An

Internship Report on

Conversation Engine For Deaf And Dumb Using IBM Watson

Submitted
in partial fulfillment of the requirements for
the award of the degree of

BACHELOR OF TECHNOLOGY

in

Electronics and Communication Engineering

By

G.Gayathri

19R11A0462

Department of Electronics and Communication Engineering
GEETHANJALI COLLEGE OF ENGINEERING & TECHNOLOGY, (UGC AUTONOMOUS)
Cheeryal (V), Keesara (M), Medchal Dist, Hyderabad– 501 301
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, accredited by NAAC and NBA,New Delhi)

2019-2023

GEETHANJALI COLLEGE OF ENGINEERING & TECHNOLOGY

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the G. internship report titled **Conversation Engine For Deaf and Dumb Using IBM Watson** being submitted by **G. Gayathri** bearing roll number **19R11A0462** in partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology** in ***Electronics and Communication Engineering*** is a record of bonafide work carried out through online mechanism.

Prof. K. Somasekhara Rao
Professor &Dean-Student Affairs

Dr. P. Srihari
Professor

Dr. S. Suryanarayana
HoD-ECE

Prof. B. Hari Kumar
Dean, SE&CE

This is to certify that

G.Gayathri

has completed all courses in the learning path

Conversation Engine for Deaf and Dumb using IBM Watson

a learning path on smartbridge.skillsnetwork.site

Powered by **IBM Developer Skills Network**

Issued by

The SmartBridge

Conversation Engine for Deaf and Dumb using IBM Watson consists of the following courses:

- Design Thinking
- DevOps Fundamentals
- Agile Training Module

Ravi Ahuja
Program Director
Skills Network, IBM

Issued on:

June 16, 2021

Authenticity of this certificate can be validated by going to:

https://smartbridge.skillsnetwork.site/certificates/87be2e3b-7906-4da4-b1c3-35de3cfccee7

Amaravadee Kattasani
Founder & CEO
TheSmartBridge

Smart Internz

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

Externship Certificate

This to certify that

Gayathri G

has successfully completed the externship program on **Artificial Intelligence powered by IBM** from 23 April 2021 to 31 May 2021 and fulfilled the project work requirements.

Certificate ID: Ext-AI-2021-4851

August 23, 2021

Issued Date

Jayaprakash. Ch
Program Manager

AKNOWLEDGEMENT

I, the student of Geethanjali College of Engineering of technology in the stream Electronics and Communication Engineering would like to express my sincere gratitude to the Principal **Dr. S. Udaya Kumar**, and Chairman **Dr. G. R. Ravinder Reddy** for their encouragement to take up this internship.

I also would like to thank **Dr. S. Suryanarayana**, Head of the Department of Electronics and Communication Engineering; **Dr. P. Srihari** (Professor); **Dr. Shoukath Ali** (Associate Professor), who constantly provided us guidance to complete the internship successfully. I feel delighted to be under the roof of the most intellectual faculty members. A sincere gratitude towards the college management for providing the best education which ever field we are interested in.

Regards,

G. Gayathri(19R11A0462)

CONTENTS

ABSTRACT

Artificial Intelligence and Machine Learning is a burning topic in the tech industry. Artificial Intelligence (AI) is impacting our daily lives in many ways. AI is everywhere, from gaming stations to maintaining complex information at work. Computer Engineers and Scientists are working hard to impart intelligent behavior in the machines making them think and respond to real-time situations. Tech giants like Google and Facebook have placed huge bets on Artificial Intelligence and Machine Learning and are already using it in their products. Machine Learning (ML) is a subset of Artificial Intelligence. ML is a science of designing and applying algorithms that are able to learn things from past cases.

The project is based on Conversation Engine for Deaf and Dumb using AI and ML. 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. About nine billion people in the world are deaf and dumb. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as converting speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

LIST OF FIGURES

LIST OF TABLES

SYMBOLS/ ABBREVIATIONS

AI-Artificial Intelligence

ML-Machine Learning

CNN- Convolutional Neural Network

OpenCV- Open Source Computer Vision Library

HTML-Hyper Text Markup Language

CHAPTER 1. INTRODUCTION

1.1 Organization Background

Geethanjali College of Engineering and Technology has taken the initiative to train their students in their respective fields of engineering in the best possible way. The college has geared itself to impart and equipped to take on the responsibility of providing the student community with the technical education on par with the latest advances in the respective field with a great ambience. It moulds the students to face any sort of competencies.

In this journey to accomplish and integrate their vision and mission towards perfection, the college has collaborated and established the excellent partnership with the esteemed institution **"Smart Bridge"**.

Smart Bridge is an educational and technological institution whose vision is to bridge the gap between academics and industry. They provide the learning programs on the emerging technologies like Artificial Intelligence, Internet of things, Robotic Process Automation, Cloud Computing and many more. The other wing of the smart bridge **" Smart Internz",** has provided the opportunity for the students to work on the real time projects.

They have always motivated the students to develop the quality of teamwork, leadership, ethics and team responsibility. They have excellent mentors for respective technology. They imparted innovation and gave a great exposure on how the industry level working seems like.

Their **methodology** is as follows:

1. **LEARN**: Choosing the emerging technology and imparting the best possible knowledge about the tools in regard to the field.
2. **PRACTICE**: "Practice makes man perfect"; the popular saying goes. While going through the internship; they also provide assignments to assess ourselves how far we are able to cope up with it. They encourage the students to express their doubts.

3. **INTERN**: A team based real time projects are provided to the students after the training.

1.2 Training Objective

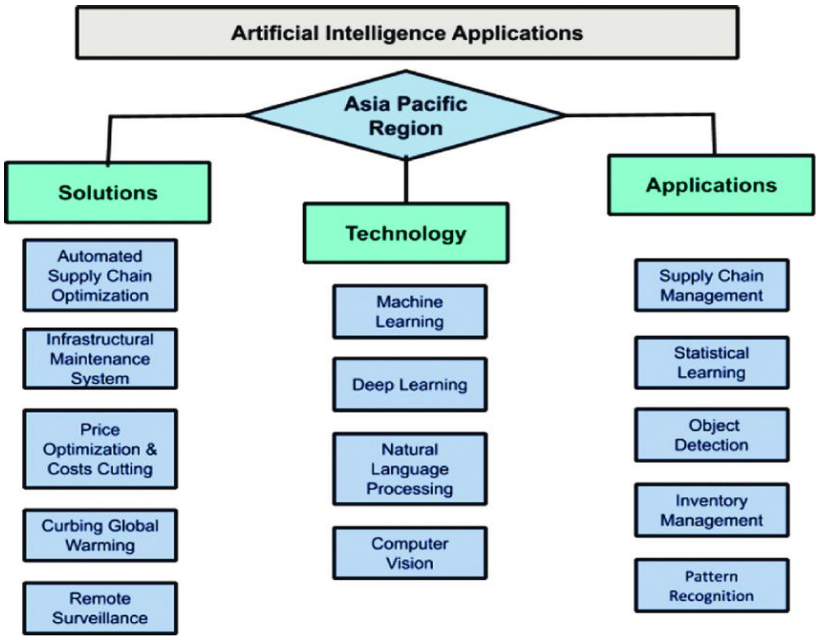
I have chosen the Artificial Intelligence and Machine Learning. I belong to the Electronics and Communication Engineering, but I also wanted to concentrate towards IT and software field other than my core area. As Artificial Intelligence and Machine Learning was most emerging and trending technology, I was attracted towards this. The main objective of this internship is to understand the the basics of ML, Deep Learning and AI and Neural Networks understanding and programming .

It is definitely a wonder how the world is revolving around AI.
These things have increased the fascination to learn this technology and no wonder I got to know many things from this internship program.

To make place for myself in this constantly progressing world ,it would be beneficial for me.
Taking up an internship would be the best thing where I can witness the real- world scenarios.

1.3 Domain of Internship

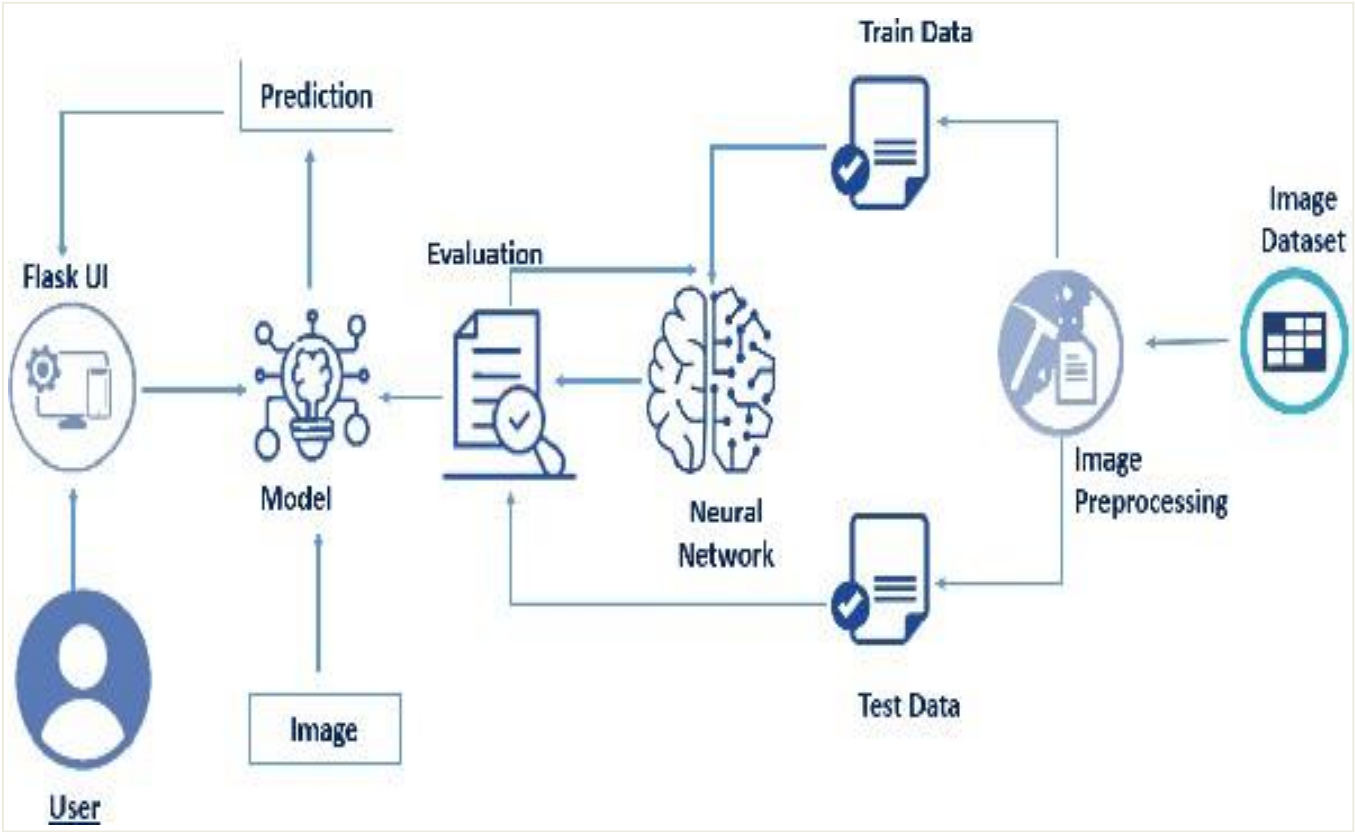
AI is an introductory course in Artificial Intelligence. The goal is to acquire knowledge on intelligent systems and agents, formalization of knowledge, reasoning with and without uncertainty, machine learning and applications at a basic level. The main research topics in AI include: problem solving, reasoning, planning, natural language understanding, computer vision, automatic programming, machine learning, and so on. Of course, these topics are closely related with each other.



(1)

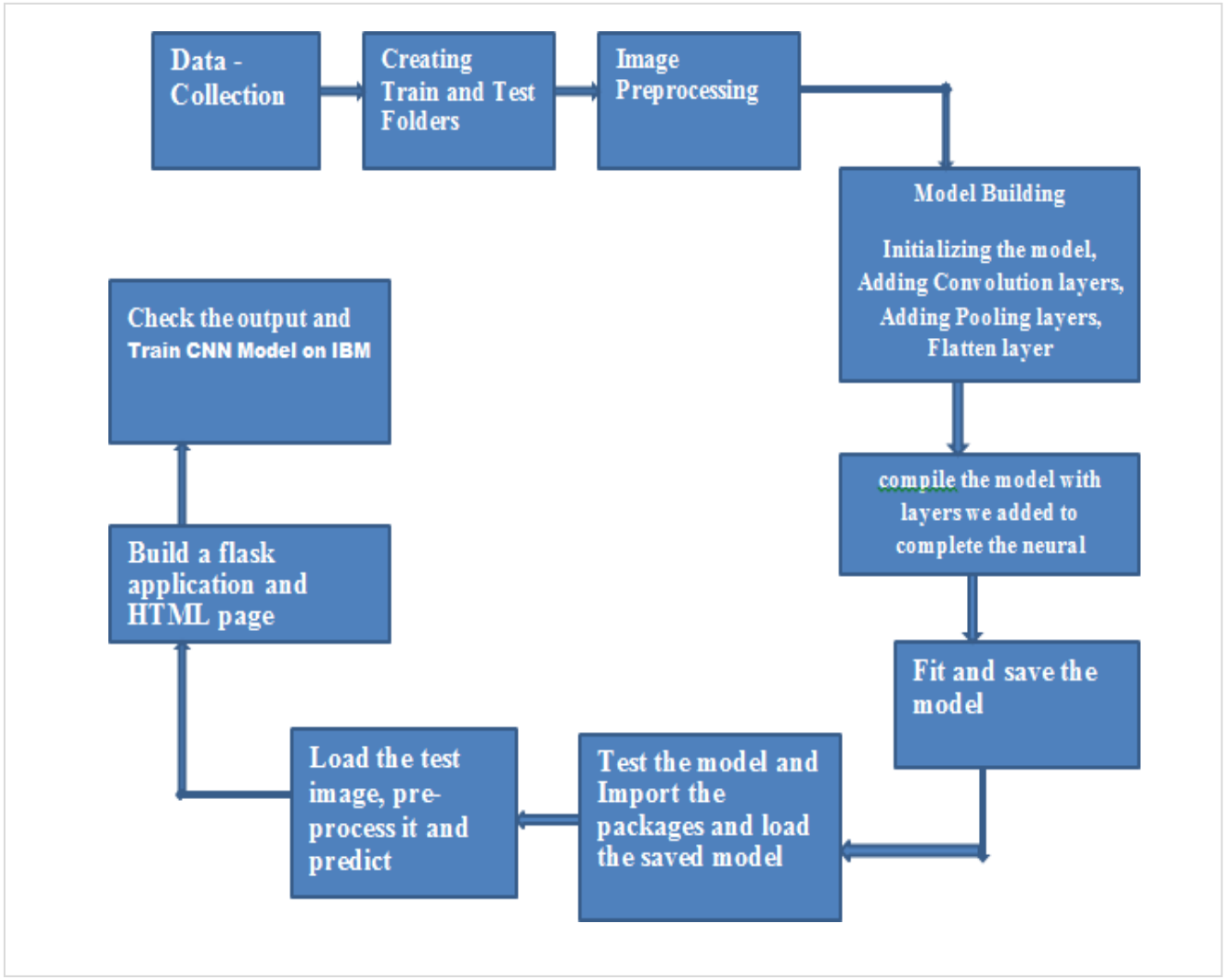
CHAPTER 2. TECHNICAL DETAILS OF INTERNSHIP DONE

2.1 Block Diagram



(2)

2.2 Flow Chart



(3)

2.3 Methodology followed

To complete this project, you must require the following software's, concepts, and packages

1.Anaconda (IDLE / Spyder / PyCharm)(Python 3.7)

2.Python Packages

3.Tensorflow- This package is used as backend support to Keras

4.Keras-This package is used for building Neural Network layers

5.OpenCV-This package is used for image processing

6.Flask- To build a web application

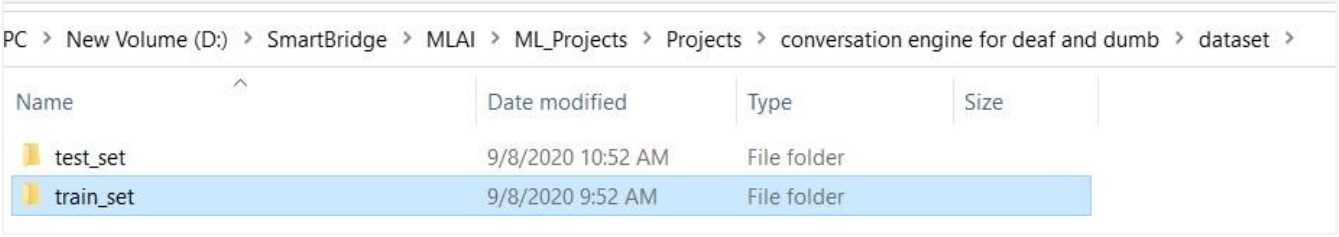
➢ Data Collection

In this, we will be collecting data for building our project. We will be creating two folders one for training and the other for testing. Images present in the training folder will be used for building the model and the testing images will be used for validating our model.

Create Train and Test Folders

S**tep1:** Create Train and Test folders with each folder having folders with images of different hand signs. A minimum of 100 images needs to be present in each category folder to get the maximum no of features.

(1.1)



➢ Image Preprocessing

In this, we will pre-process the images which will be used for building the model. Image pre-processing includes zooming, shearing, flipping to increase the robustness of the model after it is built. We will be using the Keras package for pre-processing images.

Import ImageDataGenerator Library and Configure it

Import ImageDataGenerator and create an instance for which include shearing, rescale, zooming, etc to make the model robust with different types of images.

(2.1)

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

Apply ImageDataGenerator functionality to Train and Test set

Specify the path of both the folders in the flow_from_directory method.Flow from directory loads the images from a given directory and can bring all the images to the target size. We will be loading all the images of the train and test using the flow from directory method.

(2.2)

```
x_train = train_datagen.flow_from_directory('dataset/training_set', target_size=(64,64), batch_size=300,
                                            class_mode='categorical',  color_mode = "grayscale")
```

```
x_train = train_datagen.flow_from_directory('dataset/training_set', target_size=(64,64), batch_size=300,
                                            class_mode='categorical',  color_mode = "grayscale")
```

➢ Model Building

In this milestone, we start building our model by:
1. Initializing the mode
2. Adding Convolution layers
3. Adding Pooling layers
4. Flatten layer
5. Full connection layers which include hidden layers

At last, we compile the model with layers we added to complete the neural network structure.

Import the required model building libraries.

Import the libraries that are required to initialize the neural network layer, create and add different layers to the neural network model.

(3.1)

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

Initialize the model

Initialize the neural network layer by creating a reference/object to the Sequential class.

(3.2)

```
model=Sequential()
```

Add the convolution layer

The first layer of the neural network model, the convolution layer will be added. To create a convolution layer, Convolution2D class is used. It takes the number of feature detectors, feature detector size, expected input shape of the image, activation function as arguments. This layer applies feature detectors on the input image and returns a feature map (features from the image).

(3.3)

```
model.add(Convolution2D(32, (3,3), input_shape=(64,64,1), activation = 'relu'))
#no. of feature detectors, size of featuredetector, image size, activation function
```

Add the pooling layer

After the convolution layer, usually, the pooling layer is added. Max pooling layer can be added using MaxPooling2D class. It takes the pool size as a parameter. The efficient size of the pooling matrix is (2,2). It returns the pooled feature maps. (Note: Any number of convolution layers, pooling and dropout layers can be added).

(3.4)

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

Add the flatten layer

The flatten layer is used to convert the n-dimensional array to a 1-dimensional array. This 1D array will be given as input to ANN layers.

(4.1)

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

Adding the dense layers

Three dense layers are added which usually takes the number of units/neurons. Specifying the activation function, kind of weight initialization is optional.

(4.2)

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

```
model.add(Dense(units=9, activation='softmax'))
```

Compile the model

After adding all the required layers, the model is to be compiled. For this step, loss function, optimizer, and metrics for evaluation can be passed as arguments.

(5.1)

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

Fit and save the model

Fit the neural network model with the train and test set, number of epochs, and validation steps.The weights are to be saved for future use. The weights are saved in as .h5 file using save().

(5.2)

```
model.fit_generator(x_train, steps_per_epoch=24, epochs=10, validation_data=x_test, validation_steps=40)

#steps_per_epoch = no. of train images//batch size
```

```
model.save('aslpng1.h5')
```

➢ Test the model

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

Import the packages and load the saved model

As a first step to start prediction we import packages that are used for loading the model and used to expand the dimension of the image. We use the Keras package to load the model which was saved when we built the model.

(6.1)

```
from keras.models import load_model
import numpy as np
import cv2
```

```
model=load_model('aslpng1.h5')
```

Load the test image, pre-process it and predict

Pre-processing the image includes converting the image to the array and resizing according to the model. Give the pre-processed image to the model to know to which class your model belongs to.

(7.1)

```
from skimage.transform import resize
def detect(frame):
        img = resize(frame,(64,64,1))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img = img/255.0
        prediction = model.predict(img)
        print(prediction)
        prediction = model.predict_classes(img)
        print(prediction)
```

```
frame=cv2.imread(r"G:\Gayatri Files\Smartbridge\Nidhi\Comversation Engine for Deaf and Dumb\Dataset\test_set\G\1.png")
data = detect(frame)
```

```
[[6.0201724e-13 7.6744452e-18 1.7007801e-10 7.7269103e-14 2.9694178e-15
  8.9405344e-16 9.9999082e-01 9.1214142e-06 3.0555274e-17]]
[6]
```

➢ Application Building

Now we will be building a Flask application that is used for building our UI which in backend can be interfaced to the model to get predictions. Flask application requires an HTML page for Frontend and a Python file for the backend which takes care of the interface with the model.

Build a flask application

Step 1: Load the required packages.

(8.1)

```
import numpy as np
import cv2
import os
from keras.models import load_model
from flask import Flask, render_template, Response
import tensorflow as tf
from gtts import gTTS #to convert text to speech
global graph
global writer
from skimage.transform import resize
```

Step 2: Initialize graph, load the model, initialize the flask app and load the video

Graph element is required to work with tensorflow. So, graph element is created explicitly.

(8.2)

```
graph = tf.get_default_graph()
writer = None

model = load_model('aslpng1.h5')

vals = ['A', 'B','C','D','E','F','G','H','I']

app = Flask(__name__)

print("[INFO] accessing video stream...")
vs = cv2.VideoCapture(0) #triggers the local camera

pred=""
```

Step 3: Configure the home page.

(9.1)

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

```

Each frame is taken from the camera and processed and sent to the model for prediction. As discussed image undergoes different processing steps to meet model requirements to get predictions.

```
#preprocessing the frame captured from camera
def detect(frame):
        img = resize(frame,(64,64,1))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img = img/255.0
        with graph.as_default():
                prediction = model.predict_classes(img)
        print(prediction)
        pred=vals[prediction[0]]
        print(pred)
        return pred

```

This below in the snippet for calling video feed from the HTML page.

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

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)

```

Build the HTML page

Build an HTML page to display the processed video on the screen, so that the person can show signs which can be detected. Run the application by going to the location of your program and run python webstreaming.py

When the python file is executed the localhost is activated on 5000 port and can be accessed through it.

(10.1)

```
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
[INFO] accessing video stream...
 * Debugger is active!
 * Debugger PIN: 257-358-499
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

➢ Output

Step 2: Open the browser and navigate to localhost:8000 to check your application.

(4)

➢ Train CNN Model on IBM

You can also train your Image classification Models on IBM Cloud using IBM Watson Studio Service.
This milestone lets you

- Train your model on IBM
- Store your Model on IBM
- Download the Stored model to the Local system

2.4 Skills acquired during Internship

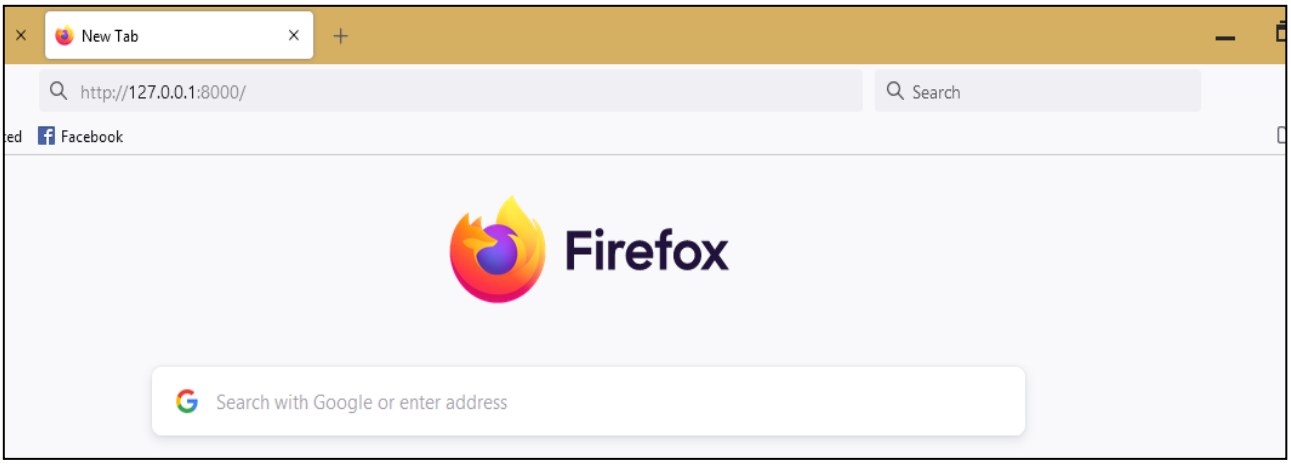
The Guided Projects were not only based on solving real world problems but were also really informative. As a part of internship I have learnt:

- the basics of ML, Deep Learning and AI
- Neural Networks understanding and programming
- Advanced programming frameworks
- Machine Learning Algorithms
- Solving real AI problems through programming.
- A project named Conversation Engine for deaf and dumb where training, testing the data ,building a flask application and building a HTML page.
- Deployment in IBM Cloud.

CHAPTER 3. RESULTS

The result is shown in localhost: 8000 where alphabet is shown and is spoken once a particular sign is shown.

(5)



The sign shown is C so clearly in the picture it is shown that “It indicates C” and a sound is also heard which says-“It indicates C”. All these signs are already present in the data set. In the same way if any other sign is shown then it indicates that particular sign.

(6)

CHAPTER 4. CONCLUSION

We finally came up with a project where a system is developed that converts thesign language into a human hearing voice in the desired language to convey a message to normal people, as well as converting speech into understandable signlanguage for the deaf and dumb.

We follow through the flowchart above where training, testing of the model isdone and UI Part of the project includes HTML File and Python file.

Advantages:

- It provides an environment for the user to send message through a sign language keyboard. It recognize the hand gestures of sign language through image processing.
- Other feature converts the speech spoken into text .

Disadvantages:

It has limitations sych as it cannot convert text to speech.

Applications:

- to cater the needs of children who are non-verbal or are having speechproblems.
- The app comes to the aid of mute, deaf and other non-verbal users when having a conversation

CHAPTER 5. FUTURE SCOPE

Lot of developments can be made in the application such as adding feature like converting text to speech.It is a very handy application for all the deaf and dumb People as it can really help them.

REFERENCES:

- **https://smartinternz.com/Student/guided_project_info/2252#**
- **https://azure.microsoft.com**
- **https://www.mygreatlearning.com/blog/what-is-artificial-intelligence/**