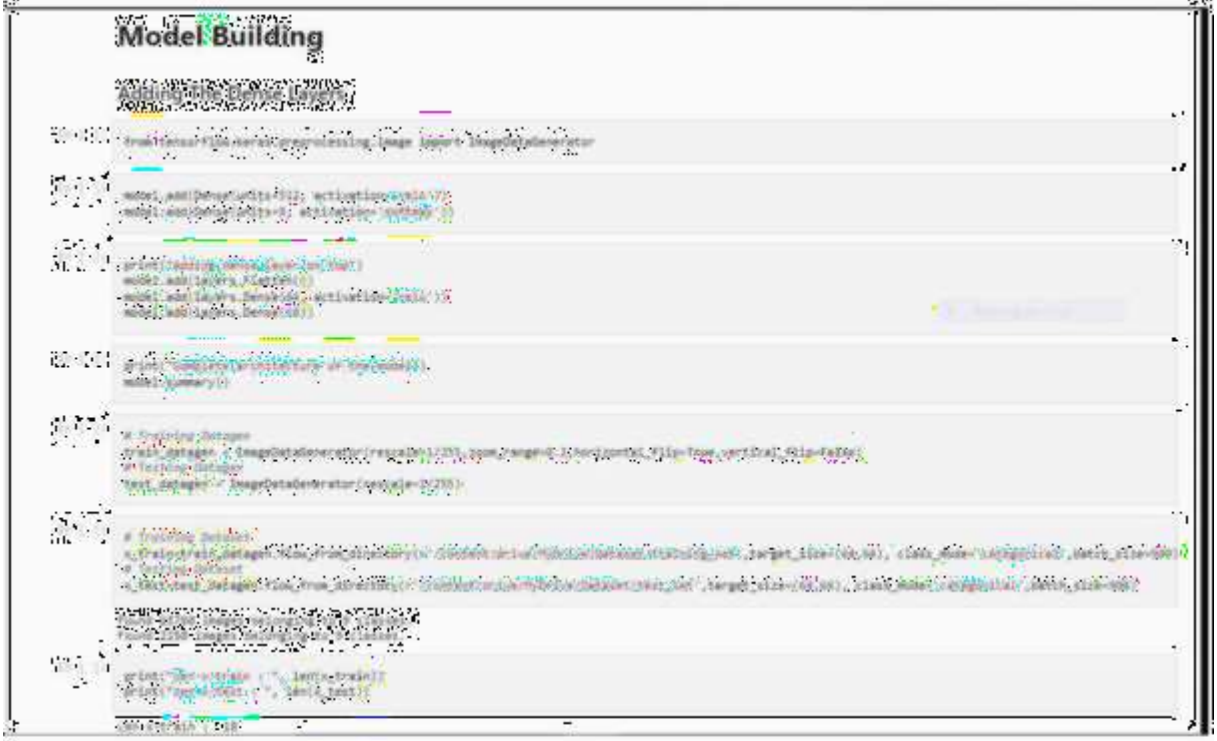
SOLUTION ARCHITECTURE



6 SPRINT DELIVERY PLAN

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Sivaranjani T Privitha K Sukitha M Bhuvamneshwari M
Sprint-1	Registration	USN-2	As a user, I will receive confirmation emailonce I have registered for the application	1	High	Sivaranjani T Sugitha M Bhuvaneshwari M Privitha k
Sprint-2	Registration	USN-3	As a user, I can register for the application through phone number	2	Medium	Sivaranjnai T Sugitha M Bhuaveshwari M Privitha K
Sprint-2	User interface	USN-4	Professional responsible for userrequirements & needs	2	Medium	Privitha K Sugitha M Bhuvaneshwari m Sivaranjani T
Sprint-3	Login	USN-5	As a user, I can log into the applicationby entering email & password	1	High	Sivaranjani T Privitha k Sugitha M Bhuvaneshwari M
Sprint-3	Dashboard	USN-6	As a user, I must receive any updates orpop ups in my dashboard	2	High	Bhuvaneshwari M Sivaranjani T Sugitha M Privitha K
Sprint-4	Details	USN-7	As a user, I should get notification about the progress and any updates via email orsms	1	Medium	Privitha k Bhuaveshwari M Sugitha M Sivaranjani T
Sprint-4	Privacy	USN-8	The developed application should be secure forthe users	2	High	Sugitha M Bhuvaneshwari m Sivaranjani t Privitha k

7 CODING AND SOLUTIONING(Explain the features added in the project along with code



Model Building

Adding The Dense Layers

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
model.add(Dense(units=512, activation='relu'))
model.add(Dense(units=5, activation='softmax'))
```

```
print('Adding dense layers [illegible]')
model2.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))
```

```
print("complete architecture of the model")
model.summary()
```

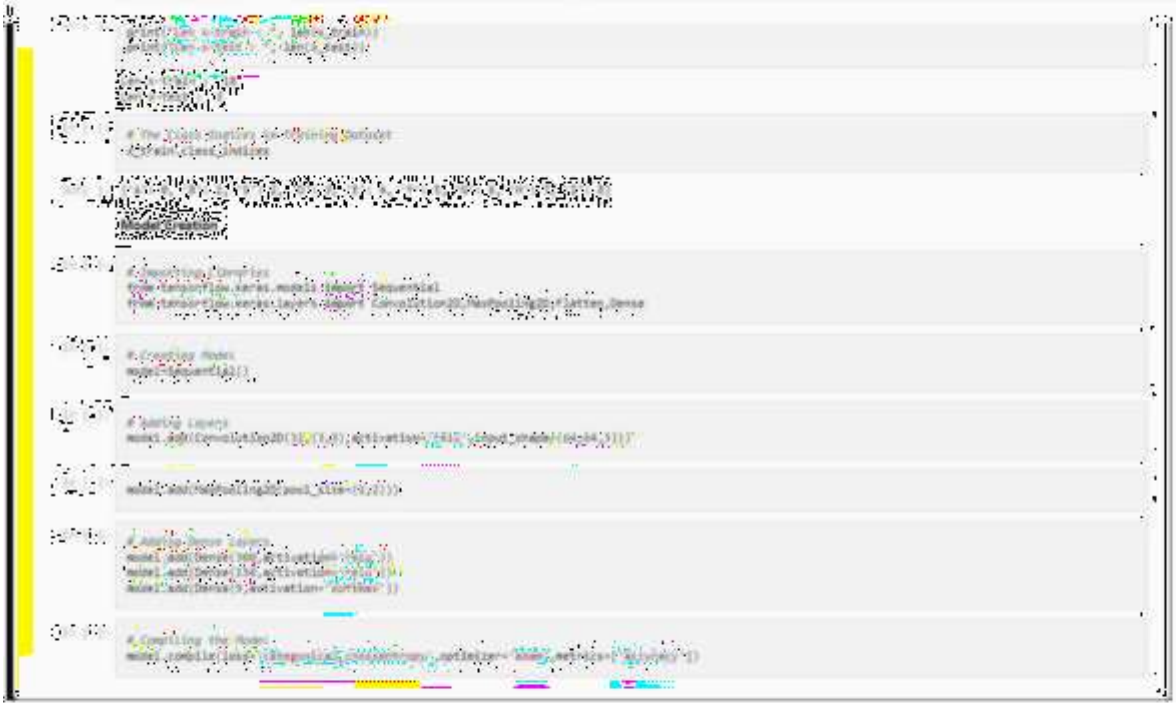
```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'[illegible]',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'[illegible]',target_size=(64,64),class_mode='categorical',batch_size=900)
```

```
Found [illegible] images belonging to 5 classes.
Found [illegible] images belonging to 5 classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train : [illegible]
```



```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train : [illegible]
Len x-test : [illegible]
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
[illegible]
```

Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

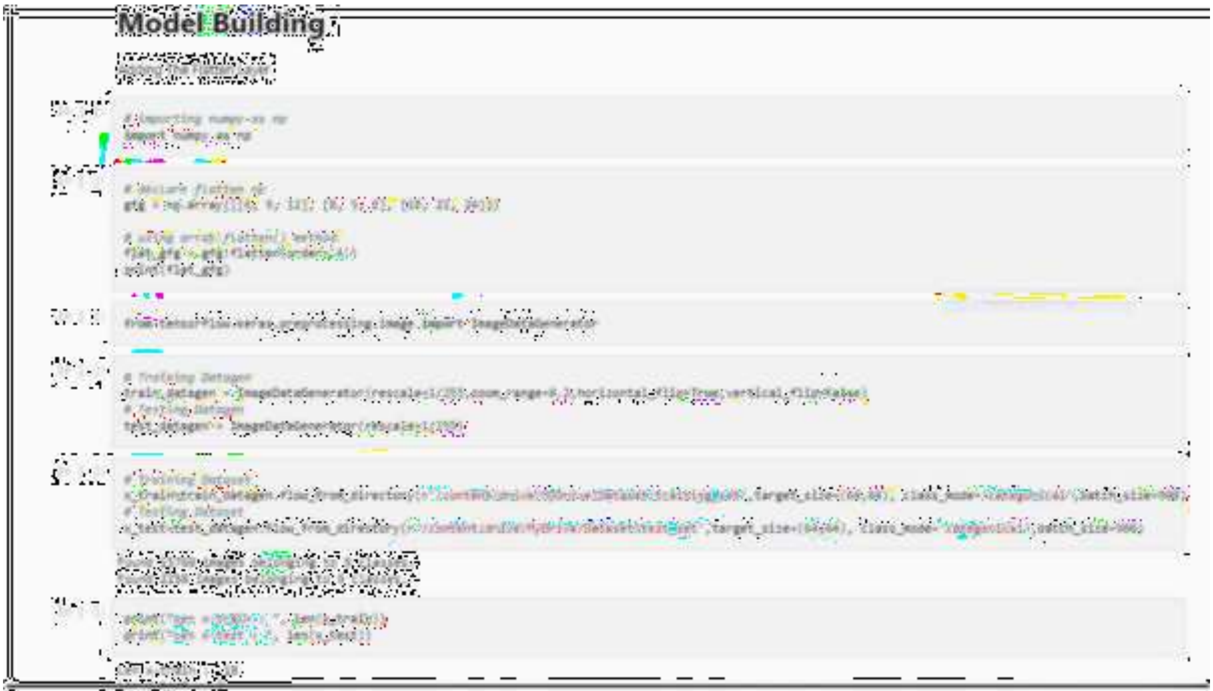
```
# Creating Model
model=Sequential()
```

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(5,activation='softmax'))
```

```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```



Model Building

Adding The Flatten Layer

```
# Importing numpy as np
import numpy as np
```

```
# Declare flatten np
grd = np.array([[illegible]])

# using array.flatten() method
flat_grd = grd.flatten(order='A')
print(flat_grd)
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

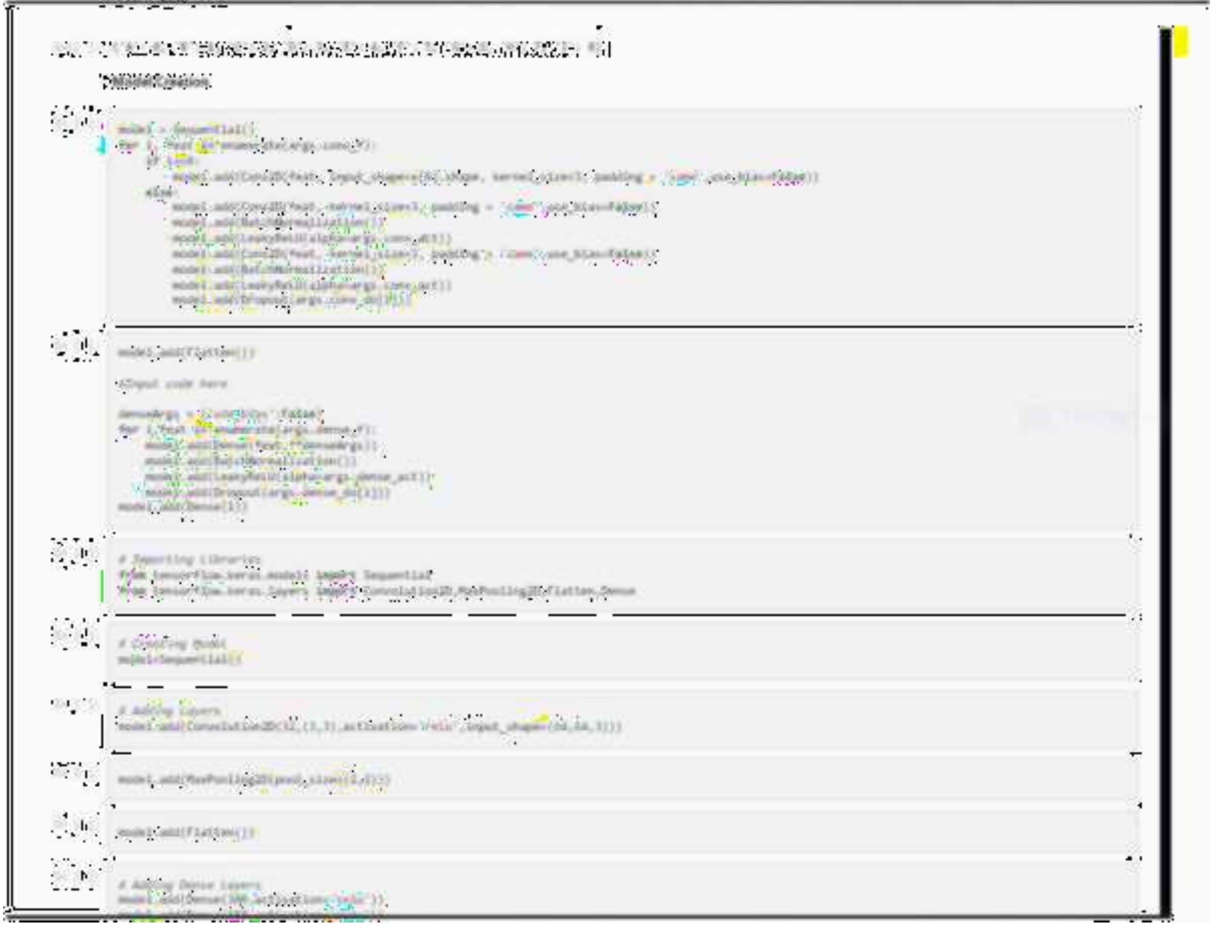
```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'[illegible]',target_size=(64,64), class_mode='categorical',batch_size=[illegible]
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'[illegible]',target_size=(64,64), class_mode='categorical',batch_size=[illegible])
```

```
Found [illegible] images belonging to [illegible] classes.
Found [illegible] images belonging to [illegible] classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train : [illegible]
```



Model Creation

```
model = Sequential()
for i, feat in enumerate(args.conv_f):
    if i==0:
        model.add(Conv2D(feat, input_shape=x[0].shape, kernel_size=3, padding = 'same',use_bias=False))
    else:
        model.add(Conv2D(feat, kernel_size=3, padding = 'same',use_bias=False))
        model.add(BatchNormalization())
        model.add(LeakyReLU(alpha=args.conv_act))
        model.add(Conv2D(feat, kernel_size=3, padding = 'same',use_bias=False))
        model.add(BatchNormalization())
        model.add(LeakyReLU(alpha=args.conv_act))
        model.add(Dropout(args.conv_do[i]))
```

```
model.add(Flatten())

#Input code here

denseArgs = {'use_bias': False}
for i,feat in enumerate(args.dense_f):
    model.add(Dense(feat, **denseArgs))
    model.add(BatchNormalization())
    model.add(LeakyReLU(alpha=args.dense_act))
    model.add(Dropout(args.dense_do[i]))
model.add(Dense(1))
```

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
# Creating Model
model=Sequential()
```

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Flatten())
```

```
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
```

Model Building

Adding The Pooling Layer

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
import numpy as np
from keras.models import Sequential
from keras.layers import MaxPooling2D
```

```
# define input image
image = np.array([[2, 2, 7, 3],
                  [9, 4, 6, 1],
                  [8, 5, 2, 4],
                  [3, 1, 2, 6]])
image = image.reshape(1, 4, 4, 1)
```

```
# define model containing just a single max pooling layer
model = Sequential(
        [MaxPooling2D(pool_size = 2, strides = 2)])

# generate pooled output
output = model.predict(image)
```

```
# print output image
output = np.squeeze(output)
print(output)
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=[illegible])
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=[illegible])
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=[illegible])
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=[illegible])
```

```
Found [illegible] images belonging to [illegible] classes.
Found [illegible] images belonging to [illegible] classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
[illegible]
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
[illegible]
```

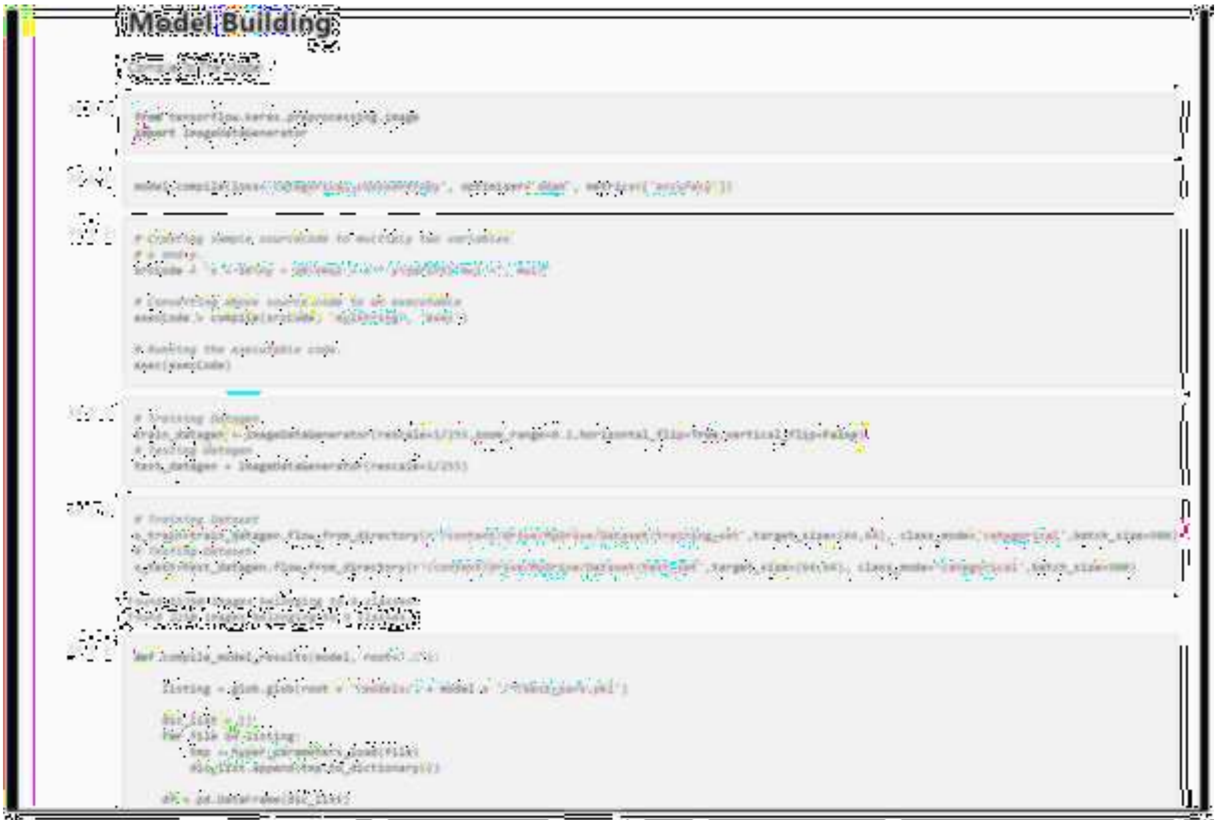
Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
# Creating Model
model=Sequential()
```

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```



Model Building

Compile the model

```
from tensorflow.keras.preprocessing.image
import ImageDataGenerator
```

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

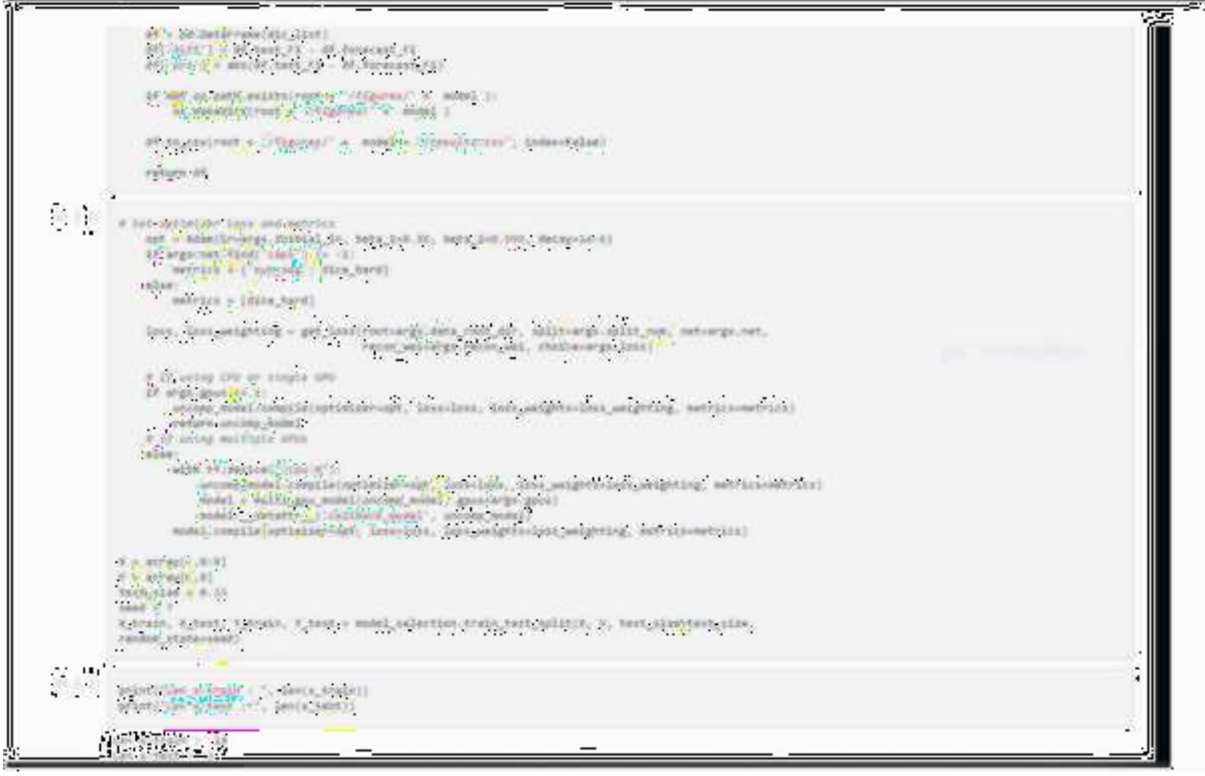
```
# Creating simple sourcecode to multiply two variables
# x and y.
srcCode = 'x = 10\ny = 20\nmul = x * y\nprint("mul =", mul)'

# Converting above source code to an executable
execCode = compile(srcCode, 'mulstring', 'exec')

# Running the executable code.
exec(execCode)
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found [illegible] images belonging to [illegible] classes.
Found [illegible] images belonging to [illegible] classes.
```

```
def compile_model_results(model, root='./'):

    listing = glob.glob(root + 'models/' + model + '/*hist_*.pkl')

    dic_list = []
    for file in listing:
        tmp = hyper_parameters.load(file)
        dic_list.append(tmp.to_dictionary())

    df = pd.DataFrame(dic_list)
```

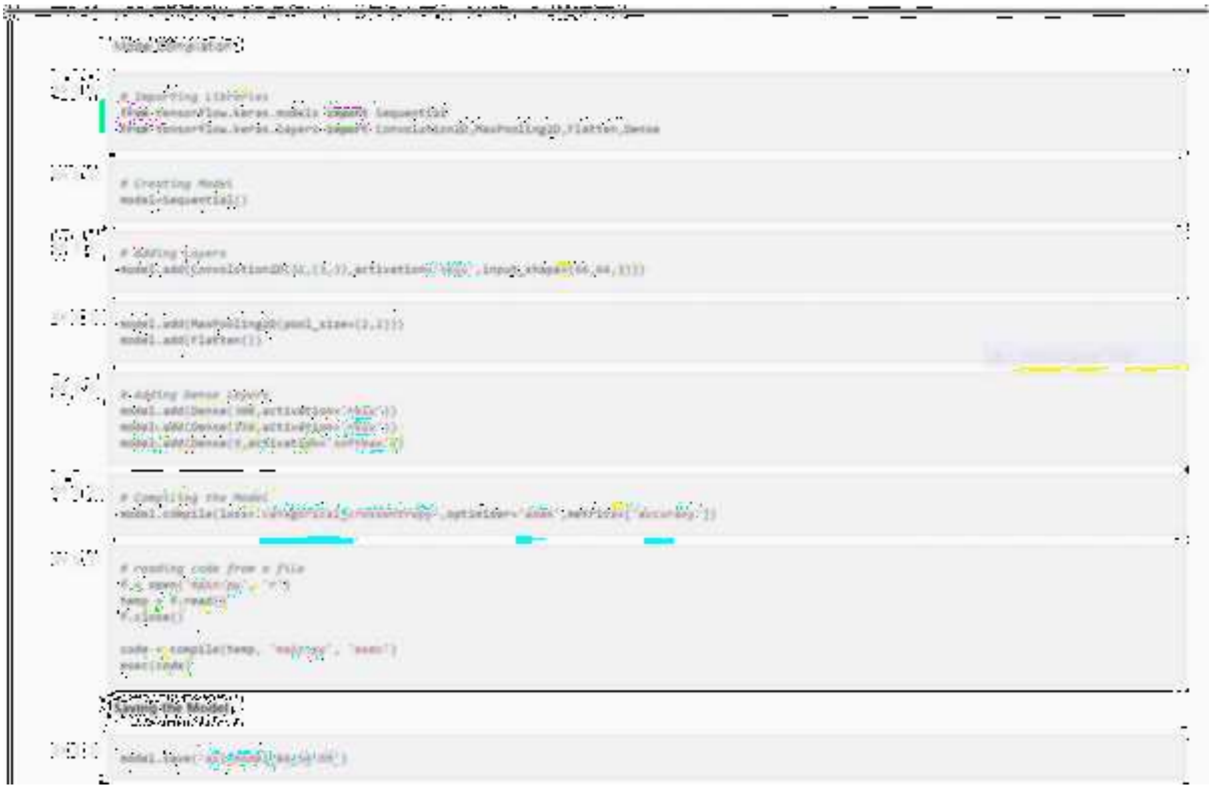


```
    df = pd.DataFrame(dic_list)
    df['diff'] = df.test_f1 - df.forecast_f1
    df['abs_diff'] = abs(df.test_f1 - df.forecast_f1)

    if not os.path.exists(root + '/figures/' + model ):
        os.makedirs(root + '/figures/' + model )

    df.to_csv(root + '/figures/' + model + '/results.csv', index=False)

    return df
```

```
# set-optimizer loss and metrics
    opt = Adam(lr=args.initial_lr, beta_1=0.99, beta_2=0.999, decay=1e-6)
    if args.net.find('caps') != -1:
        metrics = {'out_seg': dice_hard}
    else:
        metrics = [dice_hard]

    loss, loss_weighting = get_loss(root=args.data_root_dir, split=args.split_num, net=args.net,
                                    recon_wei=args.recon_wei, choice=args.loss)

    # If using CPU or single GPU
    if args.gpus <= 1:
        uncomp_model.compile(optimizer=opt, loss=loss, loss_weights=loss_weighting, metrics=metrics)
        return uncomp_model
    # If using multiple GPUs
    else:
        with tf.device("/cpu:0"):
            uncomp_model.compile(optimizer=opt, loss=loss, loss_weights=loss_weighting, metrics=metrics)
            model = multi_gpu_model(uncomp_model, gpus=args.gpus)
            model.__setattr__('callback_model', uncomp_model)
        model.compile(optimizer=opt, loss=loss, loss_weights=loss_weighting, metrics=metrics)

X = array[:,0:8]
Y = array[:,8]
test_size = 0.33
seed = 7
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(X, Y, test_size=test_size,
random_state=seed)
```

```
print("len x_train : ", len(x_train))
print("len x_test : ", len(x_test))
```

```
len x_train :  [illegible]
len x_test :  [illegible]
```



Model Compilation

```
# Importing libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
# Creating Model
model=Sequential()
```

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

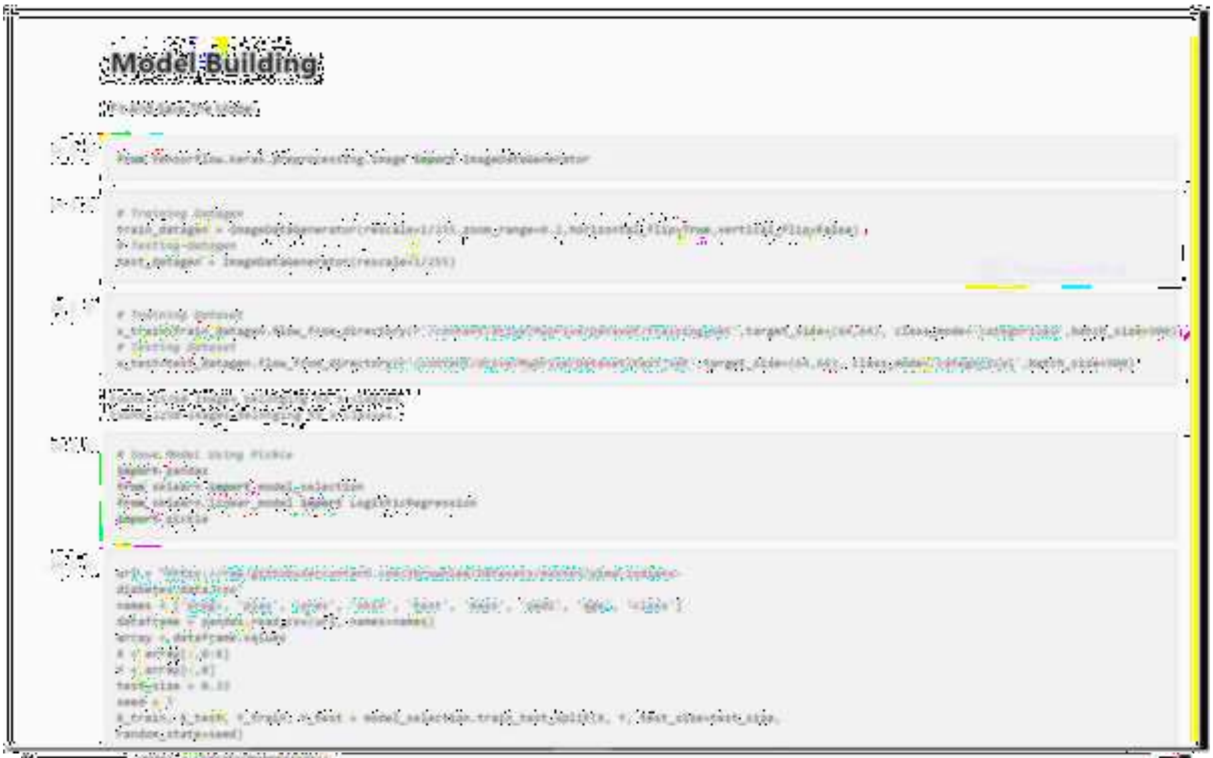
```
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense([illegible],activation='softmax'))
```

```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
# reading code from a file
f = open('main.py', 'r')
temp = f.read()
f.close()

code = compile(temp, 'main.py', 'exec')
exec(code)
```

Saving the Model

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



Model Building

[illegible]

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

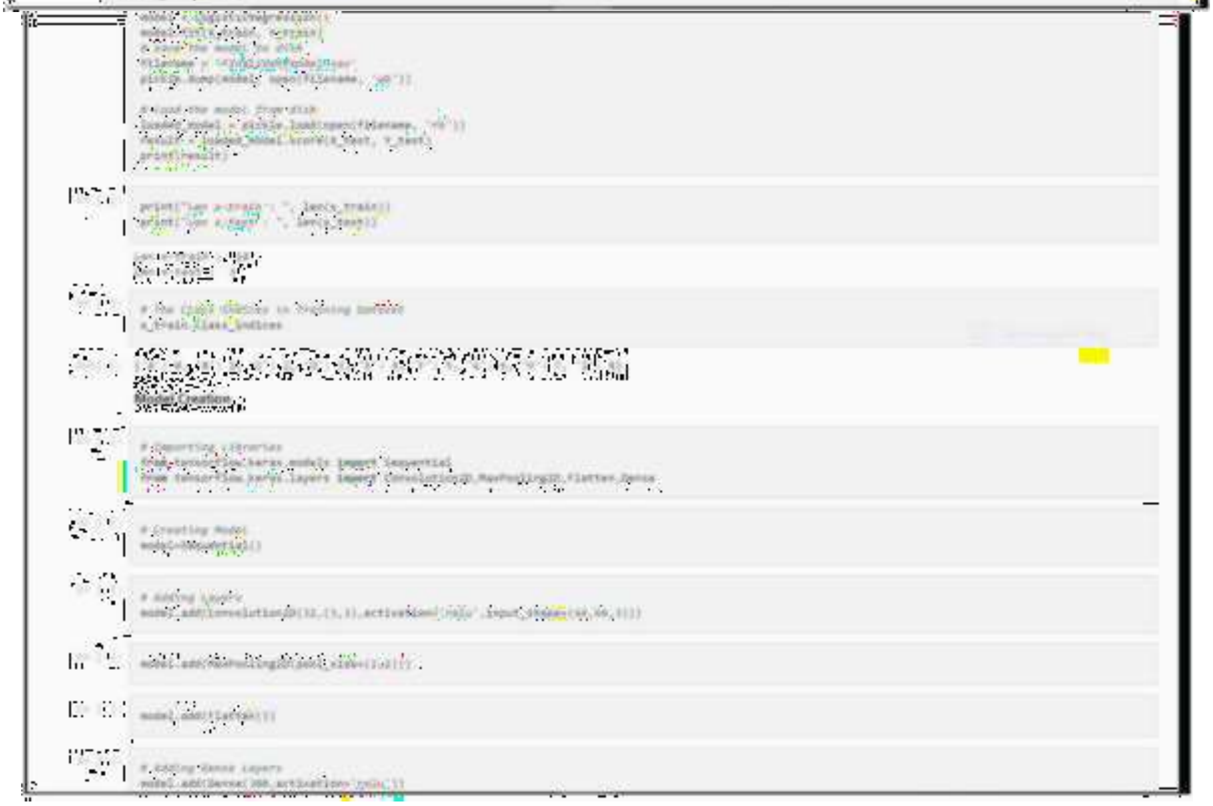
```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/Training set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/Test set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found [illegible] images belonging to [illegible] classes.
Found [illegible] images belonging to [illegible] classes.
```

```
# Save Model Using Pickle
import pandas
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
import pickle
```

```
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-
diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = pandas.read_csv(url, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
test_size = 0.33
seed = 7
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(X, Y, test_size=test_size,
random_state=seed)
```



```
model = LogisticRegression()
model.fit(X_train, Y_train)
# save the model to disk
filename = 'finalized_model.sav'
pickle.dump(model, open(filename, 'wb'))

# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.score(X_test, Y_test)
print(result)
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train : [illegible]
Len x-test : [illegible]
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
[illegible]
```

Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

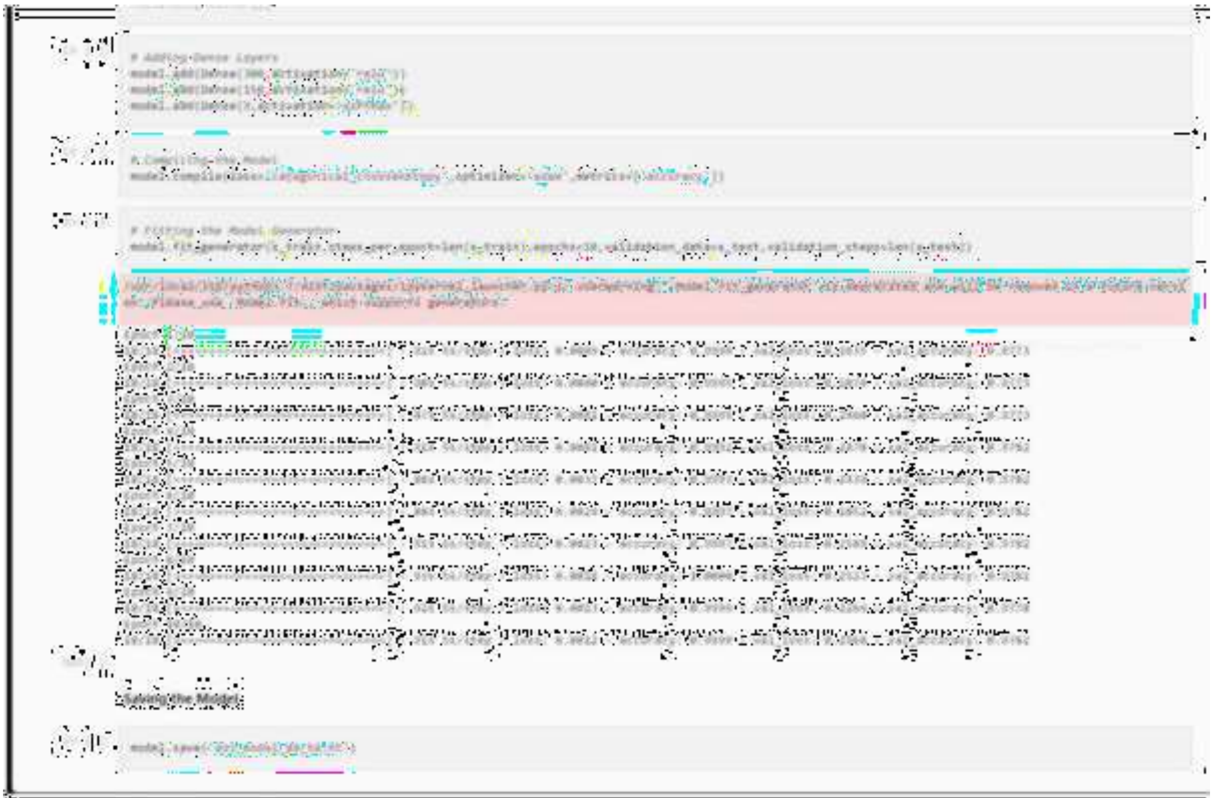
```
# Creating Model
model=Sequential()
```

```
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Flatten())
```

```
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
```



```
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(3,activation='softmax'))
```

```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

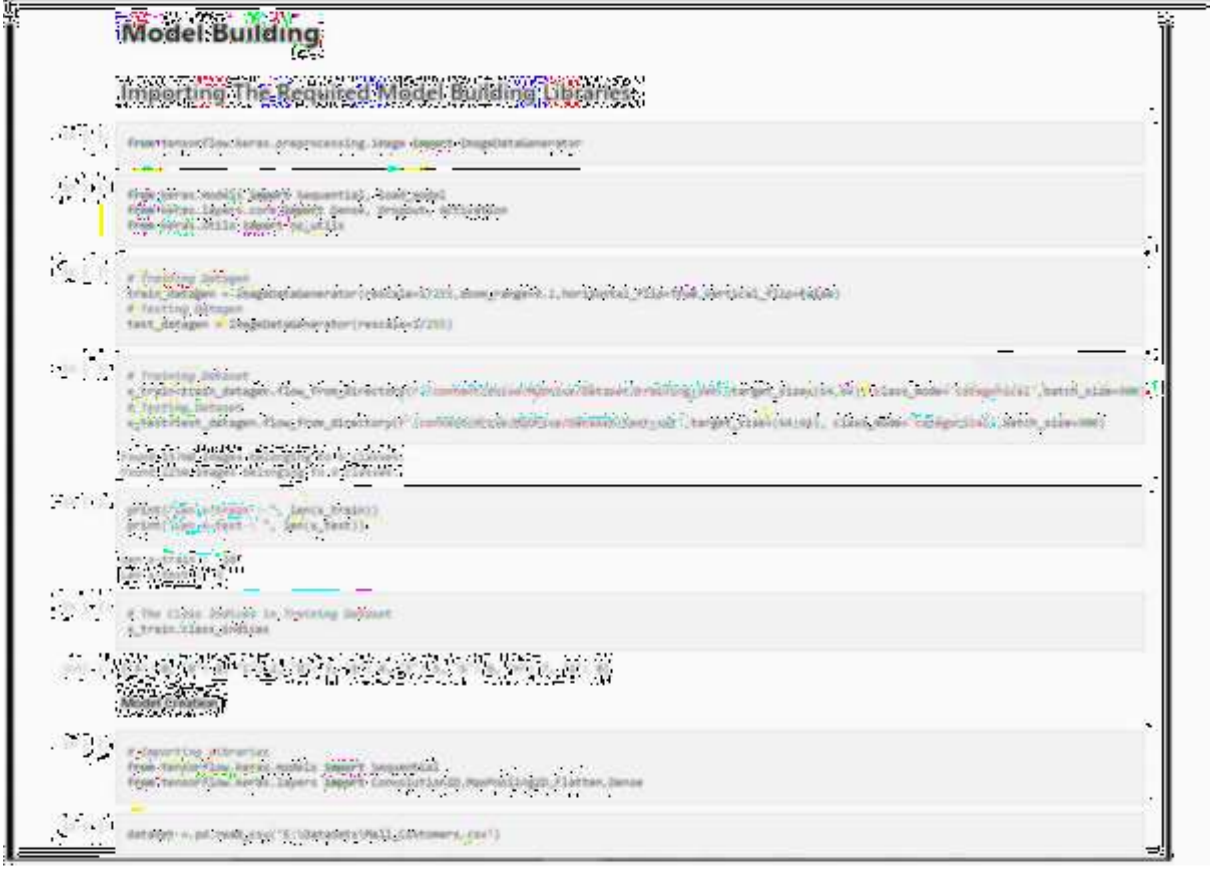
```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:[illegible]: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
```

```
Epoch 1/10
[illegible]
Epoch 2/10
[illegible]
Epoch 3/10
[illegible]
Epoch 4/10
[illegible]
Epoch 5/10
[illegible]
Epoch 6/10
[illegible]
Epoch 7/10
[illegible]
Epoch 8/10
[illegible]
Epoch 9/10
[illegible]
Epoch 10/10
[illegible]
```

Saving the Model

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



Model Building

Importing The Required Model Building Libraries

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
from keras.models import Sequential, load_model
from keras.layers.core import Dense, Dropout, Activation
from keras.utils import np_utils
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64),class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found [illegible] images belonging to 3 classes.
Found [illegible] images belonging to 3 classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  [illegible]
Len x-test :  [illegible]
```

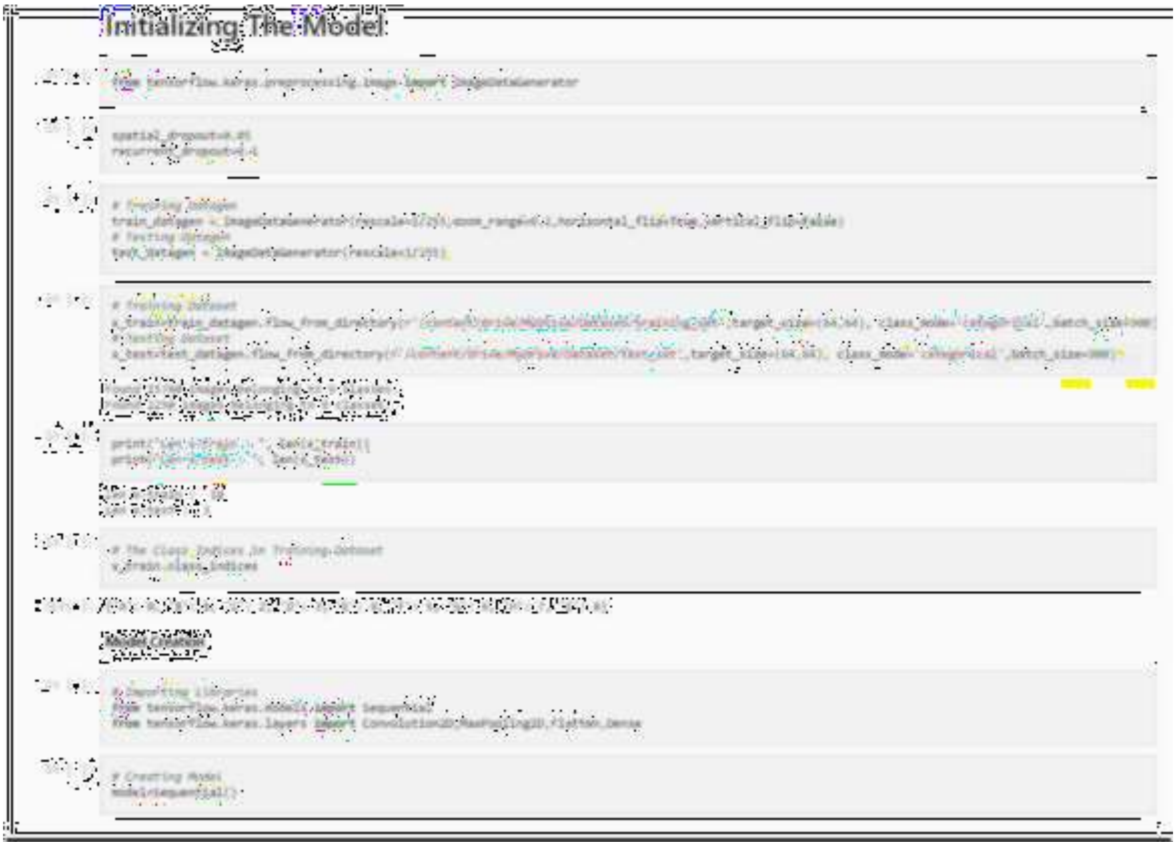
```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
[illegible]
```

Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
dataset = pd.read_csv('E:\Datasets\Mall_Customers.csv')
```



Initializing The Model

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
spatial_dropout=0.05
recurrent_dropout=0.1
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'[illegible]',target_size=(64,64), class_mode='categorical',batch_size=900
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'[illegible]',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found [illegible] images belonging to [illegible] classes.
Found [illegible] images belonging to [illegible] classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train : [illegible]
Len x-test : [illegible]
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
[illegible]
```

Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
# Creating Model
model=Sequential()
```

Initializing The Model

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
spatial_dropout=0.05
recurrent_dropout=0.1
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'[illegible]',target_size=(64,64), class_mode='categorical',batch_size=900
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'[illegible]',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found [illegible] images belonging to [illegible] classes.
Found [illegible] images belonging to [illegible] classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train : [illegible]
Len x-test : [illegible]
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
[illegible]
```

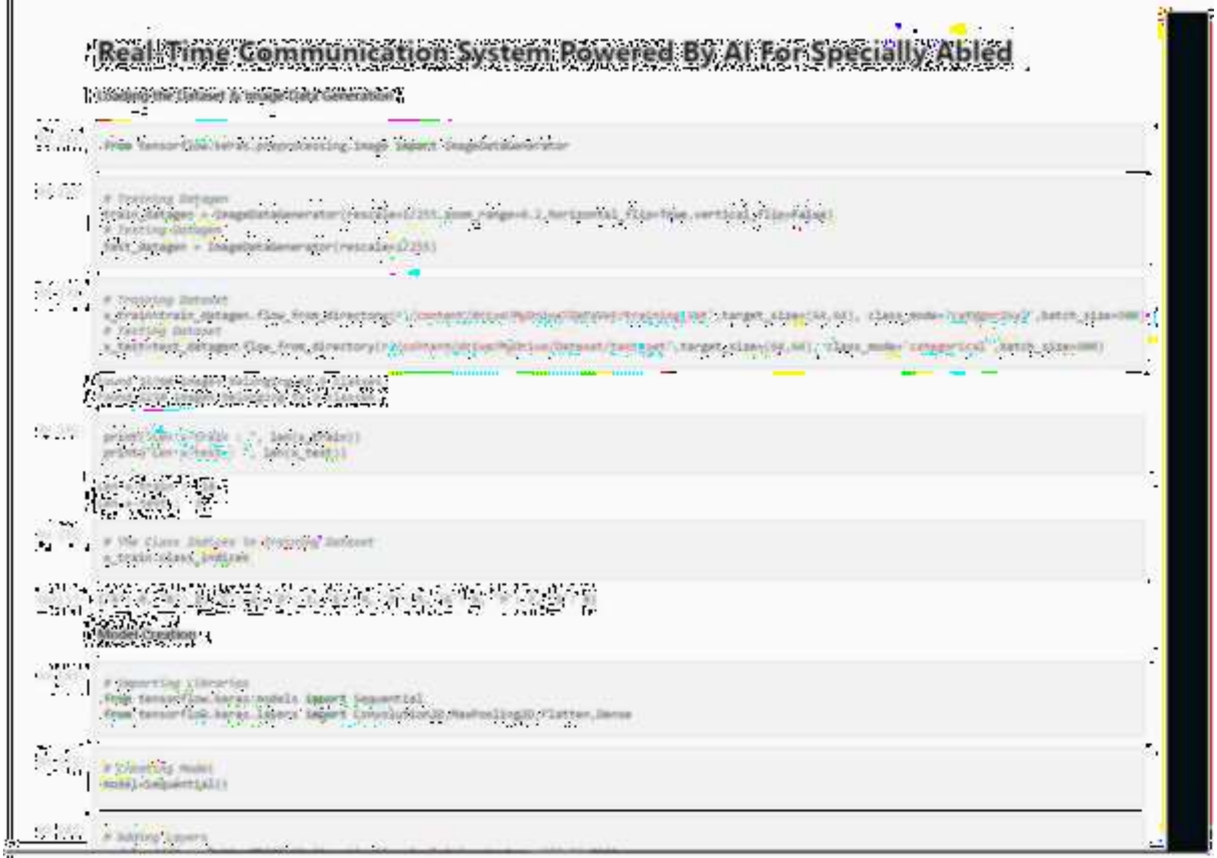
Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
# Creating Model
model=Sequential()
```

8 TESTING

8.1 Test cases



Real Time Communication System Powered By AI For Specially Abled

Loading the Dataset & Image Data Generation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found [illegible] images belonging to 9 classes.
Found [illegible] images belonging to 9 classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train : [illegible]
Len x-test : [illegible]
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

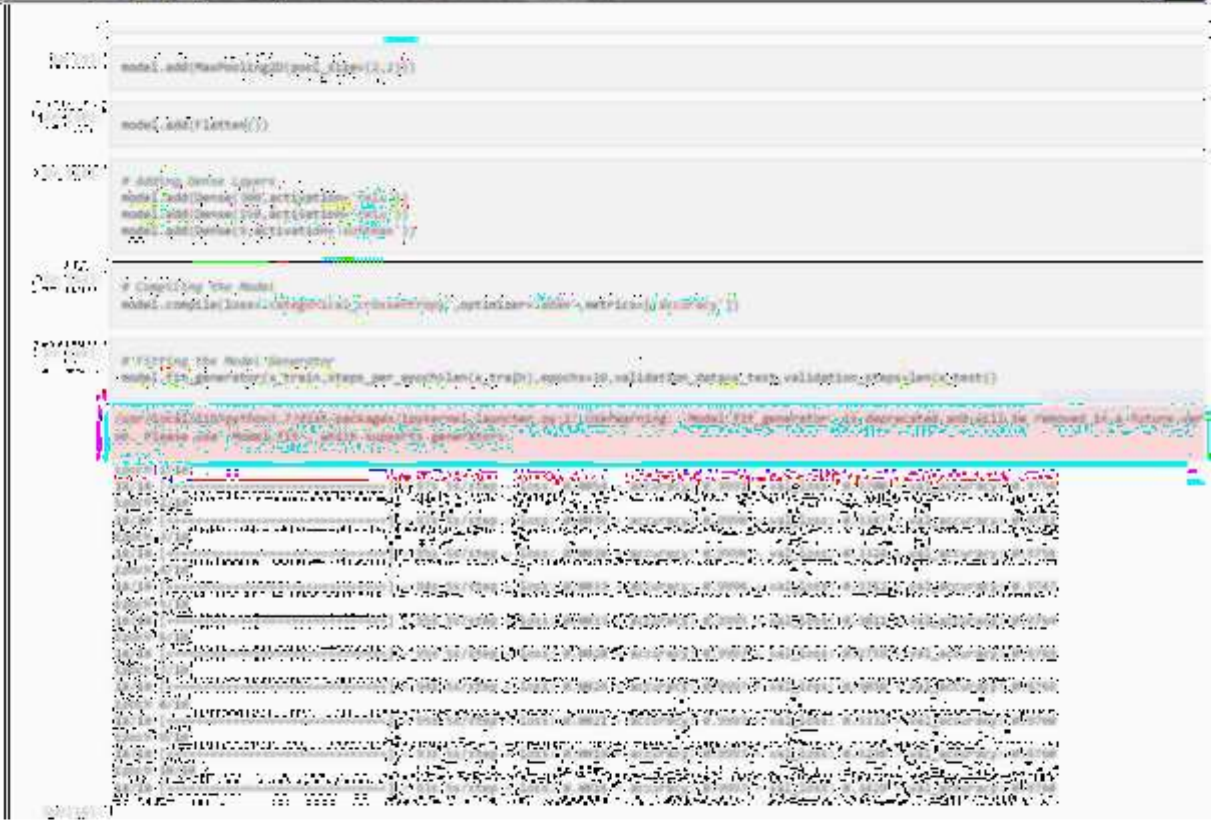
```
[illegible]
```

Model Creation

```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
# Creating Model
model=Sequential()
```

```
# Adding Layers
```



```
model.add(MaxPooling2D(pool_size=(2,2)))
```

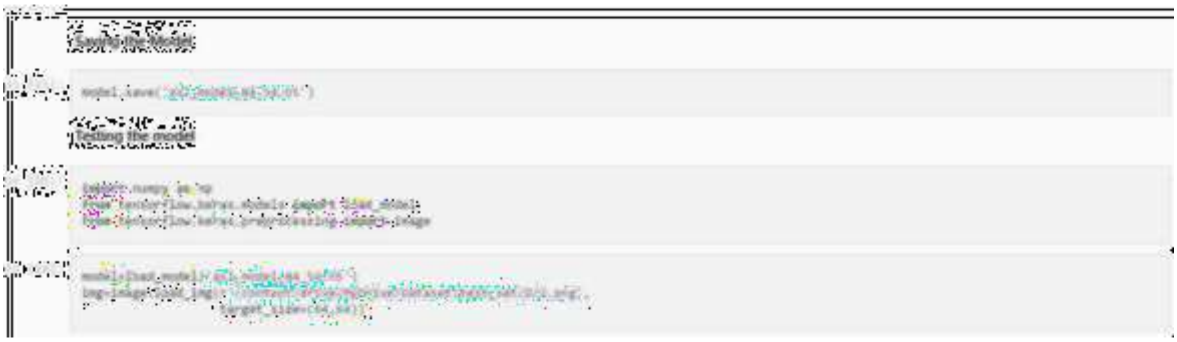
```
model.add(Flatten())
```

```
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
```

```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future ver
on. Please use `Model.fit`, which supports generators.
Epoch 1/10
[illegible]
```



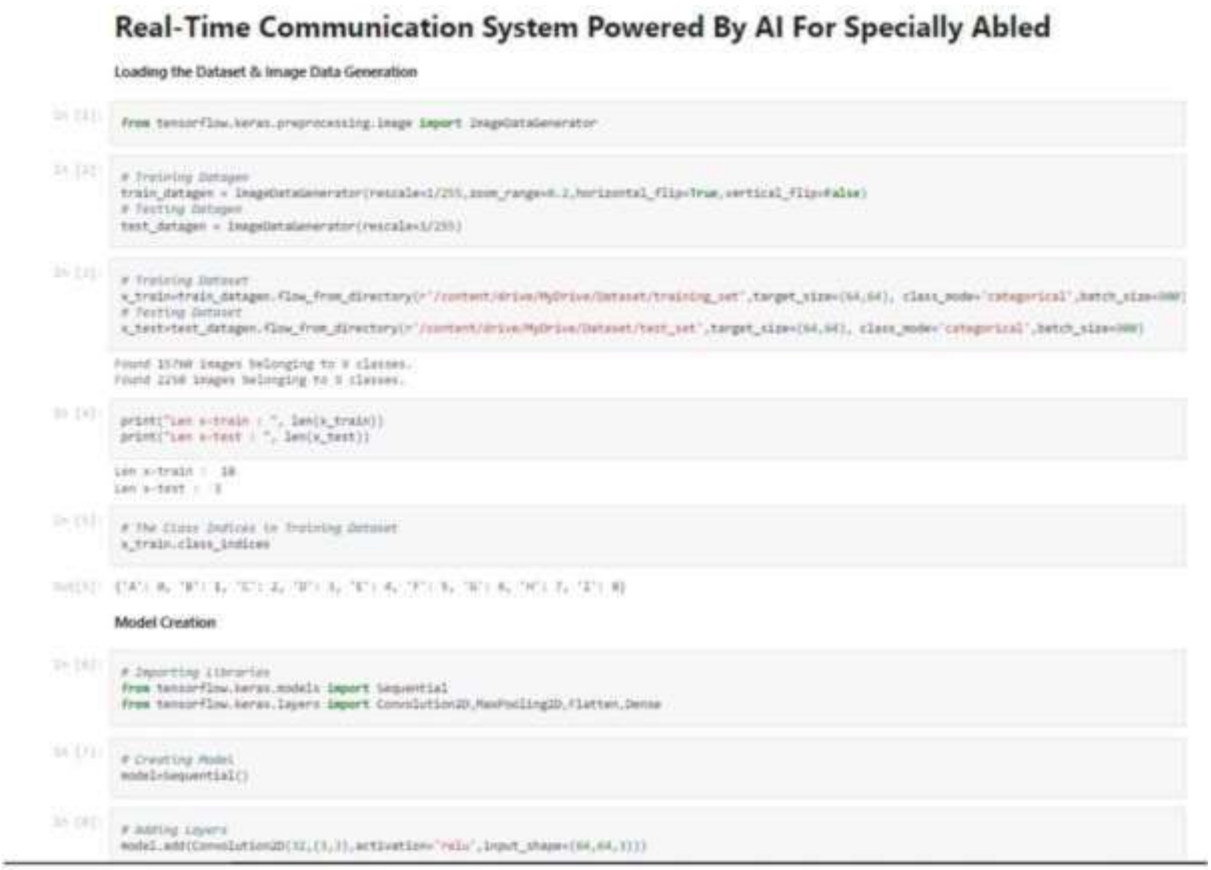
Saving the Model

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

Testing the model

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

```
model=load_model('[illegible].h5')
img=image.load_img('[illegible].png',
                   target_size=(64,64))
```



Real-Time Communication System Powered By AI For Specially Abled

Loading the Dataset & Image Data Generation

```
In [1]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
In [2]: # Training Datagen
        train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
        # Testing Datagen
        test_datagen = ImageDataGenerator(rescale=1/255)
```

```
In [3]: # Training Dataset
        x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
        # Testing Dataset
        x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)

        Found 15760 images belonging to 9 classes.
        Found 2250 images belonging to 9 classes.
```

```
In [4]: print("Len x-train : ", len(x_train))
        print("Len x-test : ", len(x_test))

        Len x-train :  18
        Len x-test :  3
```

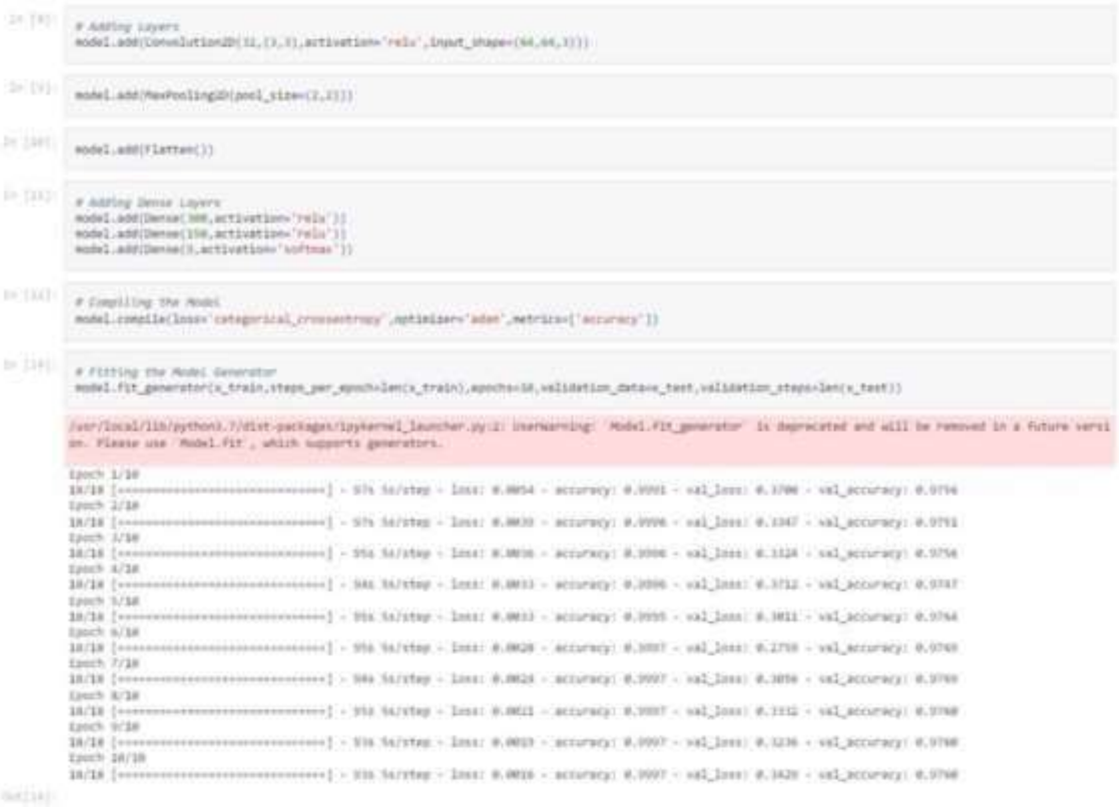
```
In [5]: # The Class Indices in Training Dataset
        x_train.class_indices

Out[5]: {'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

Model Creation

```
In [6]: # Importing Libraries
        from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
In [7]: # Creating Model
        model=Sequential()
```

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



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

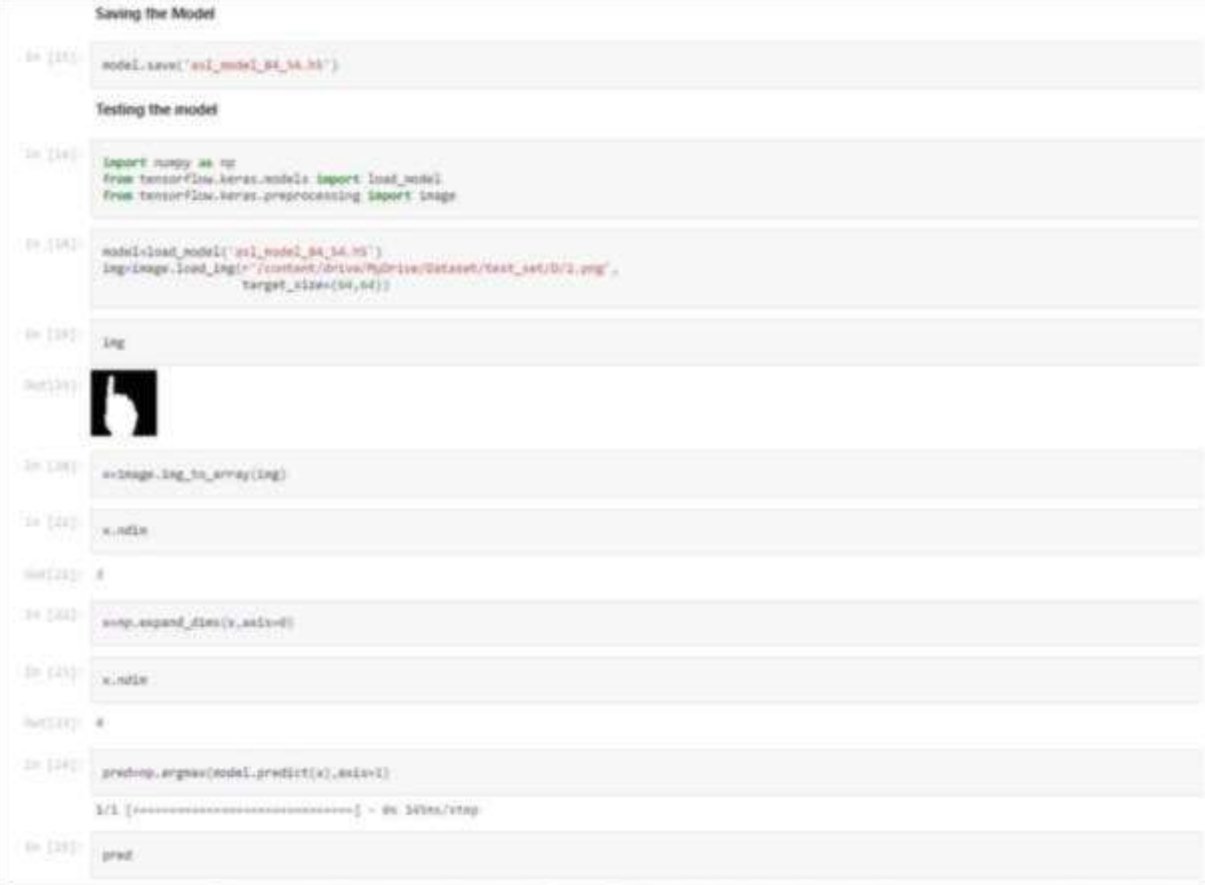
```
In [9]: model.add(MaxPooling2D(pool_size=(2,2)))
```

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

```
In [11]: # Adding Dense Layers
         model.add(Dense(300,activation='relu'))
         model.add(Dense(150,activation='relu'))
         model.add(Dense(9,activation='softmax'))
```

```
In [12]: # Compiling the Model
         model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
In [14]: # Fitting the Model Generator
         model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))

         /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versi
         on. Please use `Model.fit`, which supports generators.

         Epoch 1/10
         18/18 [==============================] - 57s 3s/step - loss: 0.0054 - accuracy: 0.9991 - val_loss: 0.3700 - val_accuracy: 0.9756
         Epoch 2/10
         18/18 [==============================] - 57s 3s/step - loss: 0.0039 - accuracy: 0.9996 - val_loss: 0.1347 - val_accuracy: 0.9751
         Epoch 3/10
         18/18 [==============================] - 55s 3s/step - loss: 0.0036 - accuracy: 0.9996 - val_loss: 0.1124 - val_accuracy: 0.9756
         Epoch 4/10
         18/18 [==============================] - 54s 3s/step - loss: 0.0033 - accuracy: 0.9996 - val_loss: 0.3712 - val_accuracy: 0.9747
         Epoch 5/10
         18/18 [==============================] - 55s 3s/step - loss: 0.0033 - accuracy: 0.9995 - val_loss: 0.3011 - val_accuracy: 0.9764
         Epoch 6/10
         18/18 [==============================] - 55s 3s/step - loss: 0.0028 - accuracy: 0.9997 - val_loss: 0.2759 - val_accuracy: 0.9769
         Epoch 7/10
         18/18 [==============================] - 54s 3s/step - loss: 0.0024 - accuracy: 0.9997 - val_loss: 0.3056 - val_accuracy: 0.9769
         Epoch 8/10
         18/18 [==============================] - 55s 3s/step - loss: 0.0021 - accuracy: 0.9997 - val_loss: 0.1112 - val_accuracy: 0.9760
         Epoch 9/10
         18/18 [==============================] - 53s 3s/step - loss: 0.0019 - accuracy: 0.9997 - val_loss: 0.1236 - val_accuracy: 0.9760
         Epoch 10/10
         18/18 [==============================] - 53s 3s/step - loss: 0.0016 - accuracy: 0.9997 - val_loss: 0.1429 - val_accuracy: 0.9760

Out[14]:
```

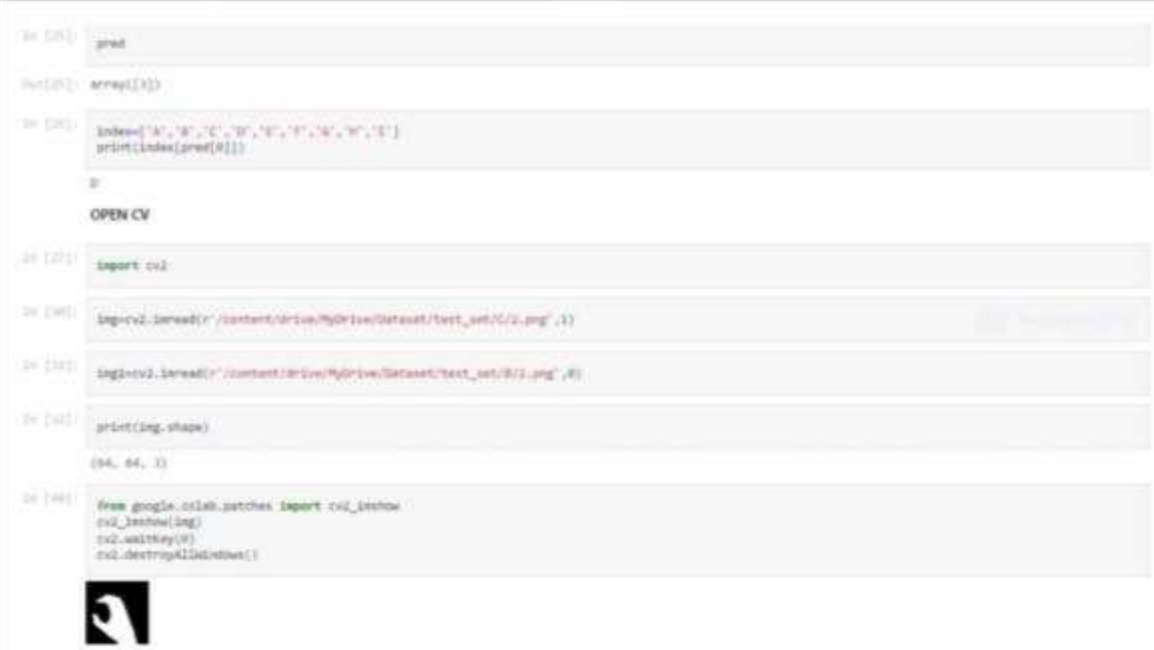


Saving the Model

```
In [15]: model.save('asl_model_84_54.h5')
```

Testing the model

```
In [16]: import numpy as np
         from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image

In [18]: model=load_model('asl_model_84_54.h5')
         img=image.load_img(r"/content/drive/MyDrive/Dataset/test_set/D/1.png",
                            target_size=(64,64))

In [19]: img

Out[19]:

In [20]: x=image.img_to_array(img)

In [21]: x.ndim

Out[21]: 3

In [22]: x=np.expand_dims(x,axis=0)

In [23]: x.ndim

Out[23]: 4

In [24]: pred=np.argmax(model.predict(x),axis=1)

         1/1 [==============================] - 0s 145ms/step

In [25]: pred
```



```
In [25]: pred

Out[25]: array([3])

In [26]: index=['A','B','C','D','E','F','G','H','I']
         print(index[pred[0]])

         D
```

OPEN CV

```
In [27]: import cv2

In [30]: img=cv2.imread(r'/content/drive/MyDrive/Dataset/test_set/C/1.png',1)

In [31]: img1=cv2.imread(r'/content/drive/MyDrive/Dataset/test_set/D/1.png',0)

In [41]: print(img.shape)

         (64, 64, 3)

In [40]: from google.colab.patches import cv2_imshow
         cv2_imshow(img)
         cv2.waitKey(0)
         cv2.destroyAllWindows()
```

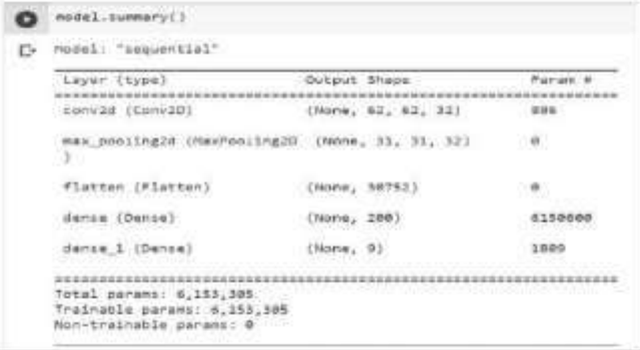
9 RESULTS

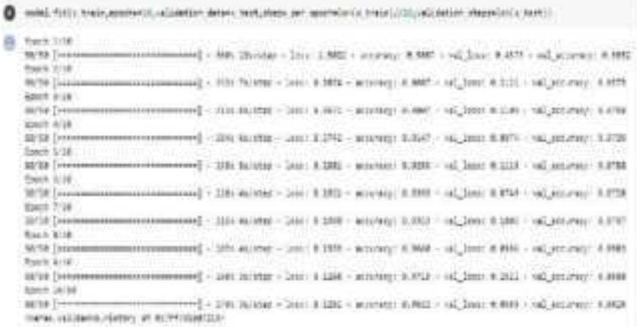
9.1 Performance metrics

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot

1.	Model Summary	Model - Sequential model
		Layers:
		Conv2D-(None,62,62,32)
		MaxPooling2D-(None,31,31,32)
		Flatten-(None,30752)
		Dense-(None,200)
		Dense_1 -(None,9)
			 model.summary()
			Model: "sequential"
			Layer (type) Output Shape Param #
			conv2d (Conv2D) (None, 62, 62, 32) 896
			max_pooling2d (MaxPooling2D (None, 31, 31, 32) 0
			)
			flatten (Flatten) (None, 30752) 0
			dense (Dense) (None, 200) 6150600
			dense_1 (Dense) (None, 9) 1809
			Total params: 6,153,305
			Trainable params: 6,153,305
			Non-trainable params: 0

2.	Accuracy	Training Accuracy - 0.9622 Validation Accuracy -0.9826	 model.fit([illegible] Epoch 1/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 2/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 3/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 4/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 5/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 6/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 7/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 8/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 9/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] Epoch 10/10 59/59 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy: [illegible] <keras.callbacks.History at [illegible]>
3	Confidence Score	Class Detected – N/A Confidence Score -N/A	N/A

10 ADVANTAGES AND DISADVANTAGES

ADVANTAGES

It enables employees from across the world to communicate with each other 24x7 and share ideas or solve problems quickly. It is a cost-effective way of getting several people from different locations to attend meetings and conferences – without having to spend time or money on travel, and accommodation.

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. This could result in damaging your relationship with them.

11 CONCLUSION

Real-time communication (RTC) workloads can be deployed on AWS to attain scalability, elasticity, and high availability while meeting the key requirements. Today, several customers are using AWS, its partners, and open source solutions to run RTC workloads with reduced cost and faster agility as well 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 AWS and optimize the solutions to meet end user requirements while optimizing for the cloud.

12 FUTURE SCOPE

1. Through image recognition technology, AI understands the context of objects in photos and describes photos to people.

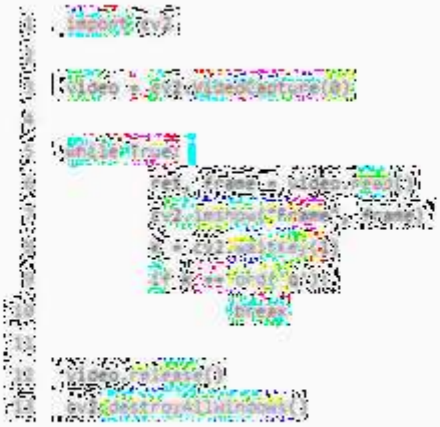
2. The speech-to-text and text-to-speech technologies helped those people who had speech impediments

3. 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 and emotions.

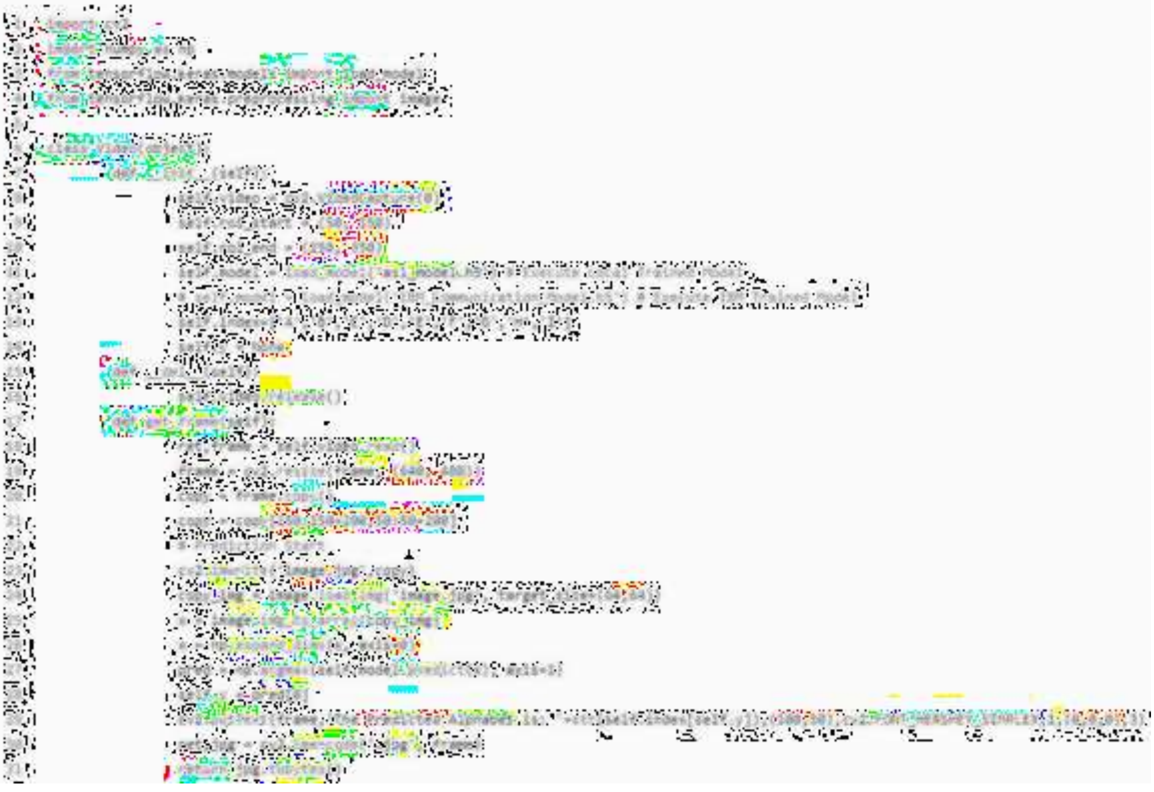
4. Autonomous vehicles are in trend and their success is due to AI technology. These vehicles can be beneficial to people living with limited physical mobility.

13. APPENDIX

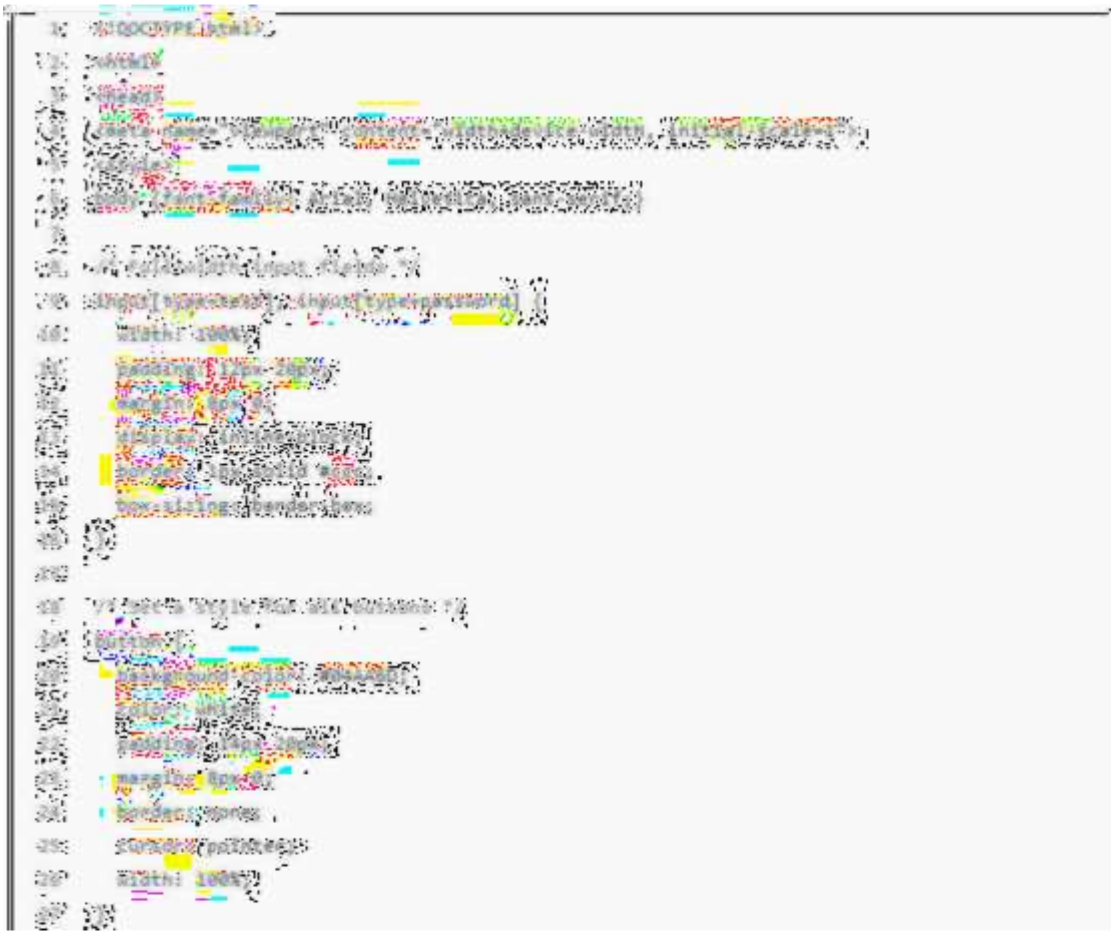
Source code



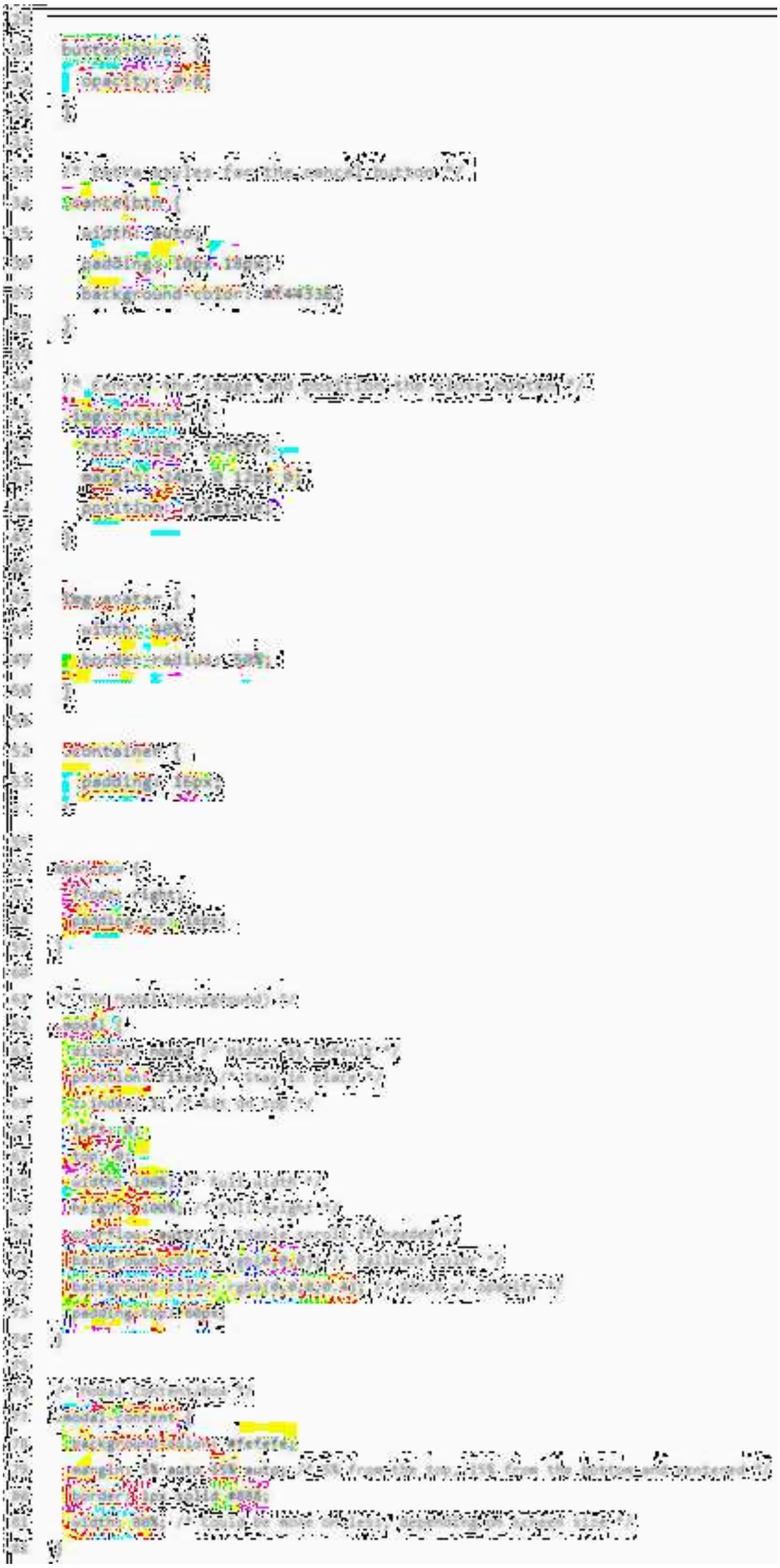
```
import cv2

video = cv2.VideoCapture(0)

while True:
        ret, frame = video.read()
        cv2.imshow("Frame", frame)
        k = cv2.waitKey(1)
        if k == ord('q'):
                break

video.release()
cv2.destroyAllWindows()
```



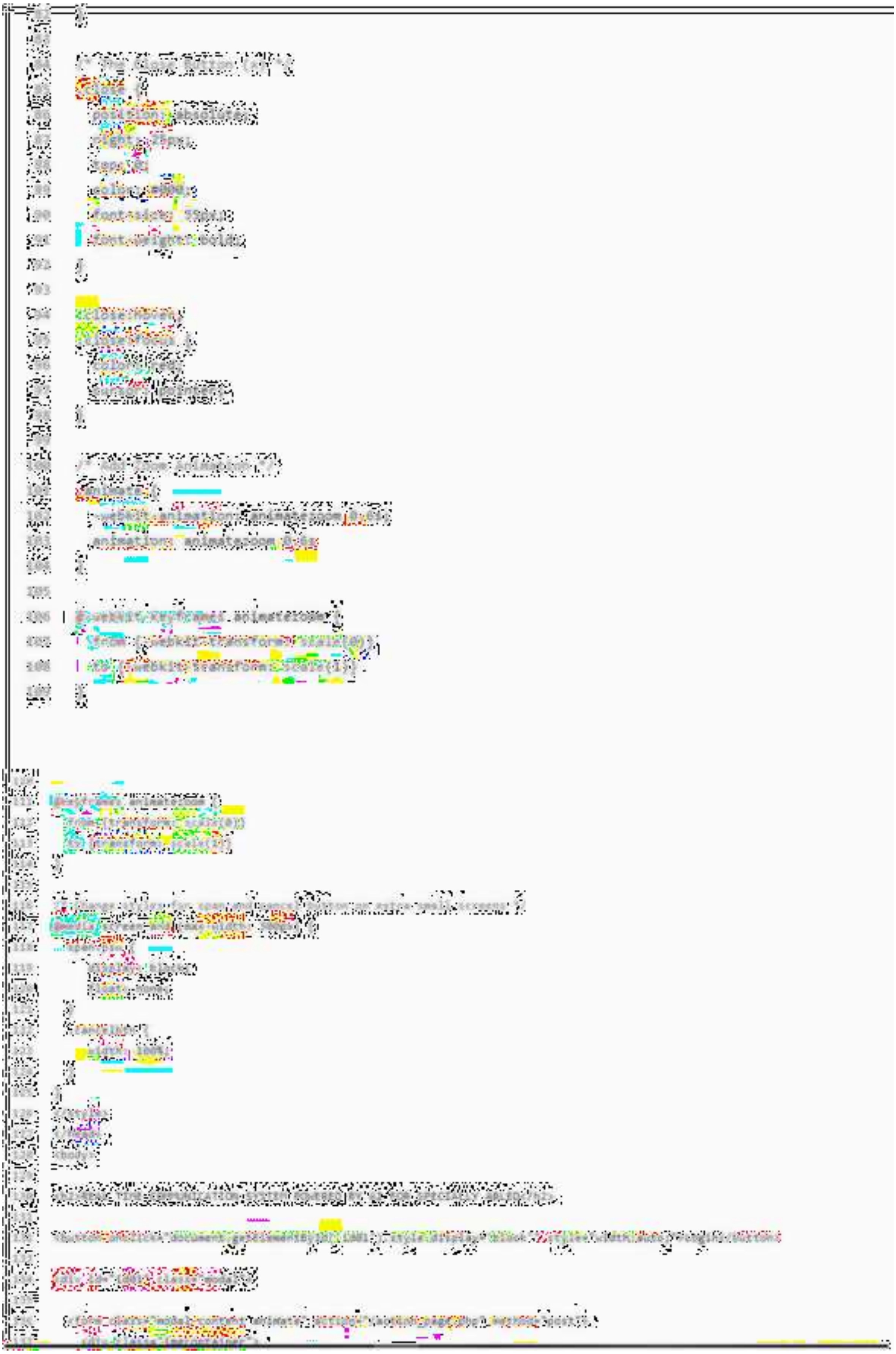
```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class VideoCamera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        self.model = load_model('asl_model.h5') # Execute Local Trained Model
        # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```



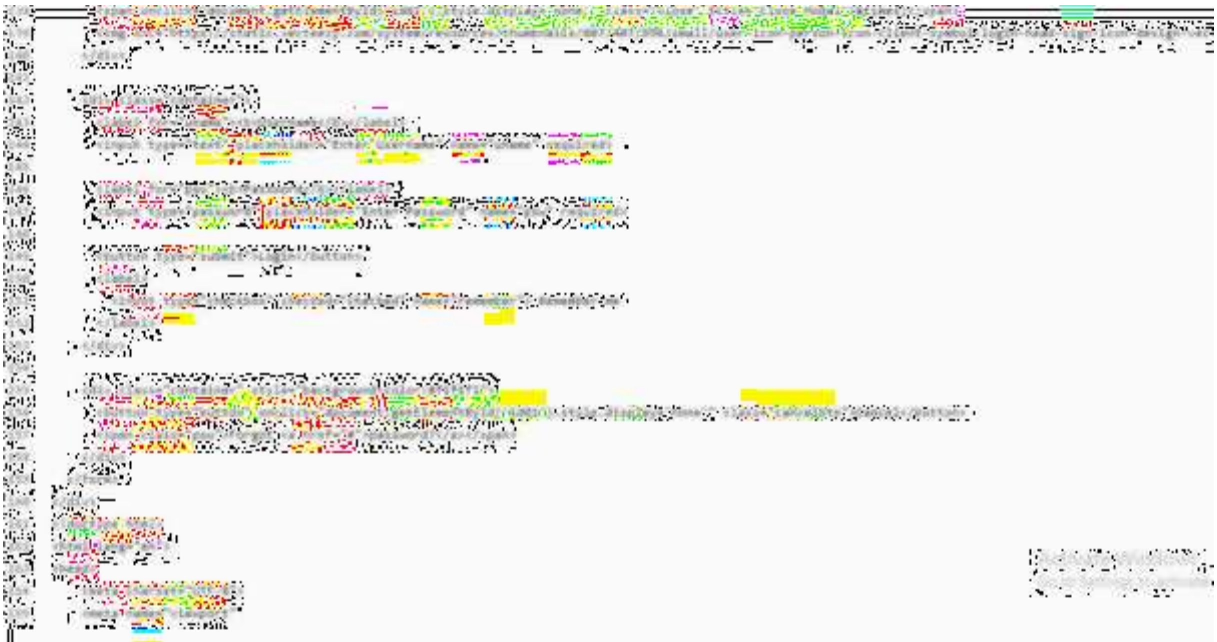
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}
```



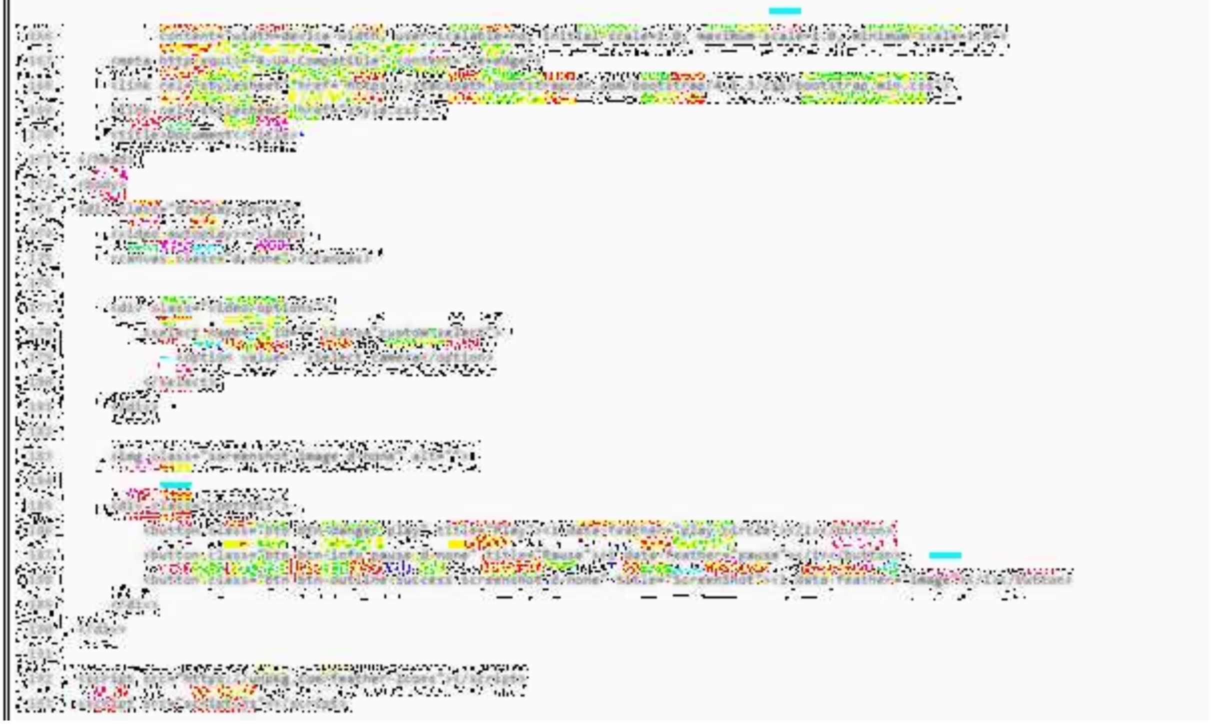
```
  button:hover {
    opacity: 0.8;
  }

  /* Extra styles for the cancel button */
  .cancelbtn {
    width: auto;
    padding: 10px 18px;
    background-color: #f44336;
  }

  /* Center the image and position the close button */
  .imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
    position: relative;
  }

  img.avatar {
    width: 40%;
    border-radius: 50%;
  }

  .container {
    padding: 16px;
  }

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}
```



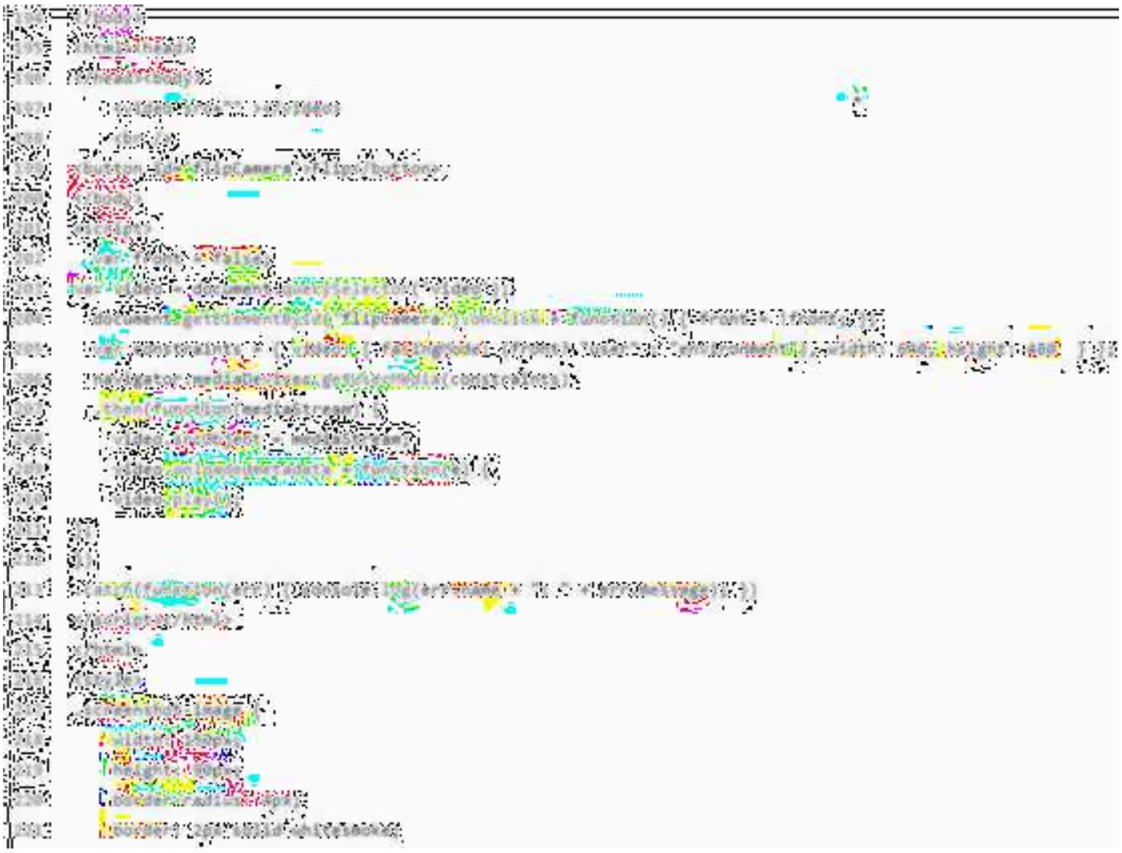
```
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}


@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}
</style>
</head>
<body>

<h2>REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED</h2>

<button onclick="document.getElementById('id01').style.display='block'" style="width:auto;">Login</button>

<div id="id01" class="modal">

  <form class="modal-content animate" action="/action_page.php" method="post">
    <div class="imgcontainer">
```



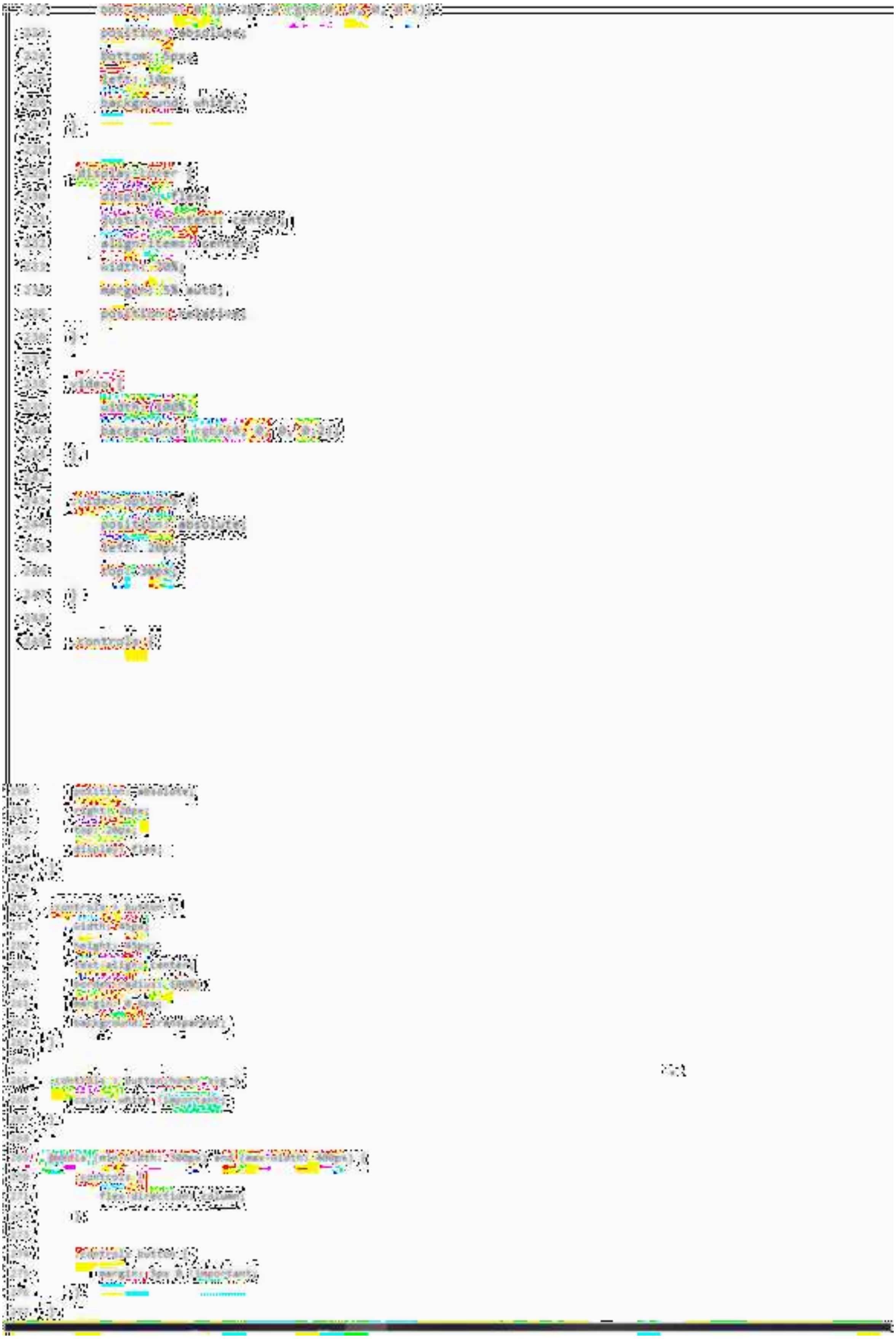
```
<div class="container">
  <label for="uname"><b>Username</b></label>
  <input type="text" placeholder="Enter Username" name="uname" required>

  <label for="psw"><b>Password</b></label>
  <input type="password" placeholder="Enter Password" name="psw" required>

  <button type="submit">Login</button>
  <label>
    [illegible]
  </label>
</div>
```



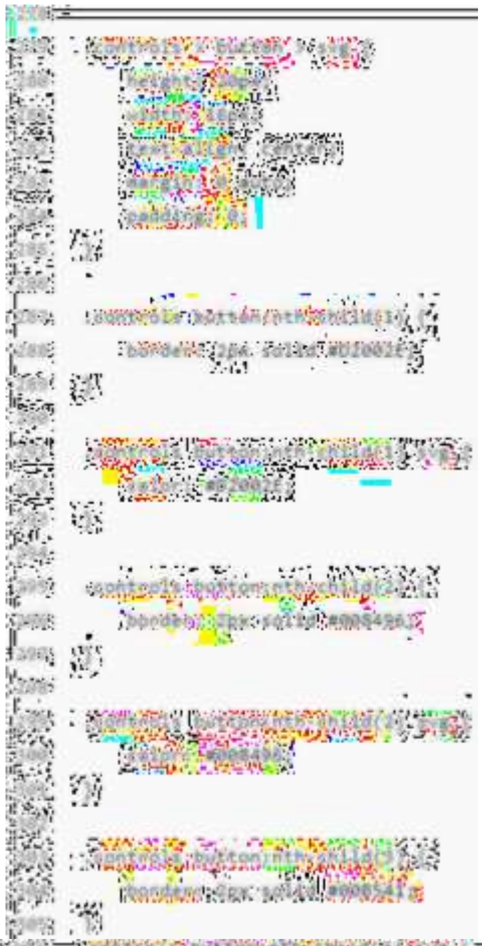
```
        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="[illegible]/bootstrap.min.css">
    <link rel="stylesheet" href="style.css">
    <title>Document</title>
</head>
<body>
<div class="display-cover">
    <video autoplay></video>
    <canvas class="d-none"></canvas>

    <div class="video-options">
        <select name="" id="" class="custom-select">
            <option value="">Select camera</option>
        </select>
    </div>

    <img class="screenshot-image d-none" alt="">

    <div class="controls">
        <button class="btn btn-danger play" title="Play"><i data-feather="play-circle"></i></button>
        <button class="btn btn-info pause d-none" title="Pause"><i data-feather="pause"></i></button>
        <button class="btn btn-outline-success screenshot d-none" title="ScreenShot"><i data-feather="image"></i></button>
    </div>
</div>
```



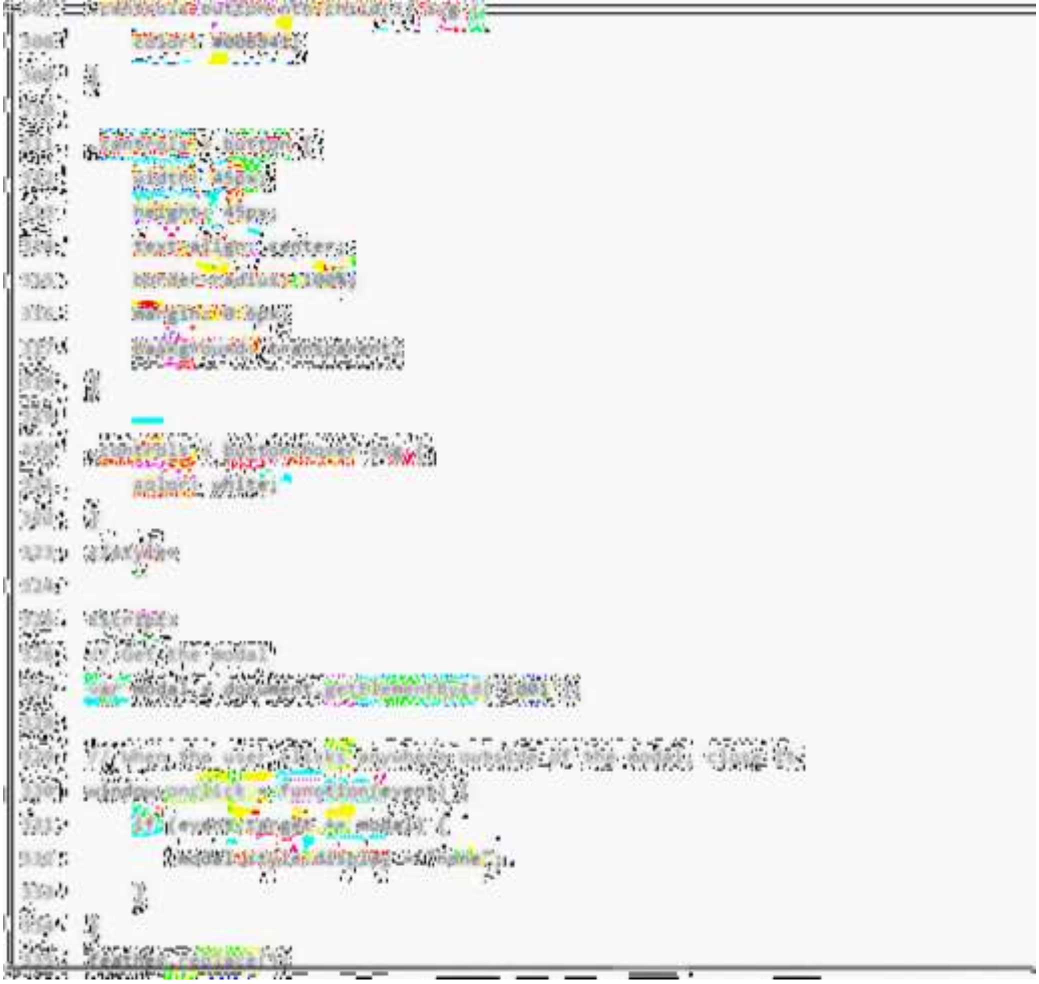
```
</body>
<html><head>
</head><body>
  <video src=""></video>
  <br/>
<button id="flipCamera">Flip</button>
</body>
<script>
  var front = false;
var video = document.querySelector('video');
  document.getElementById('flipCamera').onclick = function() { front = !front; };
  var constraints = { video: { facingMode: (front? "user" : "environment"), width: 640, height: 480 } };
  navigator.mediaDevices.getUserMedia(constraints)
  .then(function(mediaStream) {
    video.srcObject = mediaStream;
    video.onloadedmetadata = function(e) {
    video.play();
};
});
.catch(function(err) { console.log(err.name + ": " + err.message); })
</script></html>
</html>
<style>
.screenshot-image {
    width: 150px;
    height: 90px;
    border-radius: 4px;
    border: 2px solid whitesmoke;
```



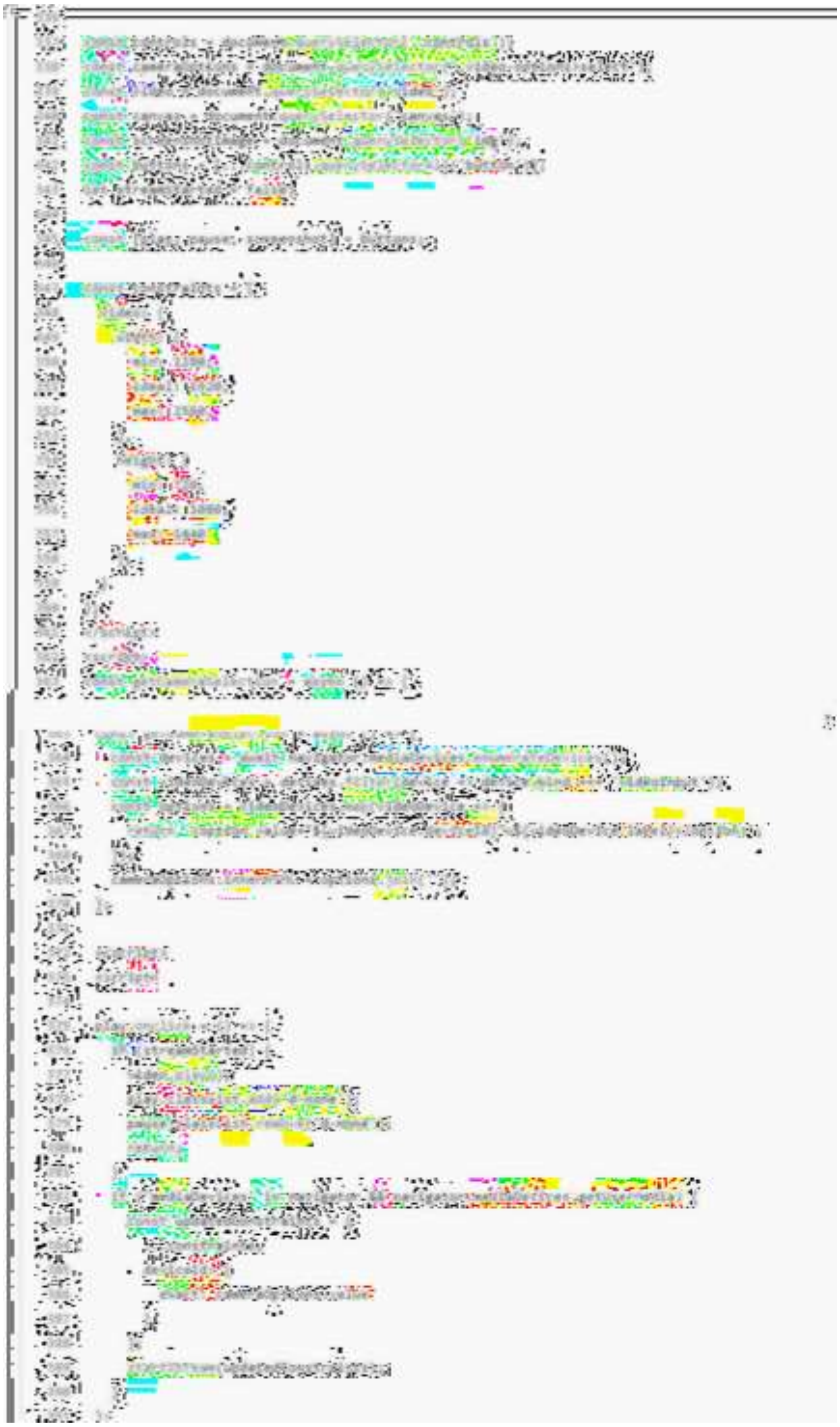
```
    box-shadow: 0 1px 2px rgba(0,0,0,0.2);
    position: absolute;
    bottom: 5px;
    left: 10px;
    background: white;
}

.display-cover {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 50%;
    margin: 5% auto;
    position: relative;
}

video {
    width: 100%;
    background: rgba(0,0,0,0.2);
}

.video-options {
    position: absolute;
    left: 10px;
    top: 30px;
}

.controls {
  position: absolute;
  right: 10px;
  top: 20px;
  display: flex;
}

.controls > button {
  width: 45px;
  height: 45px;
  text-align: center;
  border-radius: 100%;
  margin: 0 6px;
  background: transparent;
}

.controls > button:hover svg {
  color: white !important;
}

@media (min-width: 300px) and (max-width: 400px) {
    .controls {
        flex-direction: column;
    }

    .controls button {
        margin: 5px 0 !important;
    }
}
```

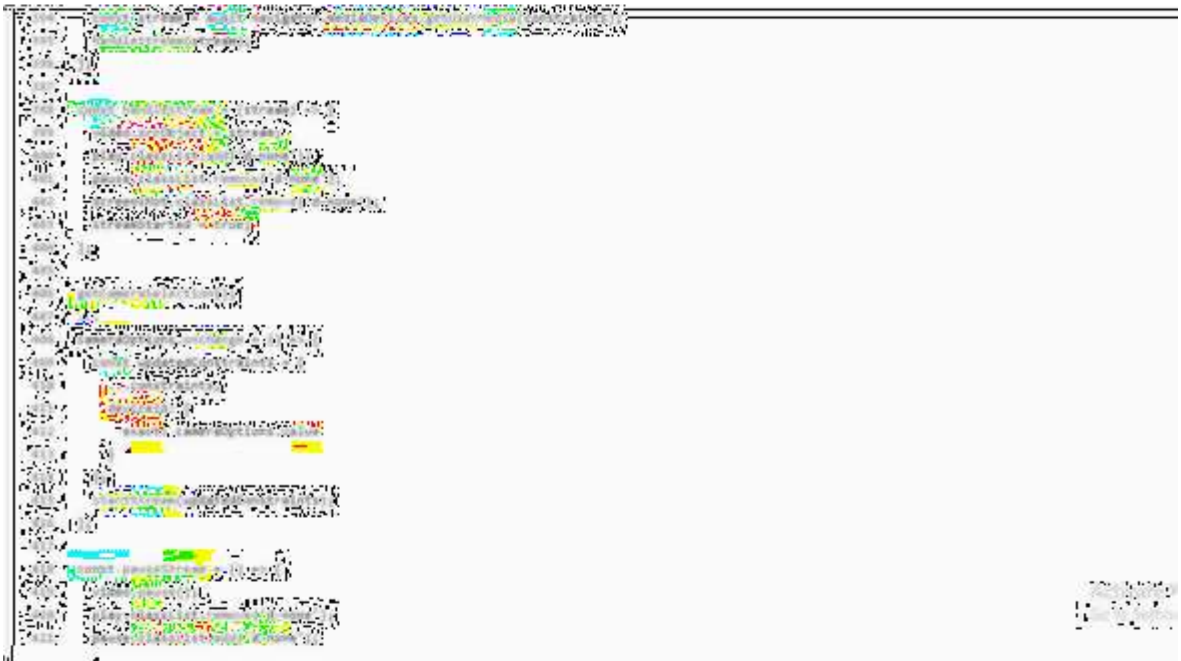


```
.controls > button > svg {
    height: 30px;
    width: 30px;
    text-align: center;
    margin: 0 auto;
    padding: 0;
}

.controls button:nth-child(1) {
    border: 2px solid #D2002E;
}

.controls button:nth-child(1) svg {
    color: #D2002E;
}

.controls button:nth-child(2) {
    border: 2px solid #008496;
}

.controls button:nth-child(2) svg {
    color: #008496;
}

.controls button:nth-child(3) {
    border: 2px solid #008541;
}
```

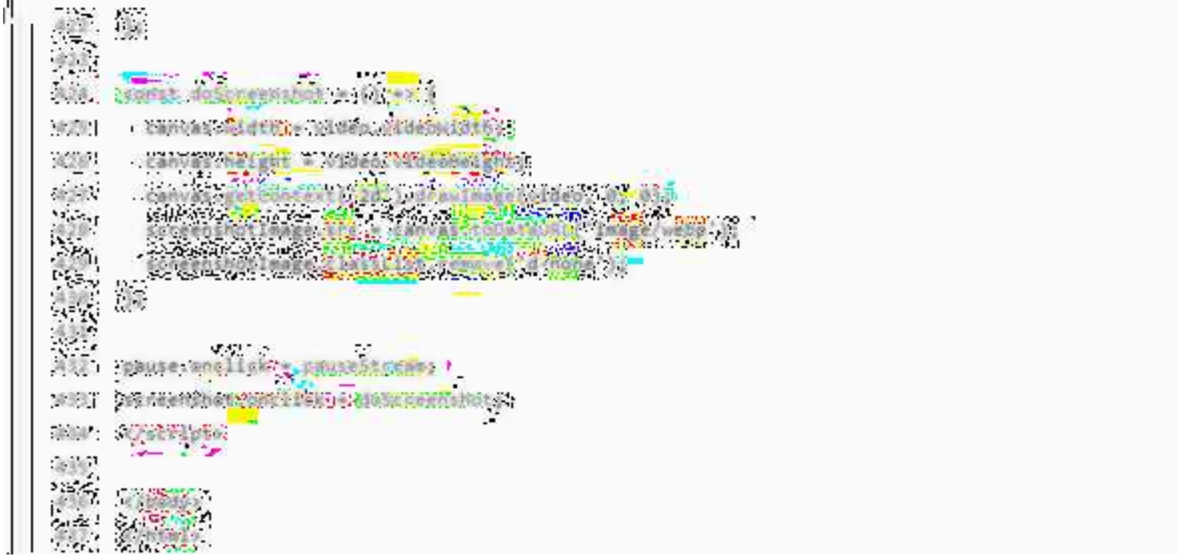


```
.controls button:nth-child(3) svg {
    color: #008541;
}

.controls > button {
    width: 45px;
    height: 45px;
    text-align: center;
    border-radius: 100%;
    margin: 0 10px;
    background: transparent;
}

.controls > button:hover svg {
    color: white;
}
</style>

<script>
// Get the modal
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
[illegible]
[illegible]
```

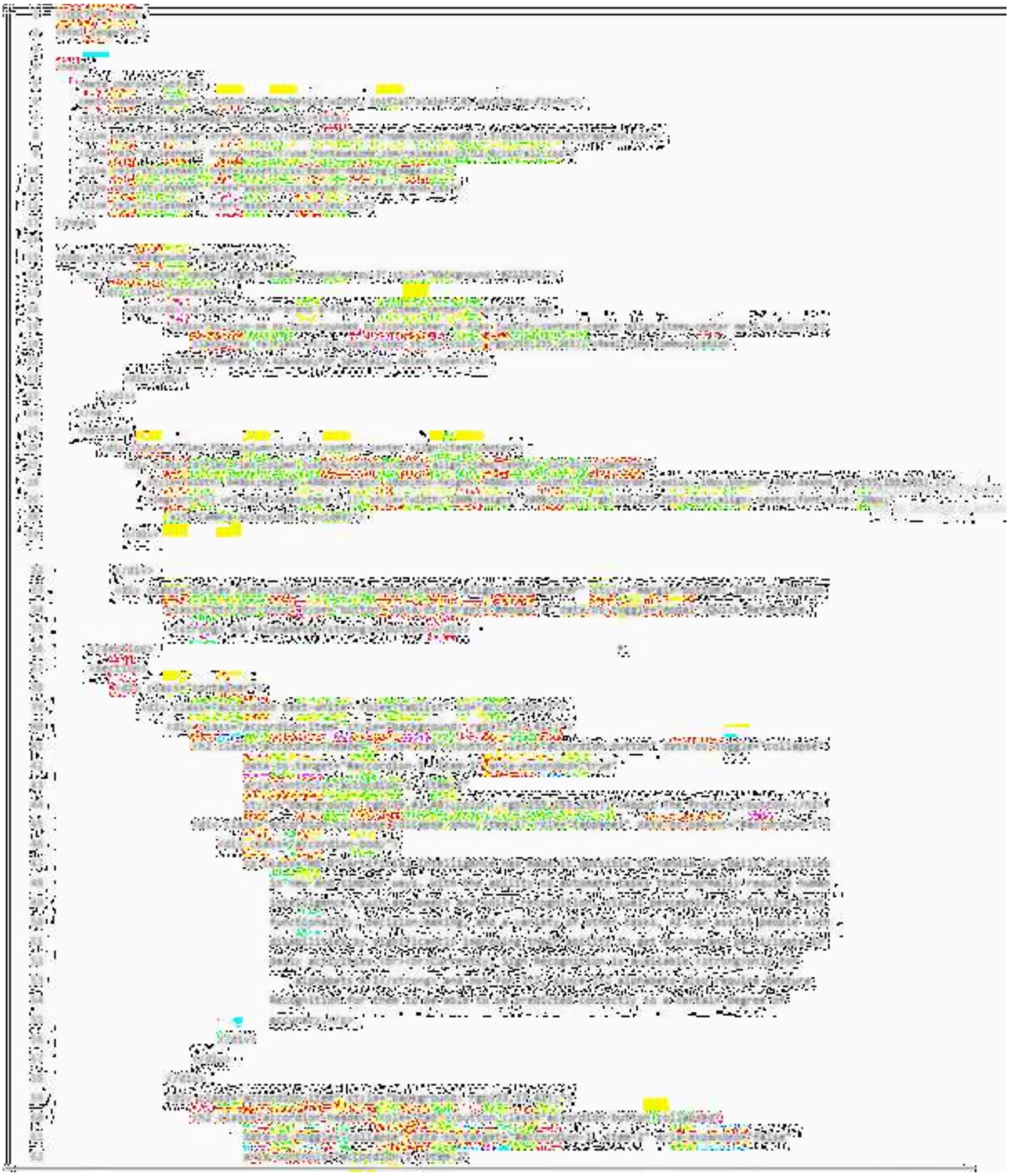


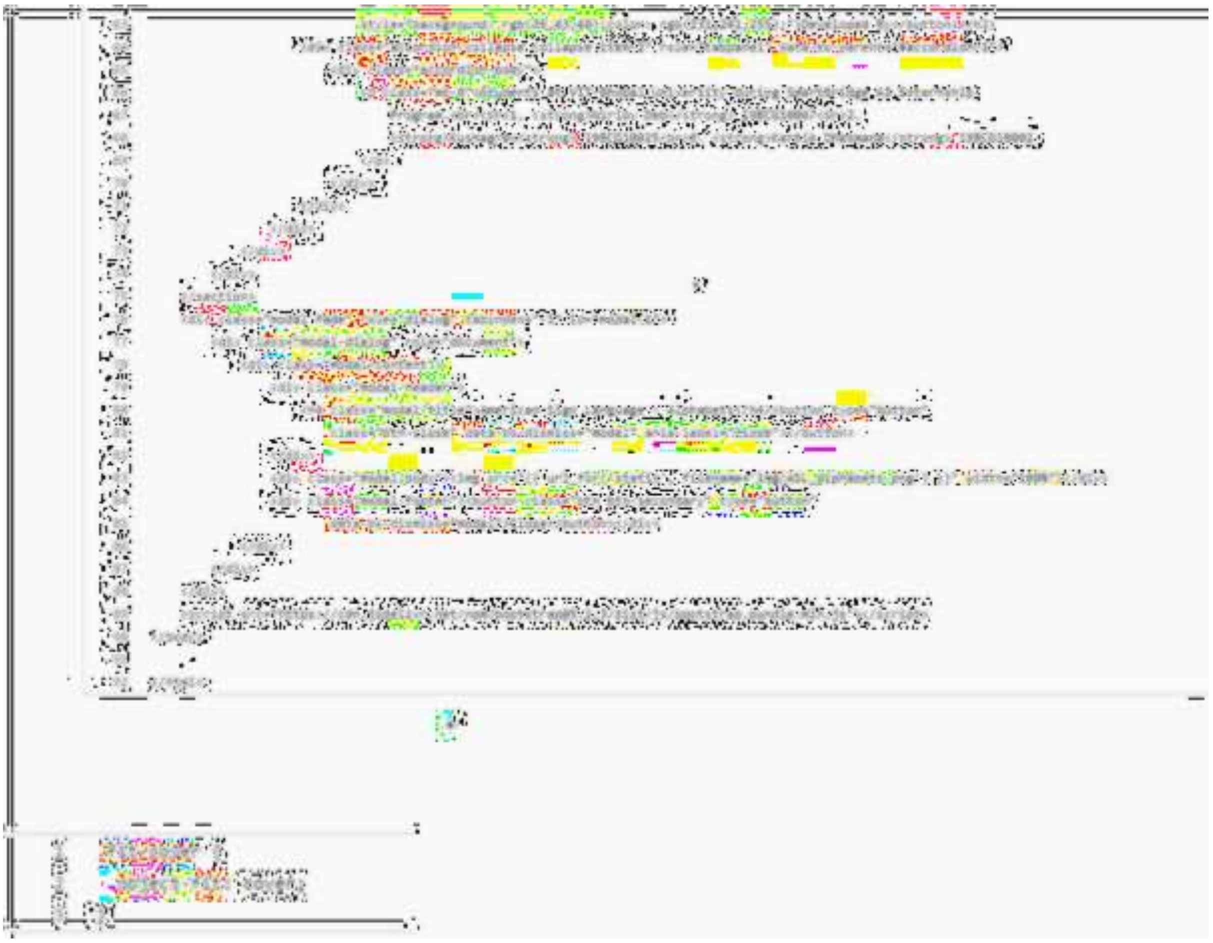


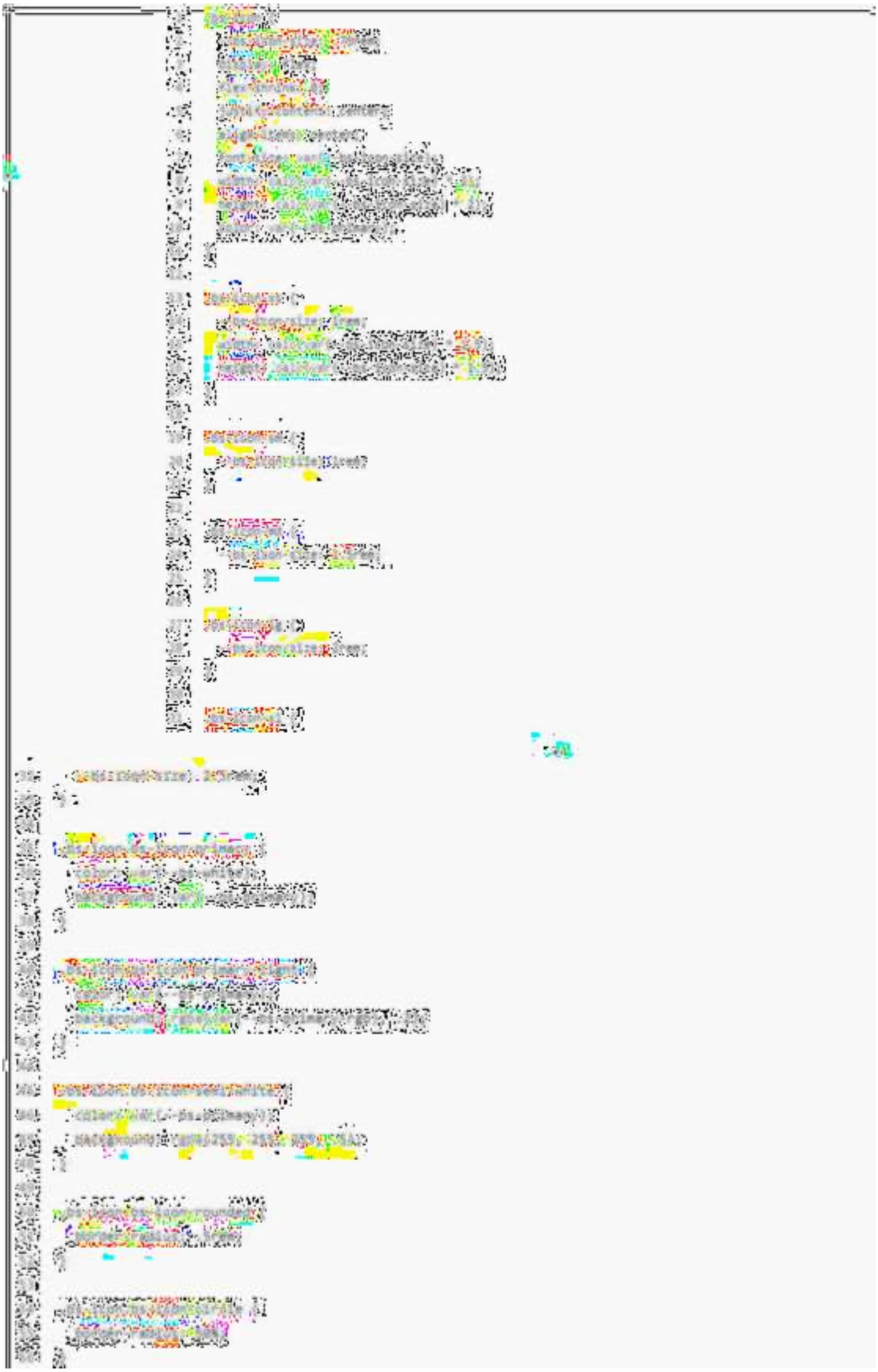
```
    const stream = await navigator.mediaDevices.getUserMedia(constraints);
    handleStream(stream);
};

const handleStream = (stream) => {
    video.srcObject = stream;
    play.classList.add('d-none');
    pause.classList.remove('d-none');
    screenshot.classList.remove('d-none');
    streamStarted = true;
};

getCameraSelection();

cameraOptions.onchange = () => {
    const updatedConstraints = {
        ...constraints,
        deviceId: {
            exact: cameraOptions.value
        }
    };
    startStream(updatedConstraints);
};

const pauseStream = () => {
    video.pause();
    play.classList.remove('d-none');
    pause.classList.add('d-none');
```



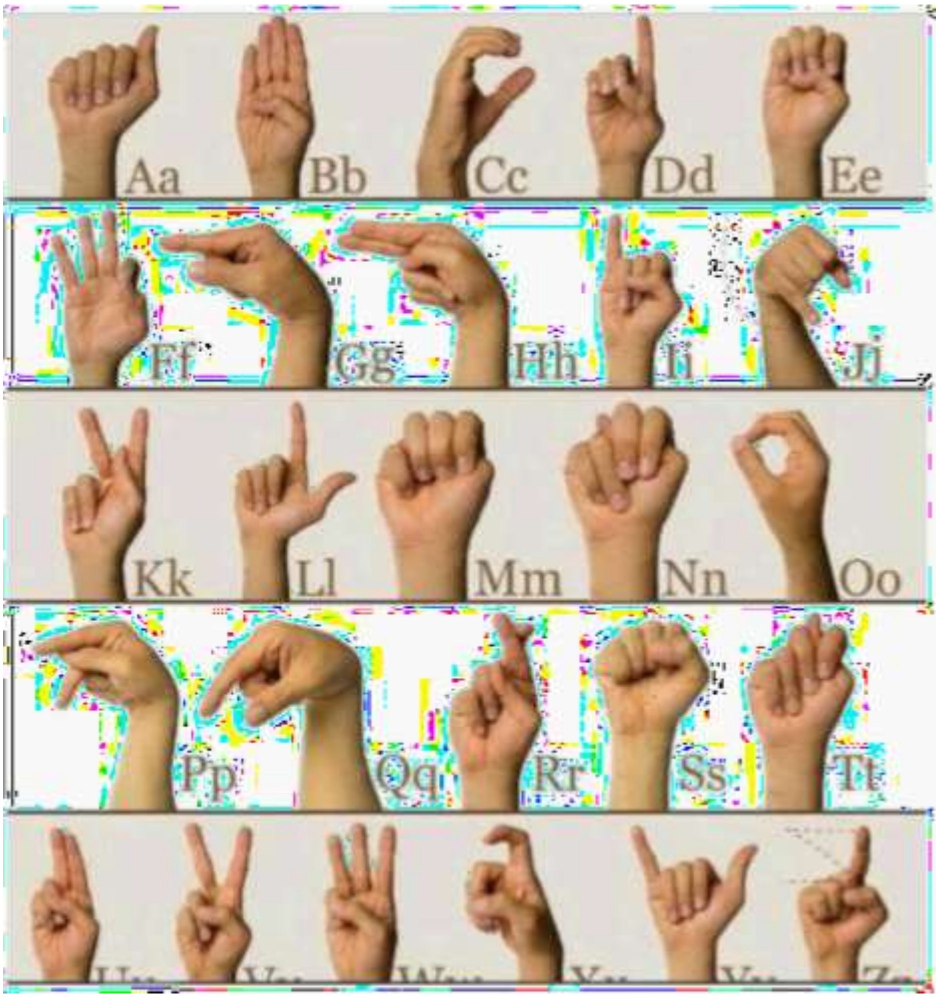
```
};

const doScreenshot = () => {
    canvas.width = video.videoWidth;
    canvas.height = video.videoHeight;
    canvas.getContext('2d').drawImage(video, 0, 0);
    screenshotImage.src = canvas.toDataURL('image/webp');
    screenshotImage.classList.remove('d-none');
};

pause.onclick = pauseStream;
screenshot.onclick = doScreenshot;
</script>

</body>
</html>
```

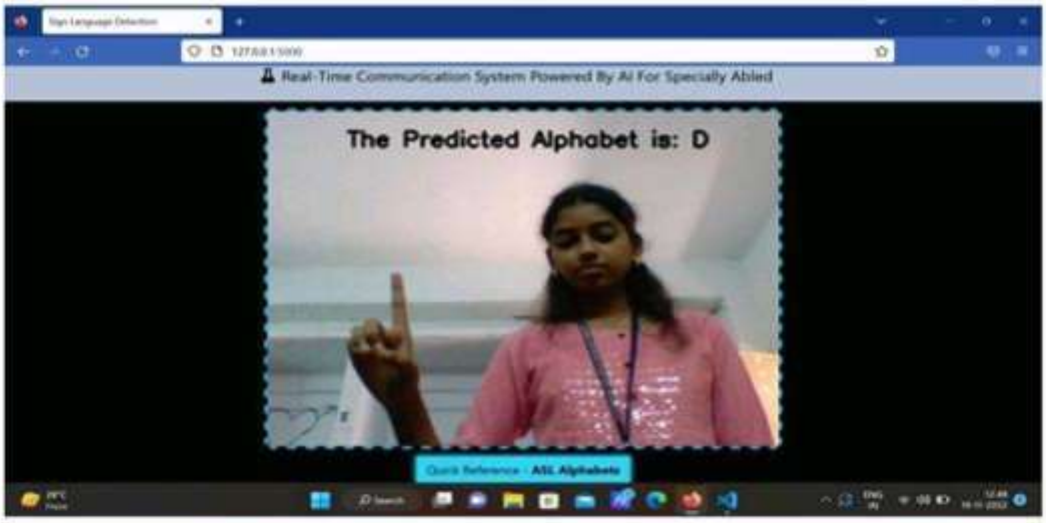






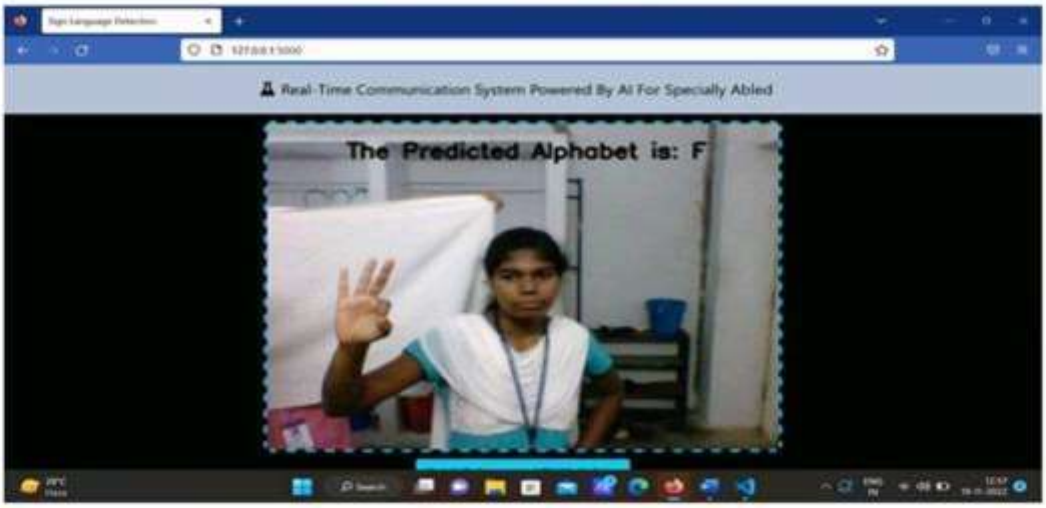
```
.ds-icon {
  --ds-icon-size: [illegible];
  display: flex;
  flex-shrink: 0;
  justify-content: center;
  align-items: center;
  font-size: var(--ds-icon-size);
  width: calc(var(--ds-icon-size) [illegible]);
  height: calc(var(--ds-icon-size) [illegible]);
  color: var(--ds-[illegible]);
}

.ds-icon-sm {
  --ds-icon-size: 1rem;
  width: calc(var(--ds-icon-size) [illegible]);
  height: calc(var(--ds-icon-size) [illegible]);
}

.ds-icon-[illegible] {
  --ds-icon-size: [illegible];
}

.ds-icon-[illegible] {
  --ds-icon-size: [illegible];
}

.ds-icon-[illegible] {
  --ds-icon-size: [illegible];
}

.ds-icon-[illegible] {
  [illegible]
  [illegible]-size) - .25rem;
}

.ds-icon.ds-icon-primary {
  color: var(--ds-white);
  background: var(--ds-primary);
}

.ds-icon.ds-icon-primary-light {
  color: var(--ds-primary);
  background: rgba(var(--ds-primary-rgb), .1);
}

.ds-icon.ds-icon-semi-white {
  color: var(--ds-gray);
  background: rgba(255, 255, 255, .5);
}

.ds-icon.ds-icon-rounded {
  border-radius: .5rem;
}

.ds-icon.ds-icon-circle {
  border-radius: 50%;
}
```


Aa
Bb
Cc
Dd
Ee
Ff
Gg
Hh
Ii
Jj
Kk
Ll
Mm
Nn
Oo
Pp
Qq
Rr
Ss
Tt
Uu
Vv
Ww
Xx
Yy
Zz



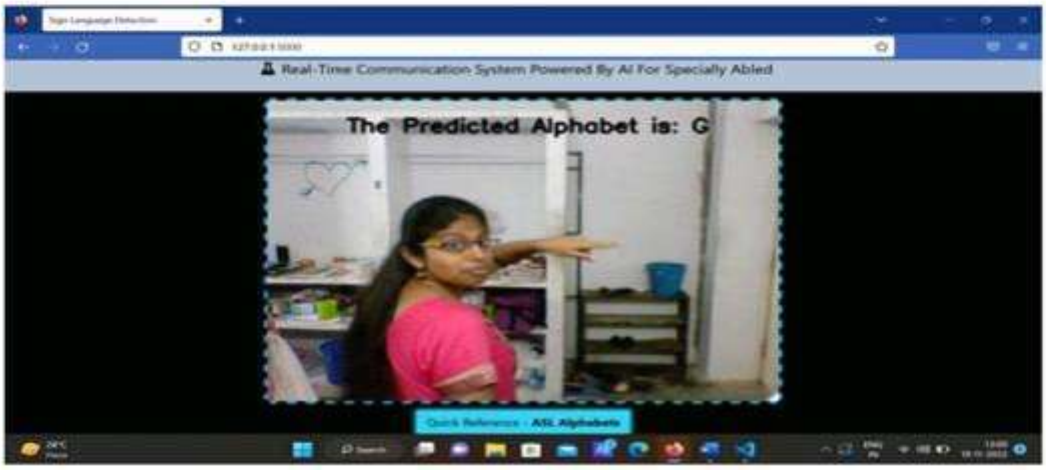
Real-Time Communication System Powered By AI For Specially Abled

The Predicted Alphabet is: D



Real-Time Communication System Powered By AI For Specially Abled

The Predicted Alphabet is: F



Real-Time Communication System Powered By AI For Specially Abled

The Predicted Alphabet is: G

PROJECT DEMO LINK

https://drive.google.com/drive/folders/1X7M2if1rQura9N6CrWdzT0rPOM-c4yBC

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-8147-1658910120