AI-POWERED REAL-TIME COMMUNICATION SYSTEM FOR PEOPLE WITH DISABILITIES

An Internship/Industrial Training/Practical Training Report

Submitted to the Faculty of Engineering of

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA,
KAKINADA

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

By

A.YASEEN
(20481A1201)

G. DHANRAJ
(20481A1246)

G. DINESH KUMAR
(20481A1254)

G. KIRANN MAHY
(20481A1257)

Under the Esteemed Guidance of
Smt Dr.D.N.V.S.L.S. INDIRA, M. Tech., Ph.D.,
Professor & HOD

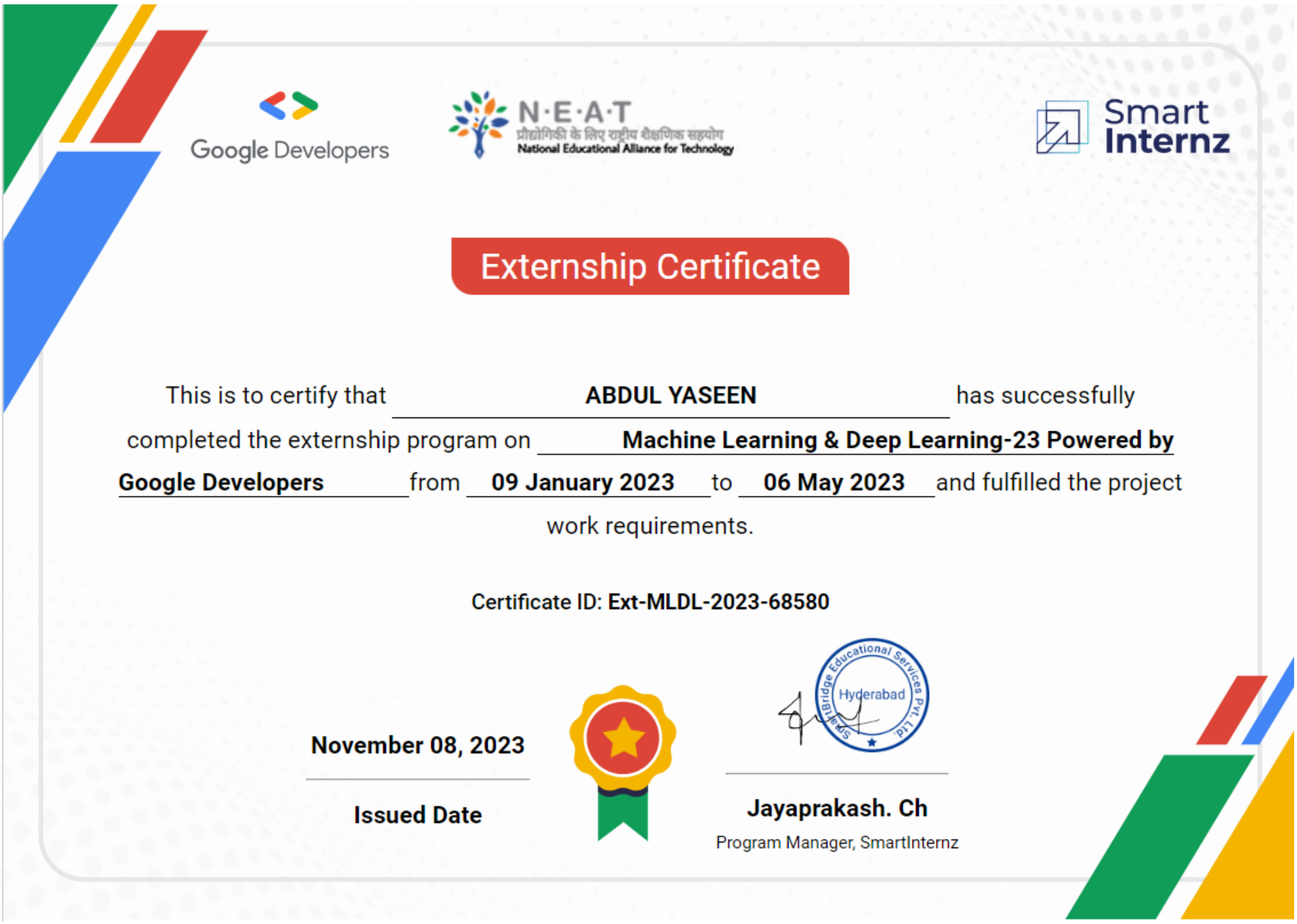
DEPARTMENT OF INFORMATION TECHNOLOGY

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU–521356

ANDHRA PRADESH

2023-2024



Google Developers

N·E·A·T

प्रौद्योगिकी के लिए राष्ट्रीय शैक्षणिक सहयोग

National Educational Alliance for Technology

Smart Internz

Externship Certificate

This is to certify that **ABDUL YASEEN** has successfully completed the externship program on **Machine Learning & Deep Learning-23 Powered by Google Developers** from **09 January 2023** to **06 May 2023** and fulfilled the project work requirements.

Certificate ID: **Ext-MLDL-2023-68580**

November 08, 2023

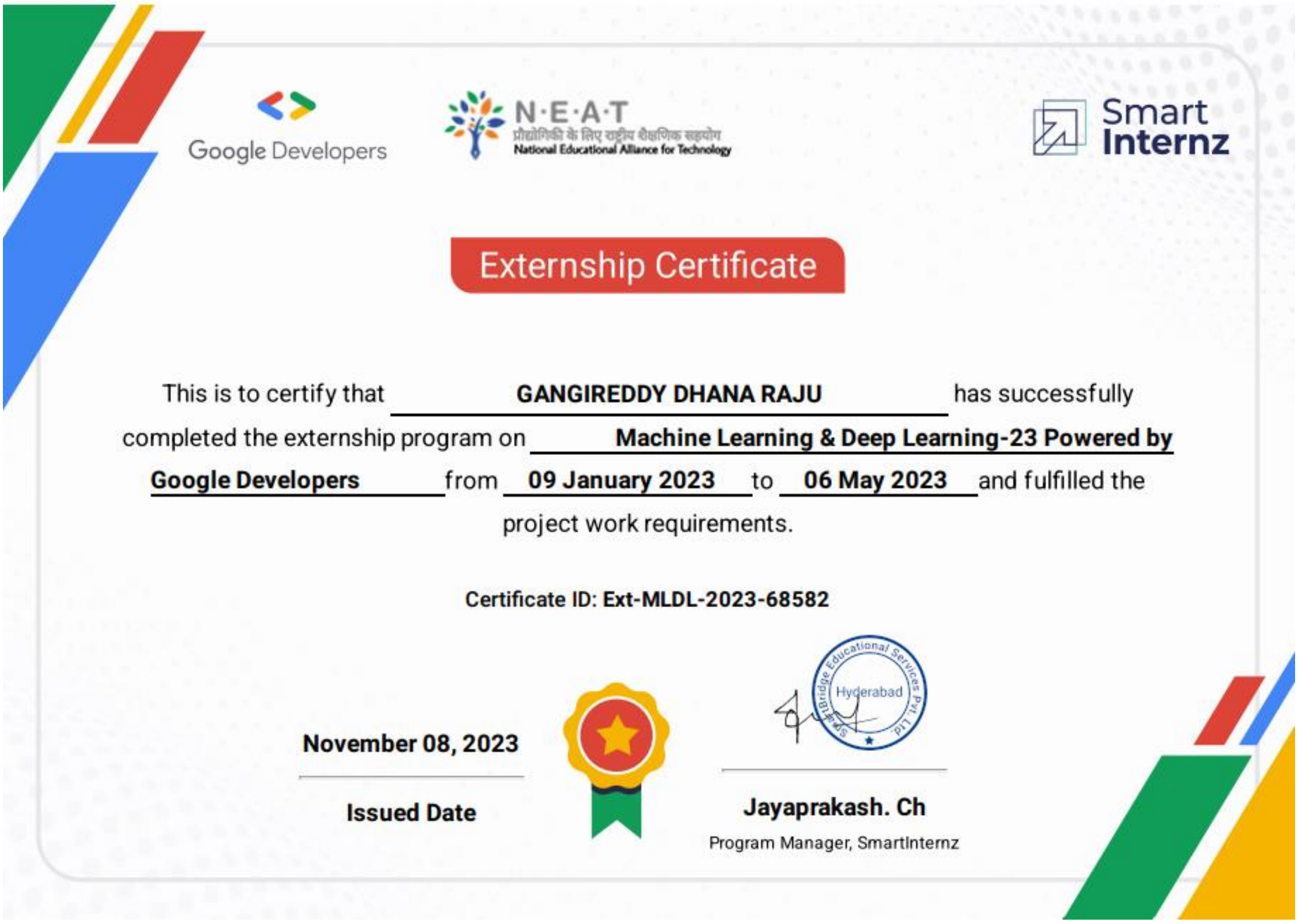
Issued Date

SmartBridge Educational Services Pvt. Ltd

Hyderabad

Jayaprakash. Ch

Program Manager, SmartInternz



Google Developers

N·E·A·T
प्रौद्योगिकी के लिए राष्ट्रीय शैक्षणिक सहयोग
National Educational Alliance for Technology

Smart Internz

Externship Certificate

This is to certify that **GANGIREDDY DHANA RAJU** has successfully completed the externship program on **Machine Learning & Deep Learning-23 Powered by Google Developers** from **09 January 2023** to **06 May 2023** and fulfilled the project work requirements.

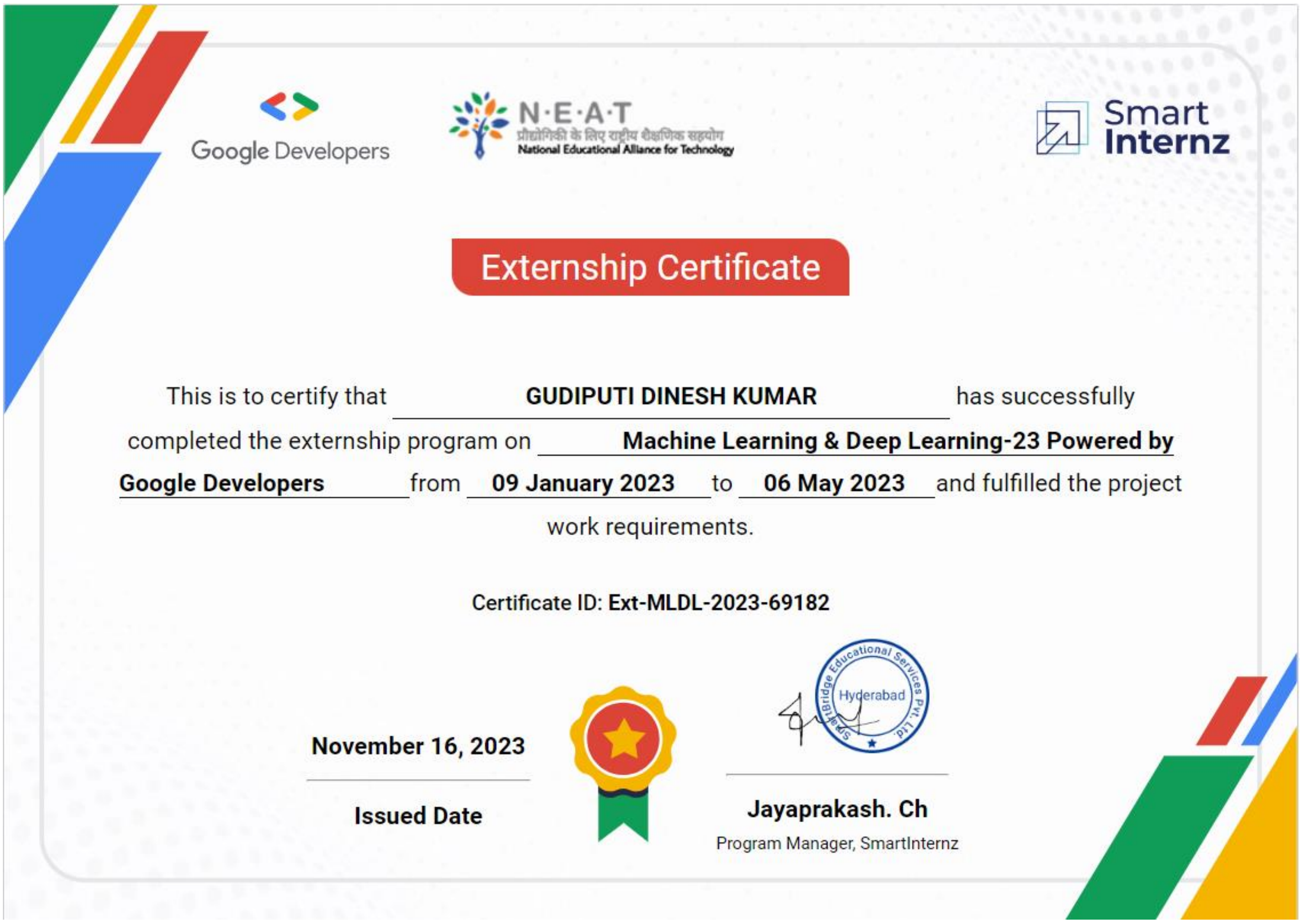
Certificate ID: Ext-MLDL-2023-68582

November 08, 2023

Issued Date

SmartBridge Educational Services Pvt. Ltd
Hyderabad

Jayaprakash. Ch
Program Manager, SmartInternz



Google Developers

N·E·A·T
प्रौद्योगिकी के लिए राष्ट्रीय शैक्षणिक सहयोग
National Educational Alliance for Technology

Smart Internz

Externship Certificate

This is to certify that **GUDIPUTI DINESH KUMAR** has successfully completed the externship program on **Machine Learning & Deep Learning-23 Powered by Google Developers** from **09 January 2023** to **06 May 2023** and fulfilled the project work requirements.

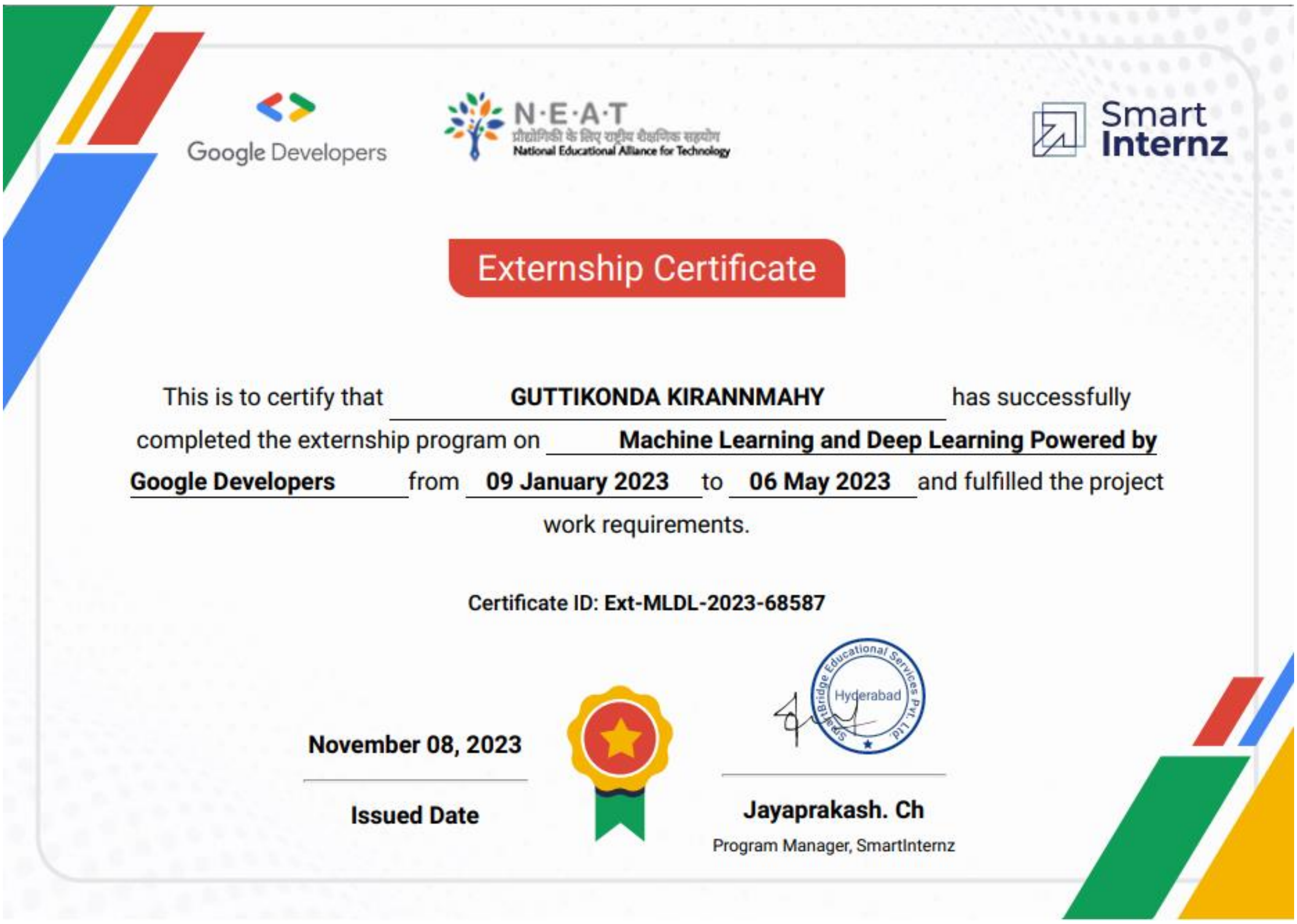
Certificate ID: **Ext-MLDL-2023-69182**

November 16, 2023

Issued Date

Smart Bridge Educational Services Pvt. Ltd.
Hyderabad

Jayaprakash. Ch
Program Manager, SmartInternz



Google Developers

N·E·A·T
प्रौद्योगिकी के लिए राष्ट्रीय शैक्षणिक सहयोग
National Educational Alliance for Technology

Smart Internz

Externship Certificate

This is to certify that **GUTTIKONDA KIRANNMAHY** has successfully completed the externship program on **Machine Learning and Deep Learning Powered by Google Developers** from **09 January 2023** to **06 May 2023** and fulfilled the project work requirements.

Certificate ID: **Ext-MLDL-2023-68587**

November 08, 2023

Issued Date

SmartBridge Educational Services Pvt. Ltd.
Hyderabad

Jayaprakash. Ch
Program Manager, SmartInternz

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report entitled **“AI-Powered Real-Time Communication System for People with Disabilities”**is a bonafide record of work carried out by **A.Yaseen (20481A1201), G.Dhanraj (20481A1246), G.Dinesh Kumar (20481A1254), G.Kirann Mahy (20481A1257)** as a part of internship in the partial fulfillment of the requirements for the award of the Degree of **Bachelor of Technology** in **Information Technology** of **Jawaharlal Nehru Technological University Kakinada,** Kakinada during the academic year 2023-2024.

Supervisor/Guide
(Smt Dr. D. N. V. S. L. S. INDIRA)

Head of the Department
(Smt Dr. D. N. V. S. L. S. INDIRA)

External Examiner

ACKNOWLEDGEMENT

I am glad to express my deep sense of gratitude to **Smart Bridge Mentors** for their guidance and cooperation in completing this externship project. Through this, I want to convey my sincere thanks to them for inspiring assistance during my project.

I express my heartful gratitude and deep in debt to our beloved Head of the Department **Dr. D. N. V. S. L. S. Indira** for her great help and encouragement in doing my project successfully.

I also express my gratitude to our principal **Dr. B. Karuna Kumar** for his encouragement and facilities provided during the course of project.

I express my heartful gratitude to all Faculty Members, all Lab Technicians, who helped me in all aspects of lab work. I thank one and all who have rendered help to me directly or indirectly in the completion of this project.

Project Associate

A. YASEEN	(20481A1201)
G. DHANRAJ	(20481A1246)
G. DINESH KUMAR	(20481A1254)
G. KIRANN MAHY	(20481A1257)

INDEX

LIST OF FIGURES

ABSTRACT

In a world reliant on seamless communication, individuals with disabilities often encounter barriers in accessing real-time communication platforms. In response to the pervasive communication challenges faced by individuals with disabilities, particularly those who are deaf and mute, this project introduces a pioneering solution titled "Real-Time Communication System for People with Disabilities." Utilizing cutting-edge technologies such as artificial intelligence, convolutional neural networks, and deep learning, the system interprets sign language, enabling seamless and effective interaction for users with speech and hearing impairments. Rooted in user-centered design and informed by extensive collaboration with the target community, the platform not only empowers individuals but also received validation through rigorous user testing, affirming its efficacy and usability. Beyond immediate applications, the project envisions a societal shift towards inclusivity, leveraging technology to bridge communication gaps. Future-oriented features include multi-digit recognition, symbol recognition, and real-time translation, showcasing the project's commitment to ongoing advancements in accessibility. This initiative stands as a beacon for transformative technology, promising to significantly enhance the quality of life for individuals with disabilities and contributing to a more inclusive and connected global community.

CHAPTER-1
INTRODUCTION

In a world that increasingly relies on seamless communication and connectivity, individuals with disabilities often face significant barriers in accessing and participating in real-time communication platforms. To address this pressing issue and promote inclusivity, we undertook the development of a ground breaking project titled "Real-Time Communication System for People with Disabilities.". The primary objective of this project was to create an innovative communication platform tailored specifically to cater to the unique needs and challenges faced by people with disabilities.

By leveraging cutting-edge technologies and incorporating user-centered design principles, our real-time communication system aimed to empower individuals with disabilities, enabling them to interact, share information, and engage in social activities with unprecedented ease and efficiency. Moreover, we present the results of extensive user testing, involving individuals with disabilities who actively engaged with our machine learning-based communication system. Their valuable feedback and insights provide crucial validation of the system's efficacy, usability, and potential to enhance the quality of life for our target audience.

1.1 OVERVIEW:

Our project aims to develop a real-time communication system using Artificial Intelligence (AI) to empower individuals who are deaf and mute. The system will leverage AI algorithms to interpret sign language and convert it into text or speech output, facilitating seamless communication for those with speech and hearing impairments. Through user-centered design and collaboration with the deaf and mute community, we seek to create an inclusive and transformative platform that enhances their lives and fosters greater connectivity with the world.

1.2 PURPOSE:

Our purpose is to create a more inclusive society, where technology facilitates accessibility for all, regardless of their abilities. Through this project, we aspire to make the world a place where everyone's voice is heard, and individuals with disabilities can engage actively in daily interactions and experiences.

CHAPTER-2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

The issue at hand revolves around the need to improve communication for deaf and mute individuals by leveraging sign language recognition technology. Recent studies have demonstrated the potential of this technology to bridge the communication gap between specially abled individuals and the wider community.

Researchers have explored various approaches to recognize hand gestures represented through video feeds or images. For instance, Taniya Sahana et.al proposed a system based on multiscale density features, achieving 98.2% accuracy using the MLP model. Another study focused on Indian hand gestures, utilizing blob detection, skin color recognition, and template matching to classify 26 classes of gestures with a mobile application.

One notable technique involves using deep convolutional neural networks, such as AlexNet and VGGNet, to process hand images after tracking and recognition. Such efforts aim to enable two-sided communication efficiently for deaf and blind individuals. Several solutions are currently in use, including technology like smartphones and laptops for text-based communication, sign language interpreters for fluent communication in sign language, and deaf individuals with residual hearing communicating with blind individuals using speech.

2.2 PROPOSED SOLUTION

Our project aims to create an innovative solution to bridge the communication gap between the speech and hearing impaired individuals and the wider community. The key objectives of our research are as follows:

1. **Communication Gap Reduction:**
   We strive to design and develop a system that facilitates seamless communication between speech and hearing impaired individuals and the rest of the world. By leveraging advanced computer vision and deep learning techniques, we aim to enable effective interaction and understanding.
2. **Deaf-Dumb Communication:**
   Our proposed communication system will act as a bridge between deaf and mute individuals and those who can communicate verbally. Through the use of hand gestures, captured and classified using a convolutional neural network (CNN), our system will facilitate effective communication and understanding.

CHAPTER-3
THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM:

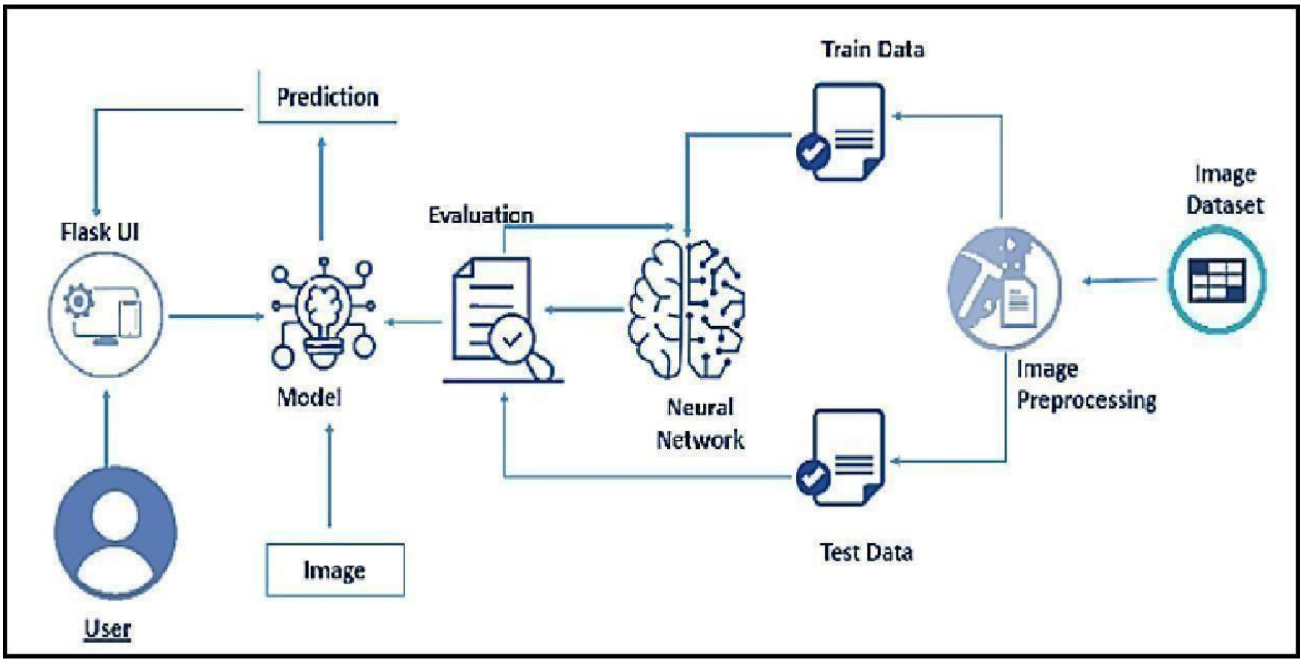


Fig3.1.1 Block diagram

Anchored by the Image Dataset, housing hand gesture images from 'A' to 'I,' the subsequent Image Preprocessing block ensures optimal formatting, priming the dataset for neural network training. The Train Data and Test Data divisions guide the convolutional neural network (CNN), our project's neural network model, through an impactful learning journey. Following training, the Evaluation stage rigorously tests the model's proficiency on unseen data, leading to precise Image Prediction where the model interprets new gestures. The Flask UI (User Interface) acts as a user-friendly gateway, facilitating interaction with the neural network. Finally, the User block encapsulates the end-users who upload their gestures, completing a symbiotic loop. This comprehensive diagram encapsulates the crux of our innovative system, demonstrating how advanced technologies harmonize to empower individuals with speech and hearing impairments, fostering inclusive communication and connectivity.

SOFTWARE REQUIREMENTS

The software components needed for the development of this project are as follows:

1.Desktop GUI:

Anaconda Navigator, which provides an integrated development environment (IDE) for data science tasks and managing Python environments.

2.Operating System:

Windows 10, as the base operating system for running the project.

3.Front-End:

HTML and CSS will be used for developing the user interface and enhancing user experience.

4.Programming Language:

Python is the primary programming language used for developing the project. It's widely used in machine learning and AI due to its rich libraries and frameworks for data manipulation, deep learning, and computer vision

3.2 HARDWARE SPECIFICATIONS

The development of this project requires the following hardware components:

1.Processor:

Intel CoreTM i3-5300H with a clock speed of 2.4GHz.

2.RAM Size:

8 GB DDR (Random Access Memory) to handle data and computations efficiently.

3.System Type:

X64-based processor architecture for compatibility and performance.

4.Webcam:

An integrated or external webcam with Full HD support for capturing high-quality images and video feed.

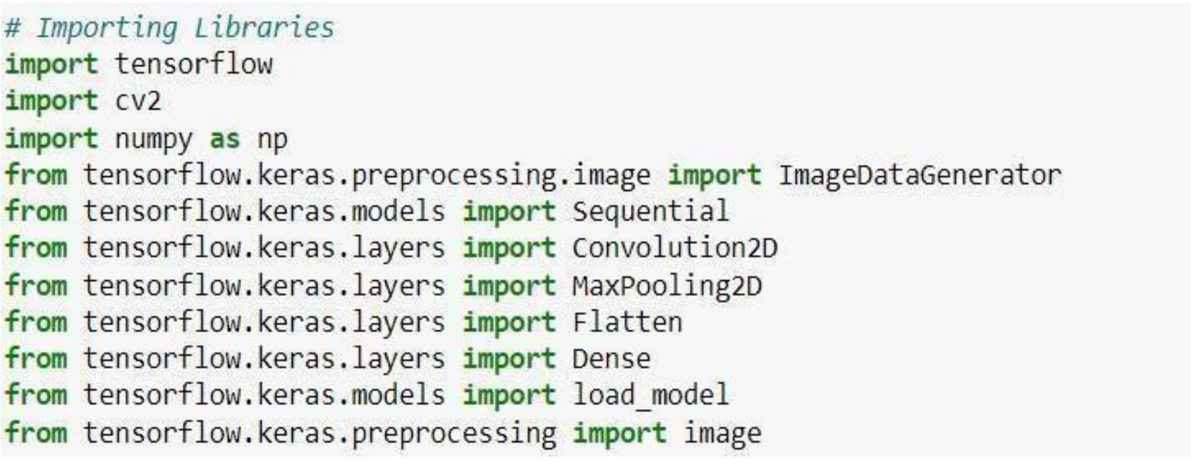
CHAPTER-4

EXPERIMENTAL INVESTIGATIONS

4.1 Exploratory Data Analysis:-

Importing the Libraries:

In this code snippet, we import the necessary libraries and modules required for building and training a deep learning model for image classification. The TensorFlow library provides the foundation for creating and training the neural network, while OpenCV is used for image processing tasks. Other modules from TensorFlow's Keras API are imported to construct the model architecture and handle image data.

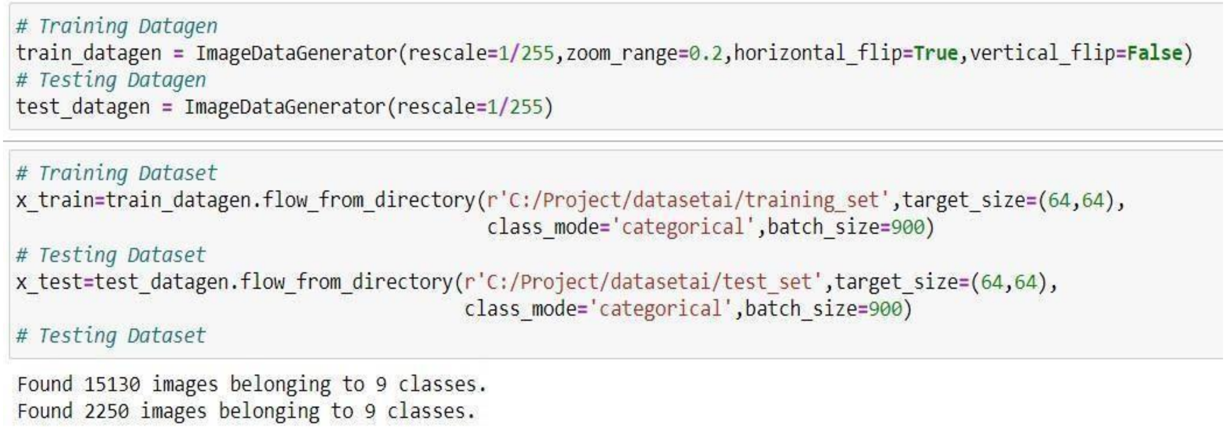


```
# Importing Libraries
import tensorflow
import cv2
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

Fig. 4.1.1. Importing the libraries

Data Augmentation:

We modify and augment the images in both the train and test data. The only operation for test data we perform is rescaling the images to 1./255 to. By rescaling images in the test and training set to 1/255 within the ImageDataGenerator, the pixel values are appropriately transformed to meet the requirements for efficient and effective model training. We do not 7 need to augment the test data since we won’t use it for training. We also perform horizontal and vertical flipping to provide robust orientations that take all kinds of directions a hand can be present in front of the system into consideration. Before passing the images to the model, we set shuffle as false in the test when flowing from the directory in order to stop the model from learning wrong labels. We also create a validation set with 20% of the test data in order to test the model how it did each epoch on unseen data.



```
# 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'C:/Project/datasetai/training_set',target_size=(64,64),
                                          class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'C:/Project/datasetai/test_set',target_size=(64,64),
                                        class_mode='categorical',batch_size=900)
# Testing Dataset
```

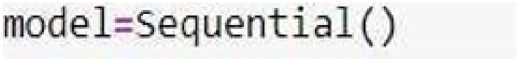
```
Found 15130 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

Fig. 4.1.2. Data Augmentation

4.2 Building And Testing Models:

Initializing the Model:

We use the sequential model, which allows us to add layers sequentially one after the other. The model is then built layer by layer to form a deep learning architecture capable of recognizing hand gesture images representing letters from 'A'to 'I'.

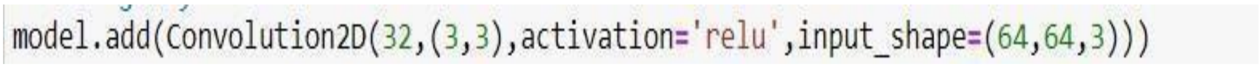


```
model=Sequential()
```

Fig.4.2.1. Initializing the model

Adding Convolution Layer :

By adding this convolutional layer, the model begins to learn lower-level image features from the input images, paving the way for subsequent layers to learn more abstract and complex features to identify the hand gestures effectively.

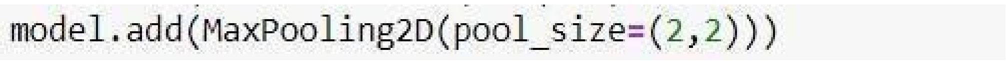


```
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

Fig.4.2.2. Adding Convolution Layer

Adding Pooling layer:

Max-pooling helps in achieving spatial invariance and reducing the number of parameters in the model, which can lead to faster computation and reduced overfitting. By adding this max pooling layer after the convolutional layer, the model learns to capture and retain the most significant features from the input images, enabling more efficient and effective recognition of hand gestures.

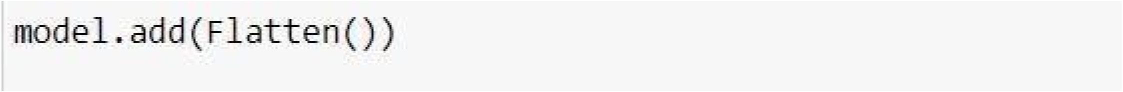


```
model.add(MaxPooling2D(pool_size=(2,2)))
```

Fig.4.2.3. Adding Pooling Layer

Adding Flatten Layer :

The flatten layer takes the output from the previous layer (in this case, the output from the max- pooling layer) and reshapes it into a 1D vector. This flattening process collapses all the spatial dimensions and preserves the information as a continuous sequence, which can be used as input to the fully connected layers.

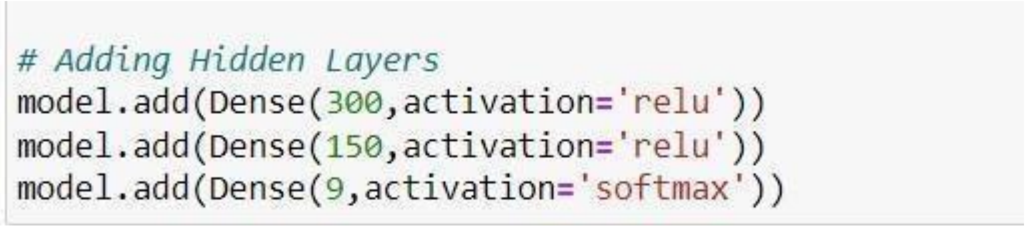


```
model.add(Flatten())
```

Fig.4.2.4. Adding Flatten Layer

Adding Hidden Layers :

With the addition of hidden layers and the output layer, the neural network is now capable of capturing more complex patterns and making predictions for each class, making it suitable for recognizing hand gestures representing letters 'A' to 'I' in the real-time communication system for deaf and mute individuals.

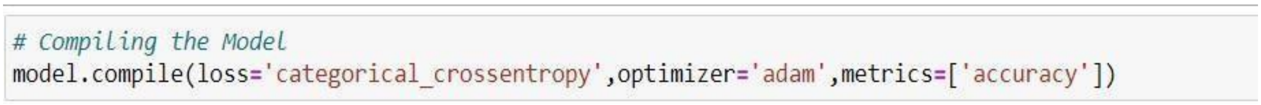


```
# Adding Hidden Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
```

Fig.4.2.5. Adding Hidden Layers

Compiling the Model:

By compiling the model with the appropriate loss function, optimizer, and evaluation metric, we are now ready to start the training process. The model will learn from the augmented and preprocessed training data and gradually improve its accuracy and predictive capabilities during the training epochs.

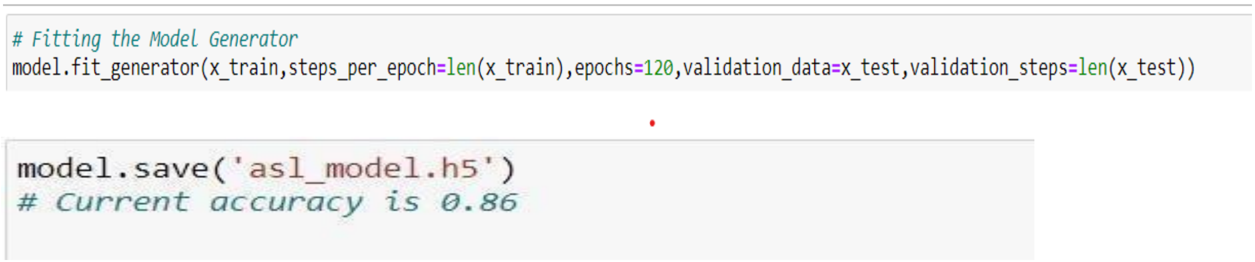


```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Fig.4.2.6. Compiling the Model

Fit and Save the Model:

During the training process, the model's performance will be monitored using the categorical cross-entropy loss and accuracy metrics on both the training and validation datasets. After training is complete, the trained model will be saved to the 'asl_model.h5' file, allowing us to use it later for making predictions or further fine-tuning, if necessary.



```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=120,validation_data=x_test,validation_steps=len(x_test))
```

```
model.save('asl_model.h5')
# Current accuracy is 0.86
```

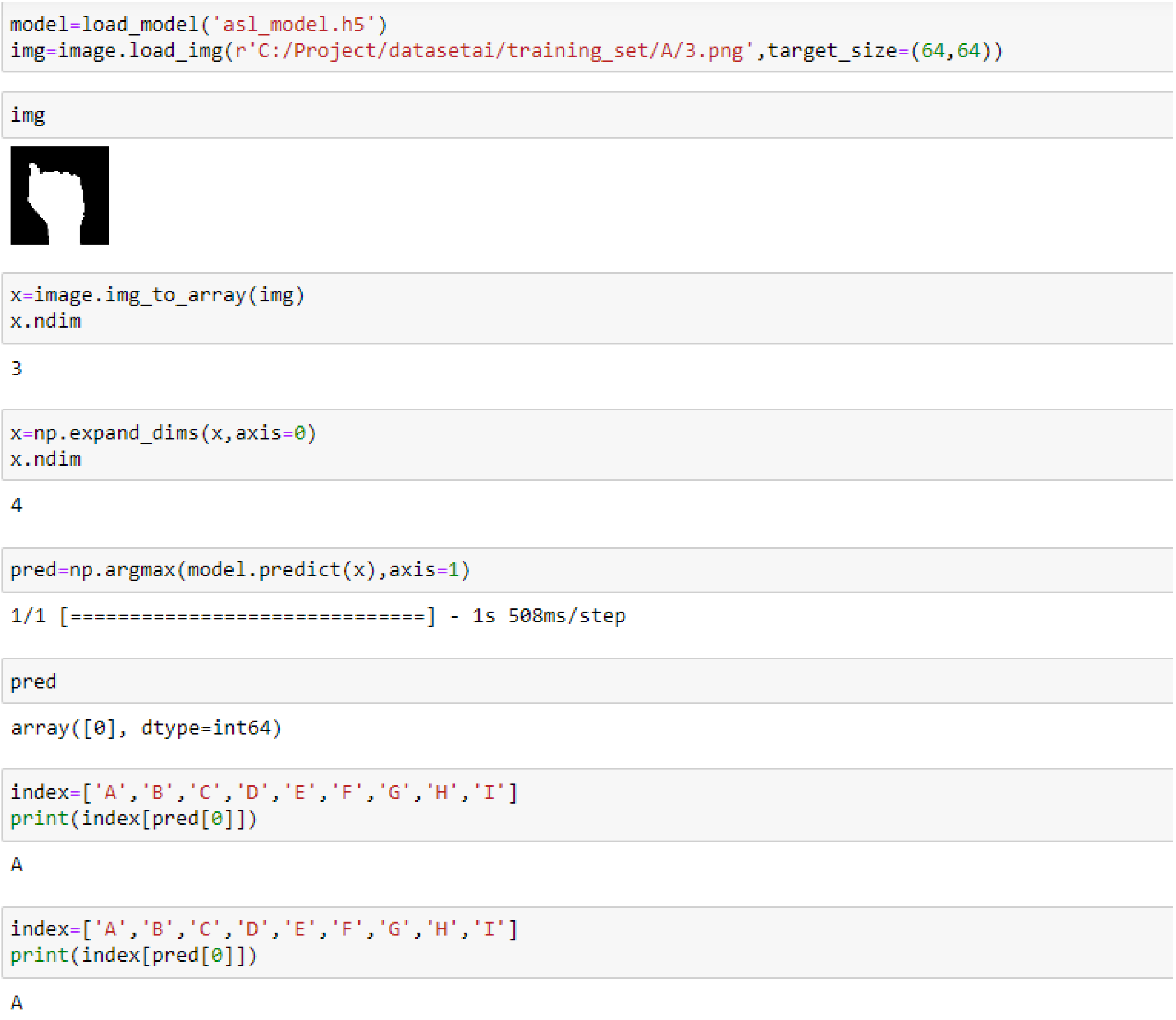
Fig.4.2.7. Fit and Save the Model

Test the Model :

To test the trained model, we can use the saved model file ('asl_model.h5') and load it back into memory. Once the model is loaded, we can use it to make predictions on new data, which in this case will be the testing dataset.

The load_model() function is used to load the model from the 'aslpng1.h5' file into memory. Then, we use the evaluate() function to assess the model's performance on the testing dataset (x_test). This will give us the test loss and test accuracy, which will help us understand how well the model generalizes to unseen data.

By testing the model on the testing dataset, we can determine its real-world performance and ensure that it effectively recognizes hand gestures representing letters 'A' to 'I' in the real-time communication system for deaf and mute individuals.



```
model=load_model('asl_model.h5')
img=image.load_img(r'C:/Project/datasetai/training_set/A/3.png',target_size=(64,64))
```

```
img
```

```
x=image.img_to_array(img)
x.ndim
```

```
3
```

```
x=np.expand_dims(x,axis=0)
x.ndim
```

```
4
```

```
pred=np.argmax(model.predict(x),axis=1)
```

```
1/1 [==============================] - 1s 508ms/step
```

```
pred
```

```
array([0], dtype=int64)
```

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

```
A
```

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

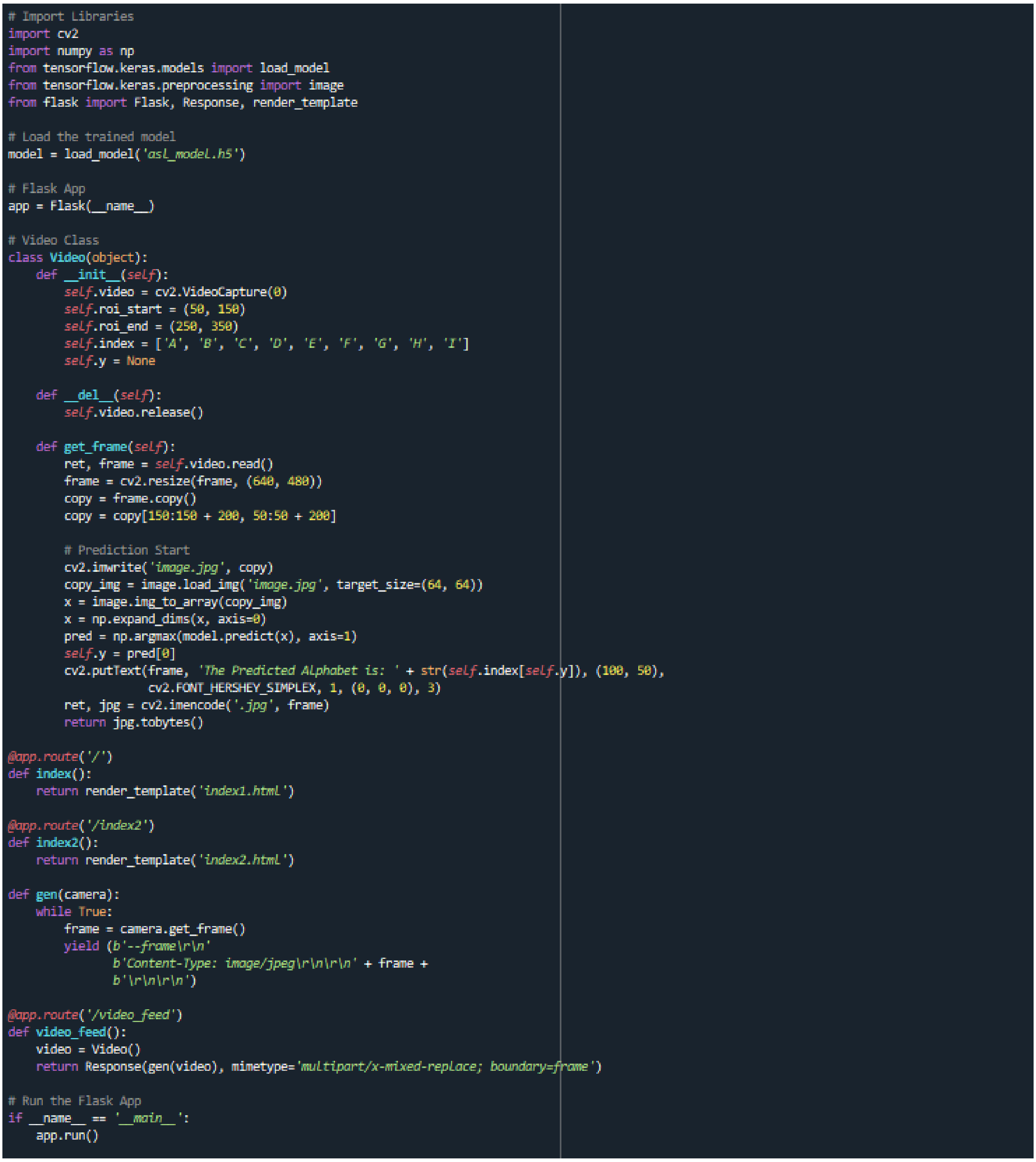
```
A
```

Fig.4.2.8. Test the Model

4.3 Executing the Application:

Building Python and Flask Code:

Python code is written to implement the system, utilizing the model. Additionally, Flask, a web application framework, is used to build the frontend for the system, making it user-friendly and accessible via a web interface.



```python
# Import Libraries
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask, Response, render_template

# Load the trained model
model = load_model('asl_model.h5')

# Flask App
app = Flask(__name__)

# Video Class
class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        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(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()

@app.route('/')
def index():
    return render_template('index1.html')

@app.route('/index2')
def index2():
    return render_template('index2.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame +
               b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary=frame')

# Run the Flask App
if __name__ == '__main__':
    app.run()
```

Fig 4.3.1. Flask Code

CHAPTER-5

FLOW CHART

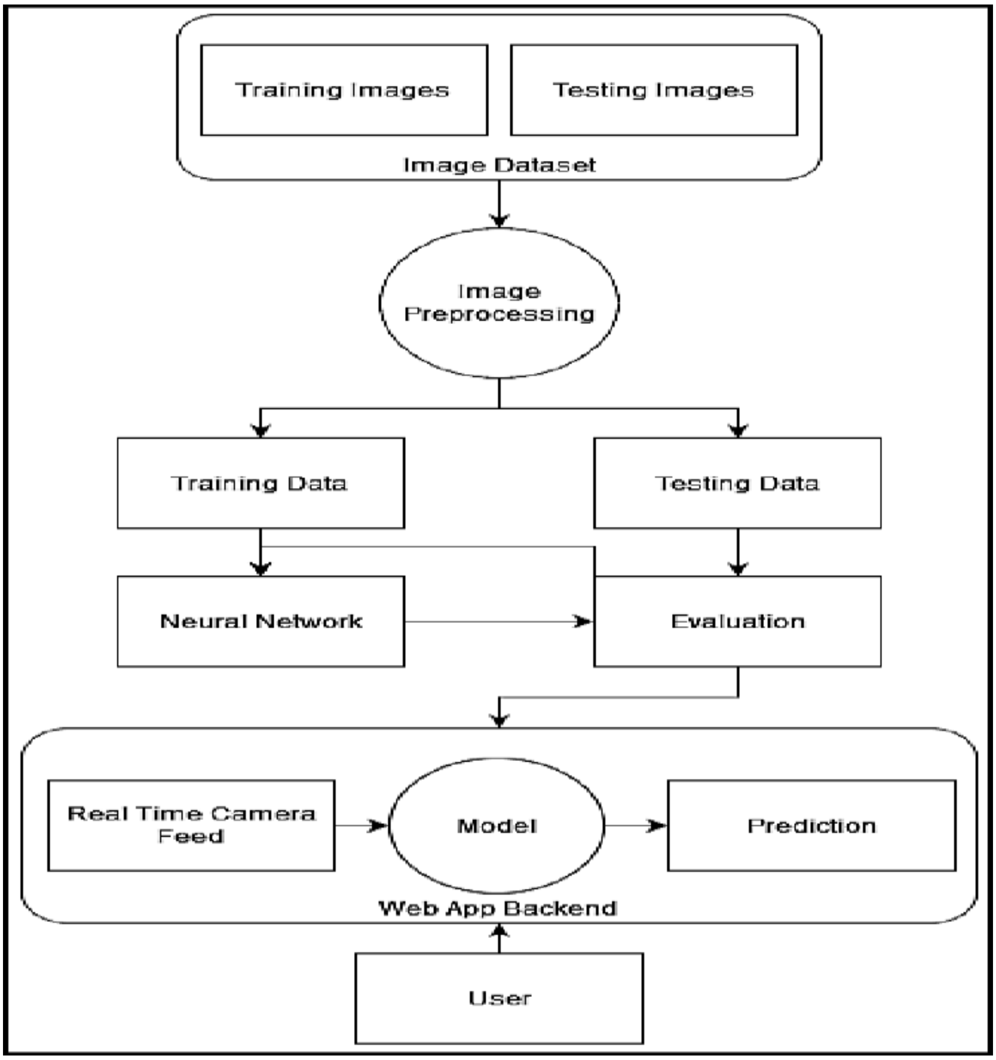
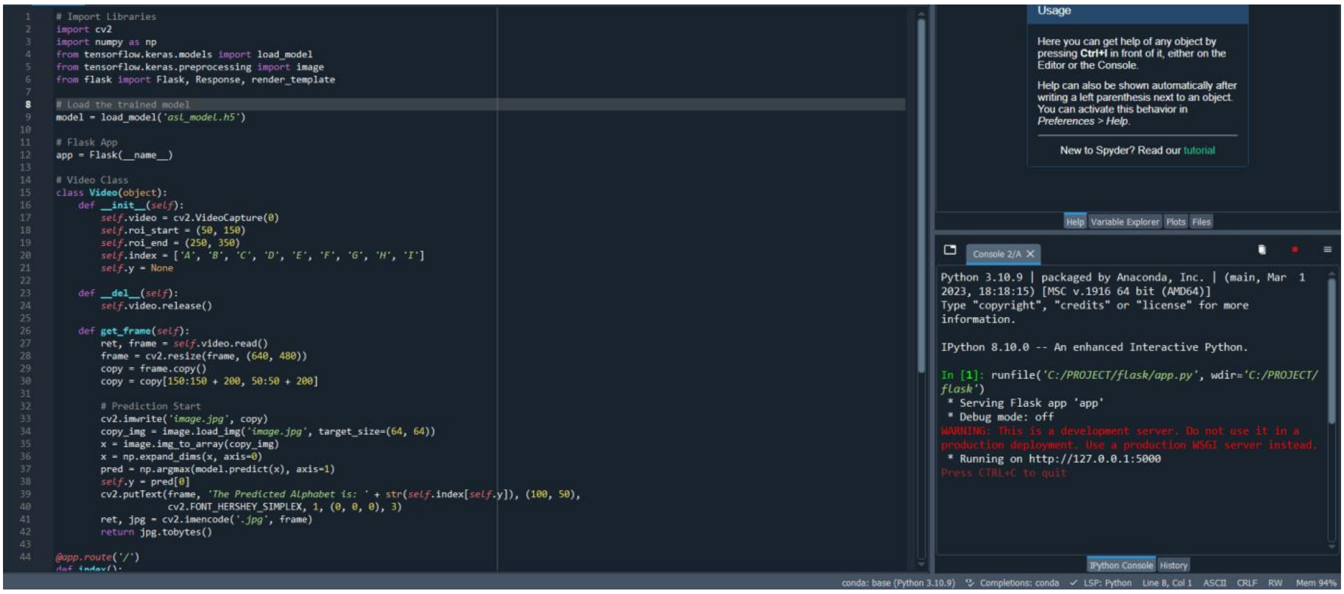


Fig 5.1 Flow Chart of our system

CHAPTER-6

RESULTS

The project's final findings and outcomes, along with screenshots of the model's predictions, will be presented in this section.



```
# Import Libraries
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask, Response, render_template

# Load the trained model
model = load_model('asl_model.h5')

# Flask App
app = Flask(__name__)

# Video Class
class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        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(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()

@app.route('/')
def index():
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

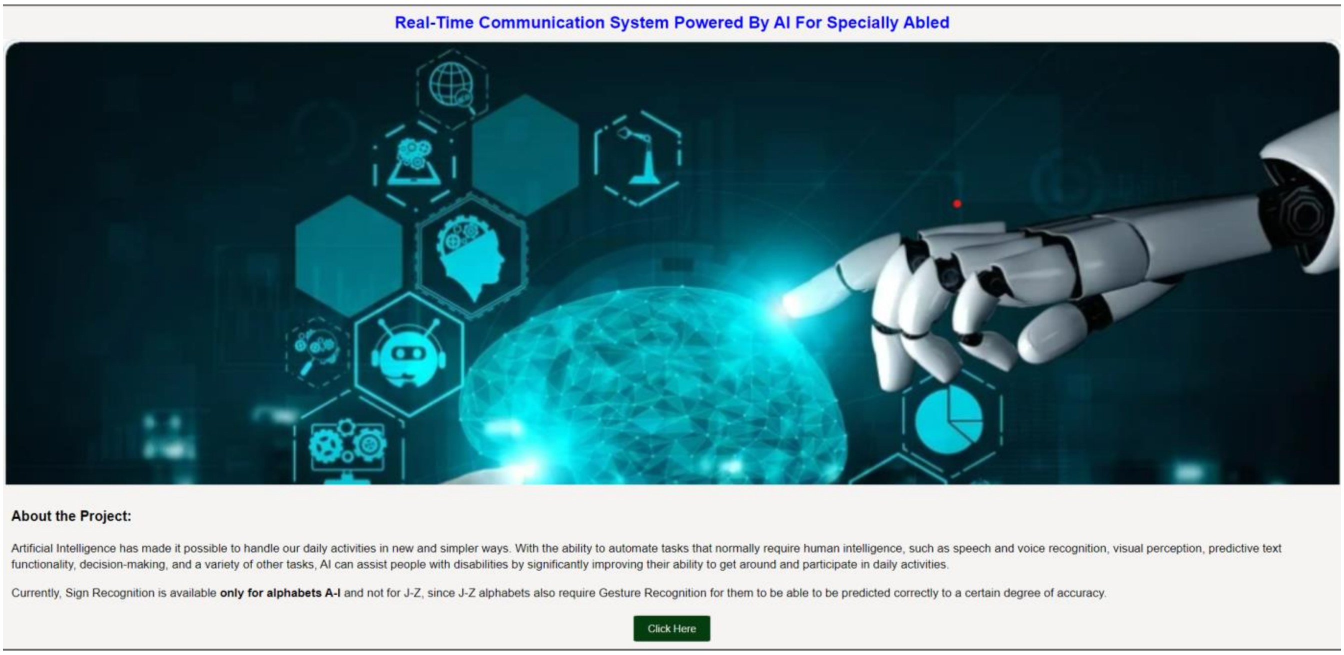
```
Python 3.10.9 | packaged by Anaconda, Inc. | (main, Mar  1
2023, 18:18:15) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more
information.

IPython 8.10.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/PROJECT/flask/app.py', wdir='C:/PROJECT/
flask')
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a
production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Fig. 6.1. Running the application

Now paste the URL on the browser, you will redirect to index.html page.



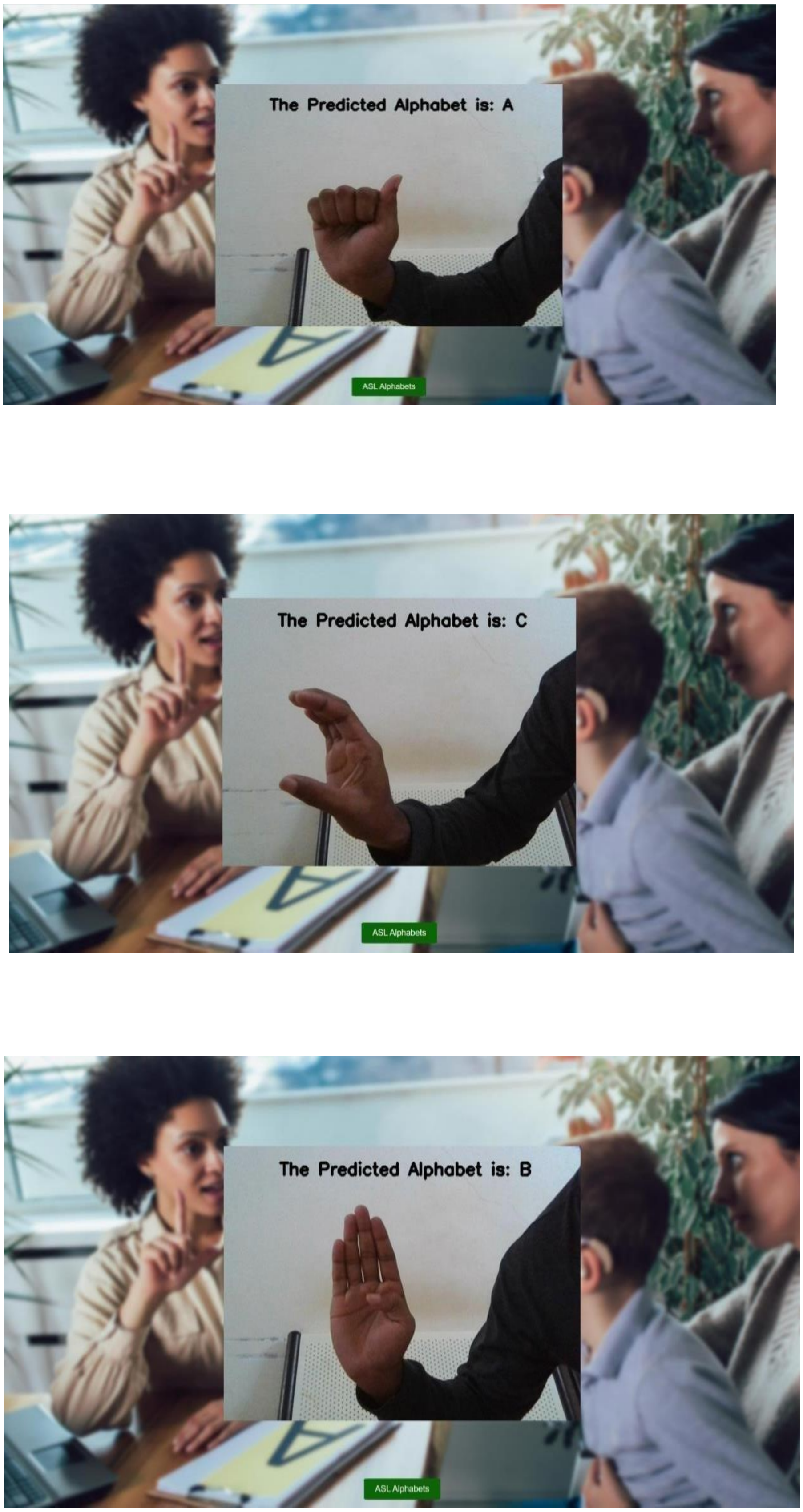
Real-Time Communication System Powered By AI For Specially Abled

About the Project:

Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly improving their ability to get around and participate in daily activities.

Currently, Sign Recognition is available **only for alphabets A-I** and not for J-Z, since J-Z alphabets also require Gesture Recognition for them to be able to be predicted correctly to a certain degree of accuracy.

Fig.6.2. Website Home Page



The Predicted Alphabet is: A

The Predicted Alphabet is: C

The Predicted Alphabet is: B

Fig.6.3. Final Results

CHAPTER-7

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

1. **Automation:** The real-time communication system reduces the need for manual work in understanding hand gestures, enabling efficient and automated communication for deaf and mute individuals.
2. **Enhanced Accuracy:** The deep learning model used in the system is more accurate than the average human in recognizing hand gestures, leading to more reliable communication outcomes.
3. **Scalability:** The system can handle large amounts of data, making it suitable for widespread usage and accommodating a diverse range of users.
4. **Accessibility:** The application can be accessed from any device with an internet connection, making it accessible to users across various platforms.

DISADVANTAGES

1. **Limited Alphabet Range:** The current model only supports recognition of hand gestures representing alphabets from 'A' to 'I'. The extension to include more alphabets requires additional gesture recognition techniques.

2. **Gesture Input Limitation:** The system may face challenges in recognizing alphabets beyond 'I' as they may require more complex gesture inputs from the user, necessitating further research and development.

3. **Low Quantity/Quality of Images:** The system's accuracy is affected by the dataset's quantity and quality. Increasing the dataset size and improving image quality could enhance the accuracy of predictions.

4. **Occasional Errors:** The system may occasionally encounter errors in hand gesture recognition, especially in challenging lighting conditions or noisy environment.

CHAPTER-8

APPLICATIONS

The proposed solution has various potential applications, including but not limited to:

1. **Smart Home Automation for Accessibility:**

   Real-time communication systems, when combined with AI, can enable individuals with disabilities to control their home environment. Voice and gesture recognition technologies allow them to operate lights, appliances, doors, and other devices through voice commands or gestures, promoting accessibility within their living spaces.

2. **Vision Assistance:**

   AI-powered real-time communication systems can provide valuable assistance to individuals with visual impairments. By employing advanced algorithms, these system scan identify objects, read text, describe scenes, and offer audio guidance, allowing visually impaired individuals to navigate their surroundings more effectively.

3. **Hearing Assistance:**

   Real-time communication systems with AI capabilities can enhance communication for individuals with hearing impairments. Speech recognition and transcription algorithms can convert spoken words into text, enabling individuals to read conversations in real-time or through visual displays.

CHAPTER-9
CONCLUSION

This project successfully demonstrates a web application that harnesses the power of deep learning, computer vision, and neural networks to recognize American Sign Language (ASL). Developed using Flask and Python, the application achieves impressive results, with a training accuracy of 98% and a testing accuracy of approximately 99%.The proposed system offers scalability, capable of handling a large number of users without compromising performance. Being a web application, it enjoys compatibility with various devices that can run a web browser, making it easily accessible to users across platforms.

The real-world applications of this project are diverse and impactful. Apart from facilitating communication for deaf and mute individuals, it can find use in recognizing numbers, processing bank cheques, and other practical scenarios. Its potential for integration into various domains makes it a valuable addition to assistive technologies. The project's success in bridging the communication gap between deaf individuals and the rest of society is commendable. By allowing two-way communication through sign language, it empowers deaf individuals to express themselves effectively and fosters inclusive interactions in society.

CHAPTER-10

FUTURE SCOPE

1. **Extended Alphabet Recognition:** With the introduction of gesture recognition, the web application can be easily expanded to recognize letters beyond 'I'. By incorporating more sign language symbols and gestures, the system can support a broader range of communication, allowing users to express themselves more comprehensively.

2. **Multi-Digit Recognition:** Building upon the existing capabilities, the application can be further developed to recognize multiple digits. This enhancement would enable users to communicate numbers and perform arithmetic operations using sign language, expanding the system's utility in various contexts.

3. **Symbol Recognition:** Expanding the system to recognize other symbols and expressions used in sign language would enable users to convey complex messages, emotions, and concepts more effectively.

4. **Interface Control:** Leveraging gesture recognition, the web application can go beyond language recognition and enable users to control software and hardware interfaces. This feature could empower specially-abled individuals to interact with technology and devices using intuitive hand gestures.

5. **Real-Time Translation:** Implementing real-time translation features would allow the application to convert sign language into spoken language or text, facilitating seamless communication between deaf individuals and the general public.

6. **Customization and Personalization:** Incorporating customization options would enable users to adapt the application to their specific sign language preferences and needs, making it more user-friendly and tailored to individual requirements.

CHAPTER-11

BIBILIOGRAPHY

[1] Environment Setup: https://www.youtube.com/watch?v=5mDYijMfSzs

[2] Keras Image Processing Doc: https://keras.io/api/preprocessing/image/

[3] CNN using Tensorflow: https://www.youtube.com/watch?v=umGJ30-15_A

[4] OpenCV Basics of Processing Image: Click Here

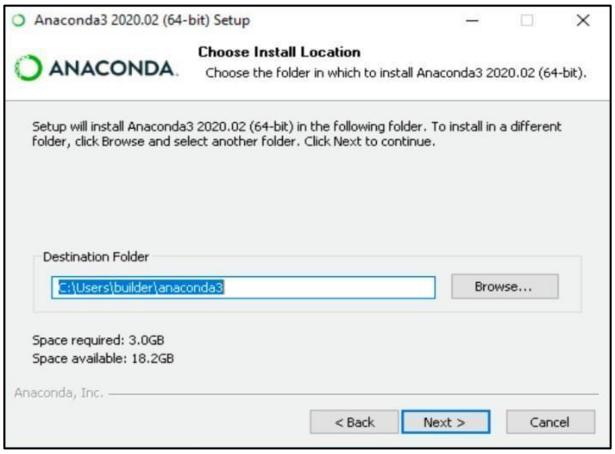
[5] Flask Basics: https://www.youtube.com/watch?v=lj4I_CvBnt0

CHAPTER-12

APPENDIX

How to install the Anaconda Navigator:

1. Download the Anaconda installer.
2. RECOMMENDED: Verify data integrity with SHA-256. For more information hashes, see what about cryptographic hash verification?
3. Double click the installer to launch.
4. Click Next.
5. Read the licensing terms and click "I Agree".
6. Select an install for "Just Me" unless you're installing for all users (which require windows administrator privileges) and click next.
7. Select a destination folder to install Anaconda and click the Next button.



Choose Install Location

Choose the folder in which to install Anaconda3 2020.02 (64-bit).

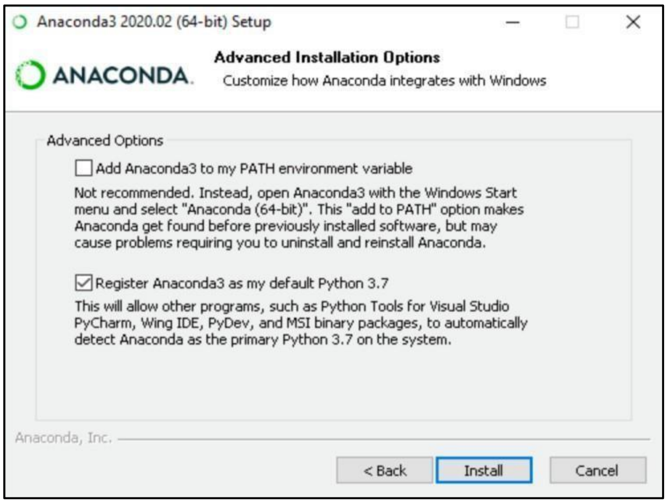
Setup will install Anaconda3 2020.02 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Users\builder\anaconda3

Space required: 3.0GB
Space available: 18.2GB

Fig12.1 Installation Step1

8. Choose whether to add Anaconda to your PATH environment variable. Were commend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.



Anaconda3 2020.02 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

☐ Add Anaconda3 to my PATH environment variable

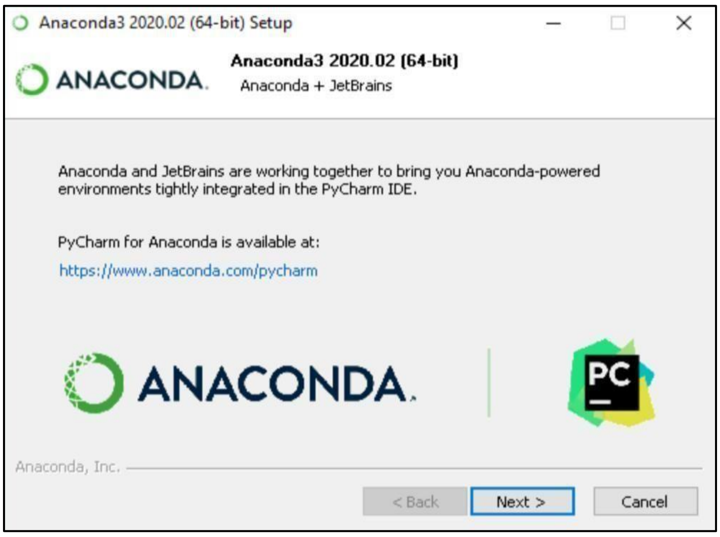
Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda3 as my default Python 3.7

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.

Fig12.2 Installation step 2

1. Choose whether to register Anaconda as your default Python. Unless you plan on installing and running multiple versions of Anaconda or multiple versions of Python, accept the default and leave this box checked.
2. Click the Install button. If you want to watch the packages Anaconda is installing, click Show Details.
3. Click the Next button.
4. Optional: To install PyCharm for Anaconda, click on the link to [https://www.anaconda.com/pycharm.](https://www.anaconda.com/pycharm)



Anaconda3 2020.02 (64-bit) Setup

Anaconda3 2020.02 (64-bit)

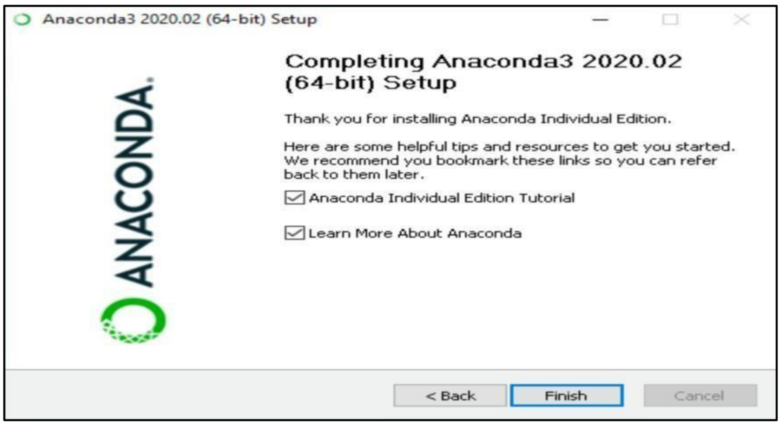
Anaconda + JetBrains

Anaconda and JetBrains are working together to bring you Anaconda-powered environments tightly integrated in the PyCharm IDE.

PyCharm for Anaconda is available at:

https://www.anaconda.com/pycharm

Fig12.3 Installation step 3



Completing Anaconda3 2020.02 (64-bit) Setup

Thank you for installing Anaconda Individual Edition.

Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.

☑ Anaconda Individual Edition Tutorial

☑ Learn More About Anaconda

Fig12.4 Installation step 4

Installing Python Packages:

1. **Tensorflow:**

   This package is used as backend support to Keras.

2. **Keras :**

   This package is used for building Neural Network layers.

3. **OpenCV**:

   This package is used for image processing.

4. **Flask** :

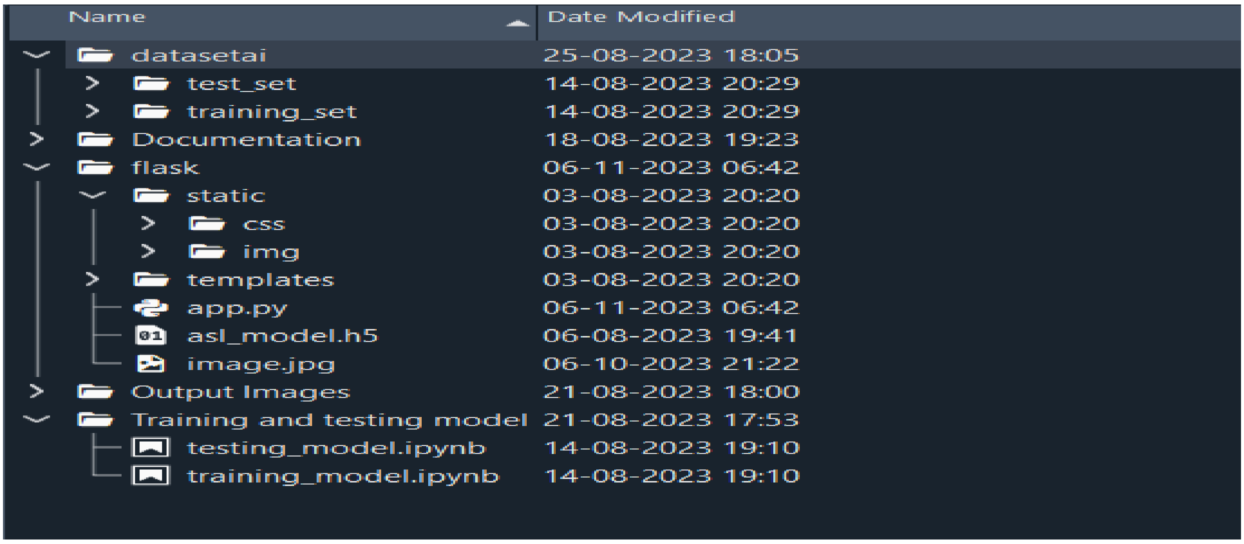
   To build a web application.

If you are using anaconda navigator, follow the below steps to download the required packages:

Open anaconda prompt as administrator.

- Type "pip install tensorflow==1.14.0" and click enter.
- Type "pip install keras=2.2.4" and click enter.
- Type "pip install opencv-python" and click enter.
- Type "pip install imutils" and click enter.
- Type "pip install flask" and click enter

PROJECT STRUCTURE



```
Name                                 Date Modified
datasetai                            25-08-2023 18:05
    test_set                         14-08-2023 20:29
    training_set                     14-08-2023 20:29
Documentation                        18-08-2023 19:23
flask                                06-11-2023 06:42
    static                           03-08-2023 20:20
        css                          03-08-2023 20:20
        img                          03-08-2023 20:20
    templates                        03-08-2023 20:20
    app.py                           06-11-2023 06:42
    asl_model.h5                     06-08-2023 19:41
    image.jpg                        06-10-2023 21:22
Output Images                        21-08-2023 18:00
Training and testing model           21-08-2023 17:53
    testing_model.ipynb              14-08-2023 19:10
    training_model.ipynb             14-08-2023 19:10
```

Fig 12.5 Project Structure

How to run project:

Step1: Open anaconda prompt.

Step2: Change the working directory to project directory.

Step3: Move to the directory named flask.

Step4: Launch the application app.py using **python app.py** command.

Step5: Application runs at **http:localhost:5000** port

Step6: Open the application in any browser. It will open web interface.

Step7: In that we show the Sign language at the webcam. It recognize the Sign Language and it will print the respective Alphabet, there only.

Step9: For Stopping the execution press CTRL+C in the flask console.

Project Link : Click Here

CHAPTER-13

PROJECT WORK MAPPING WITH PROGRAMME OUTCOMES

Pos	1	2	3	4	5	6	7	8	9	10	11	12
Project	2	3	3	2	3	3	2	2	2	2	3	3

PROGRAMMEOUTCOMES(POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME OUTCOMES	Mapping HIGH: 3/ MEDIUM: 2/LOW: 1	JUSTIFICATION
1	**2**	To apply the knowledge of mathematics.
2	**3**	By considering the problems of in real-time communication for individuals with disabilities and analyze those problems and developing Algorithms.
3	**3**	This project meets the desired specification of the society.
4	**2**	We have created this user interface by using Flask.
5	**3**	By using the Python and Pandas we have created data-frames for storing data and user friendly interface.
6	**3**	In this developing process we were able to meet the local challenges as well as global challenges.
7	**2**	This interface does not provide benefits to all types of users which help for the society.
8	**2**	It will provide some ethical, social behavior in some aspects.
9	**2**	The work is done by team to function on multi-disciplinary team.
10	**2**	As our project is done in all aspects like communicating and documenting effectively.
11	**3**	Our project is developed by Python Programming language and it will engage in lifelong learning.
12	**3**	We find a solution to our problem by developing an application, which is effective for individuals with speech and hearing impairments

PROGRAM SPECIFIC OUTCOMES

PSOs	1	2
PROJECT	3	2

Student will be able to

1. Organize, maintain and protect IT Infrastructural resources.
2. Design and Develop web, mobile, and smart apps based software solutions to the real world problems.

PROGRAM SPECIFIC OUTCOMES	Mapping HIGH:3/ MEDIUM: 2/LOW:1	JUSTIFICATION
1	**3**	Providing seamless communication for individuals with disabilities is a challenge. We made it easy by developing this project.
2	**2**	Software usage in Providing seamless communication for individuals with disabilities can reduce the time and increase efficiency.