ABSTRACT

A Virtual Assistant is a software or a device that constantly interacts with the user by responding to the queries asked by a user as input. Virtual Assistant usually takes input as text and output as text, input as speech and output as speech, input as text and output as speech, and vice versa.

In this project, I tried to implement a virtual assistant which accepts both text and speech as input and provides output in text or speech based on user requirement. I have used Natural Language Processing (NLP) for data processing and Convolutional Neural Networks (CNN) for fetching the output from the provided dataset. I have used JSON (JavaScript Object Notation) dataset, which is faster, easy to use, has better schema support. This model is trained with the same dataset.

The trained model is integrated with the web page using Flask. Web page is built using HTML, CSS, JavaScript. After successful integration user can hear the response to his query from this Assistant.

Keywords: Natural Language Processing (NLP), Convolutional Neural Networks (CNN), JSON (JavaScript Object Notation), Flask.

INDEX

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION:

In this digital world, everyone has access to digital devices, and all are using the internet widely. Many Organizations are having online platforms for different purposes mainly to connect with their customers, but all the users are not familiar with usage of digital platforms. Someone should answer their queries now, A Virtual Assistant come into the picture.

Virtual Assistant is a software or a product that constantly interacts with users by responding to the queries of the user. It can be integrated with any website and can effectively use to solve queries of users. They help organizations in enhancing user experience and reduces the amount spent on customer support executive.

It takes query as an input in text or voice and responds in text or voice based on user requirement. This is powered with deep learning algorithm, CNN so that it would respond faster than usual.

Built CNN model is used for Travel Planner when user enters the name of the place listed in dataset. Dataset contains 85 most visited places in Cleveland, when user enters the place which is not in the list then model asks to user enter the name in the dataset.

1.2 PROJECT OBJECTIVE:

My objective is to implement virtual assistant using deep learning algorithm, Convolutional Neural Networks (CNN) which enables model to respond the user. Primary objective is enabling user input as text and speech. And displaying output on the screen and speaking out the text displayed as output. Later integrating this model with an web application which acts as a user interface.

CHAPTER 2: VIRTUAL ASSISTANT

2.1 INTRODUCTION:

A Virtual Assistant is a device or software which provides support services to the organization in solving user queries. They may also assist with any other elements required by the business.

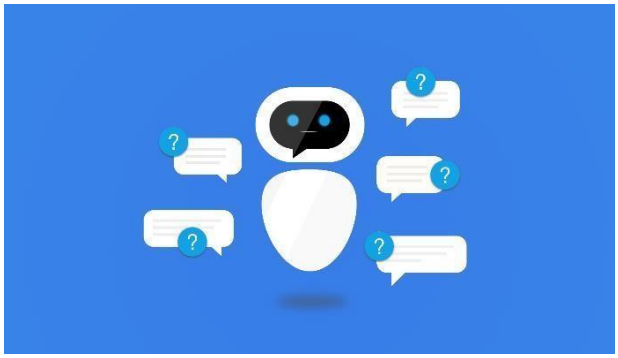


Fig 2.1 Virtual Assistant

A Virtual Assistant can do anything that support staff might do. There are some limitations, but technology is increasing, and offering ways to work around those limitations. For example, they may not be able make tea, lunch, coffee for you but they can place an order for you through a food delivery service.

Virtual Assistant is not limited to clerical work. They can also be used for assistance in marketing, social media, web design, online teaching, virtual meetings, and many other domains.

2.2 TYPES OF VIRTUAL ASSISTANTS:

There are many types of virtual assistants some of them are social media Virtual Assistant, Real Estate VA, Virtual research Assistant, Virtual Administrative Assistant, E-Commerce Virtual Assistant, Data Entry VA, Virtual Bookkeeping Assistant, Virtual Marketing Assistants.

2.3 FEATURES:

Regardless, where we are using Virtual Assistant. It should possess following features:

1) Robustness.
2) Flexibility.
3) Resource efficient.

Travel:

VA can be used in travelling, for arranging flights and hotels. VA can perform tasks like Research flights, Research hotels, Book flights and hotels, Research transportation options, Book transportation, arrange for events, Suspend newspaper or mail.

Virtual Assistant for Travel.

The INSIGHTS that I had learned from this Project was as follows and how there are used in the implementations towards the project:

1.NATURAL LANGUAGE PROCESSING

2.CONVOLUTIONAL NEURAL NETWORKS

1. The Natural Language Processing is acronym for NLP. It refers to AI method to build communication with a system using natural language. Its Overall goal is to turn natural language into data analysis of NLP.

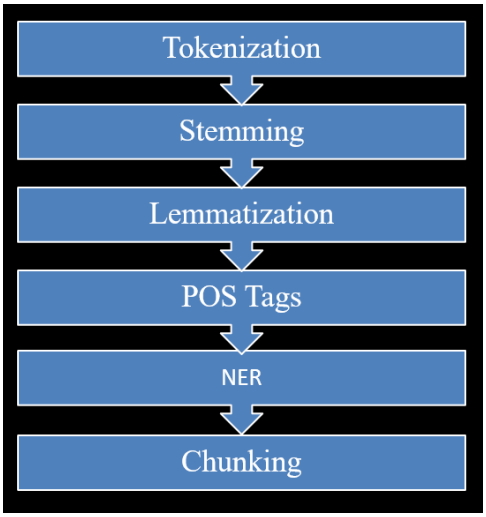
In the further NLP was divided into two components they are Natural Language Understanding (NLU) and Natural Language Generation (NLG).

Natural Language Understanding:

The NLU maps the given input into natural language to useful representation and analyzing those aspects of the language. NLU is more complex than NLG.

Natural Language Generation:

NLG generates the phrases, sentences in the form of natural language from internal system. Basically, there are six steps that involved in natural language processing.



NLTK:

NLTK stands for “Natural Language Tool Kit”. It is a library provided by python to perform Natural Language Processing. NLTK is widely used in implementing Natural Language Processing.

CONVOLUTIONAL NEURAL NETWORKS

2. CNN is a Deep learning algorithm which is usually applied to analyse visual images. It uses a technique called Convolution. CNN automatically detects the important features with any human interference. It learns the key features for each class by itself.

CNN is combination of convolutional layers and neural networks. CNN consists of layers. It contains layers that are convolutional layers, full connected layers, pooling layers, dense layer, hidden layer. CNN reduces the data size of the input by using filters resulting in feature maps.

4.1 CONVOLUTION:

Convolutional Layers are the major building blocks used in CNN. A convolution is the application of filter to input those results into activation. Repeated application of the filterto an input results in feature map. CNN has ability to a learn large number of filters in specific to dataset used for training.

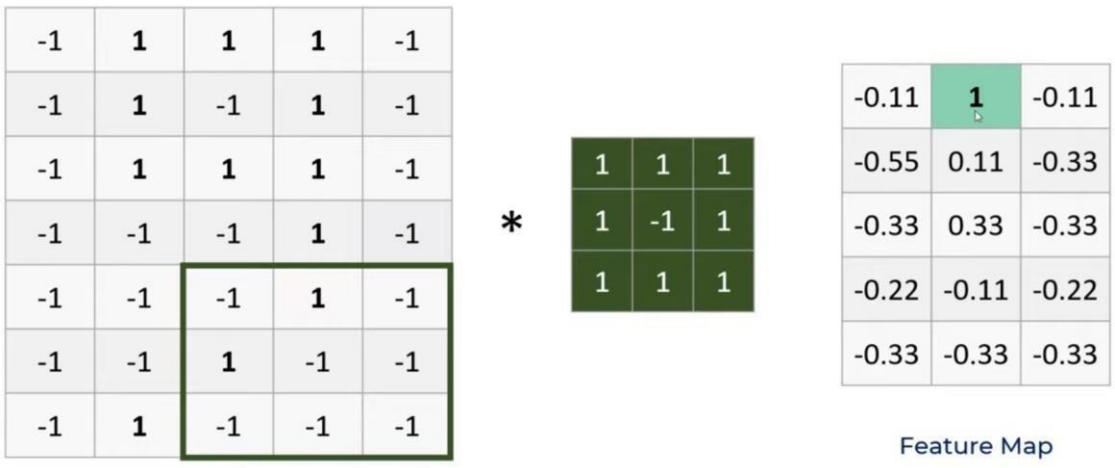


Fig 4.1 Applying filter to input.

In the above example filter applied is loopy pattern detector which results in feature map whose size much smaller than actual input. This will make computation faster.

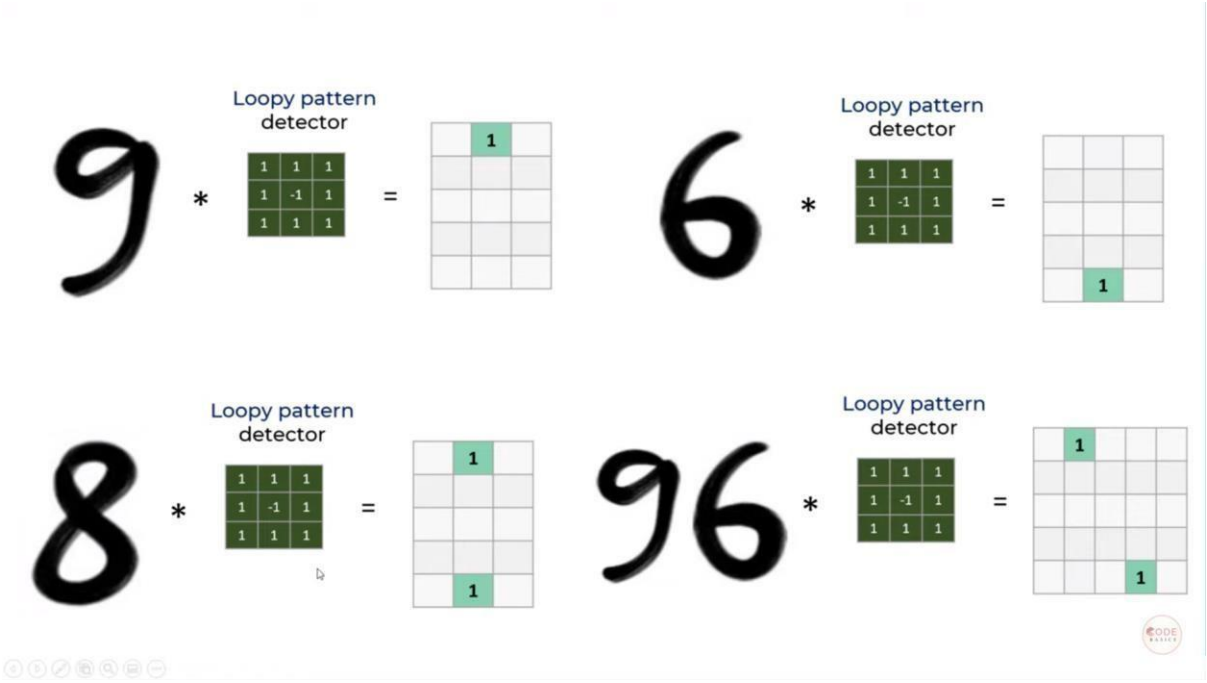


Fig 4.2 Applying Loopy pattern detector to different inputs.

When convolution applied in face recognition, animal recognition. Then we need to create feature map for different parts like nose, eyes, body, legs, and other features.

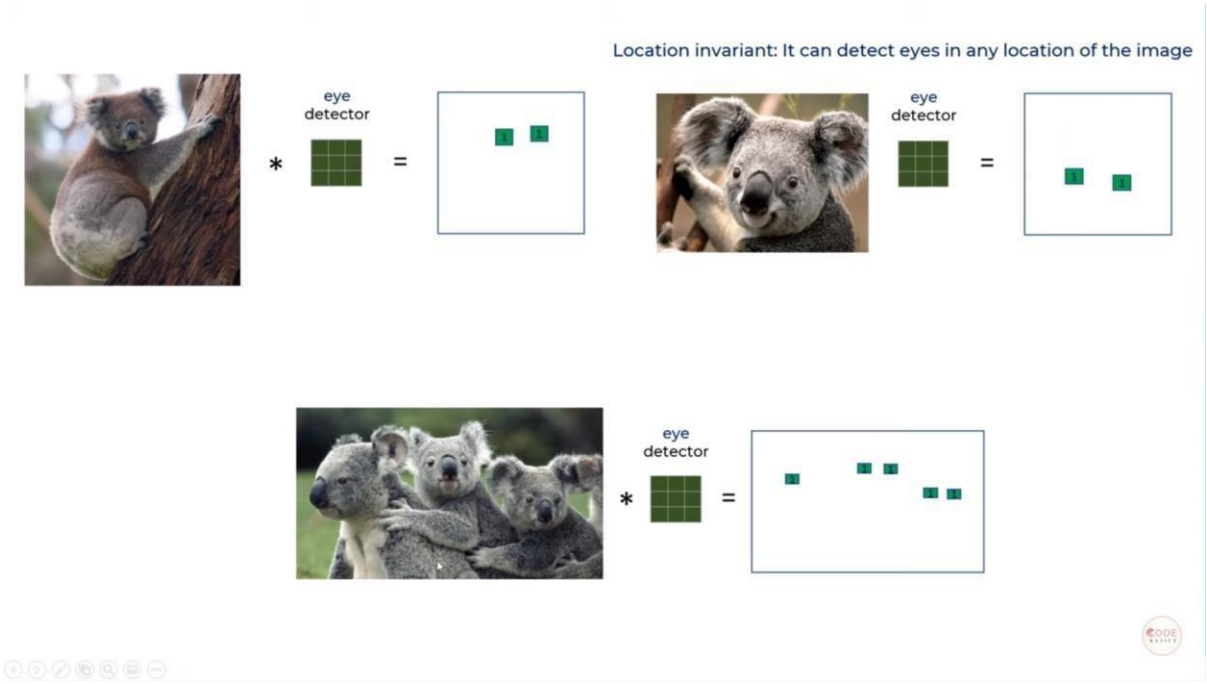


Fig 4.3 Applying eye detector to different inputs.

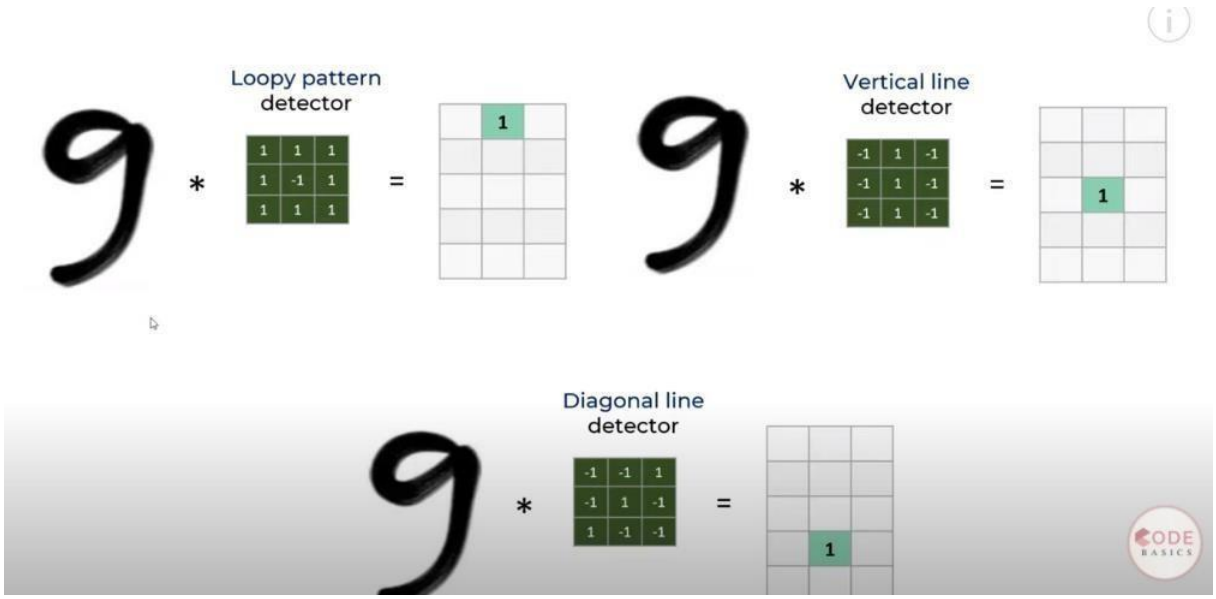


Fig 4.4 Applying different detectors to same input.

In the above picture, applying different detectors/filters same input will result in different feature maps. Each feature map depicts different feature. In the above picture Loopy pattern detector, vertical line detector and diagonal line detector results in three feature maps, each feature map describes respective features.

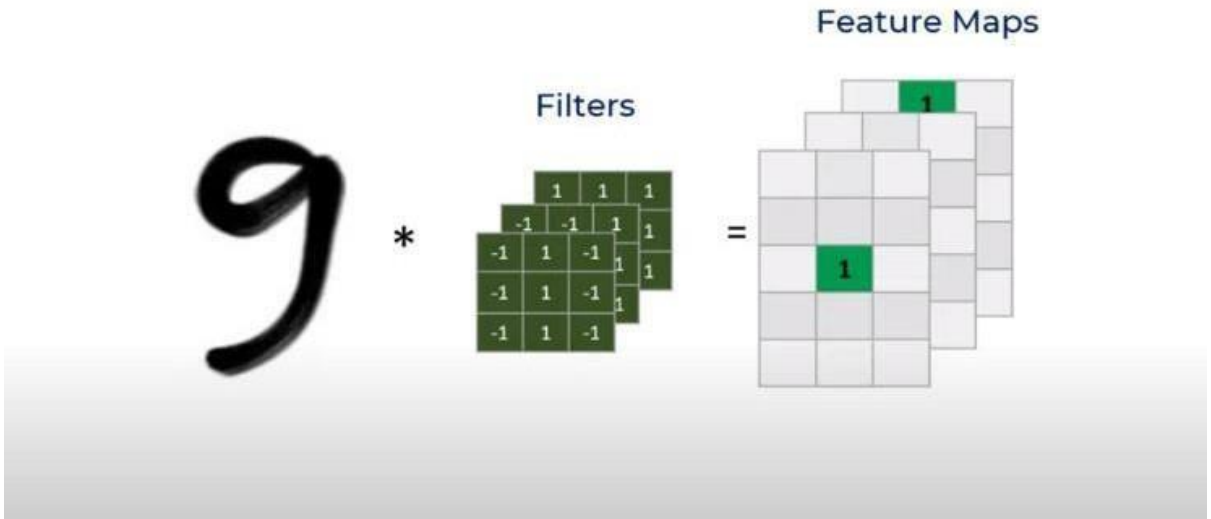


Fig 4.5 Stacking the feature maps of input.

In the above picture Different feature maps are generated by applying different filters/detector are stacked together in the process of convolution.

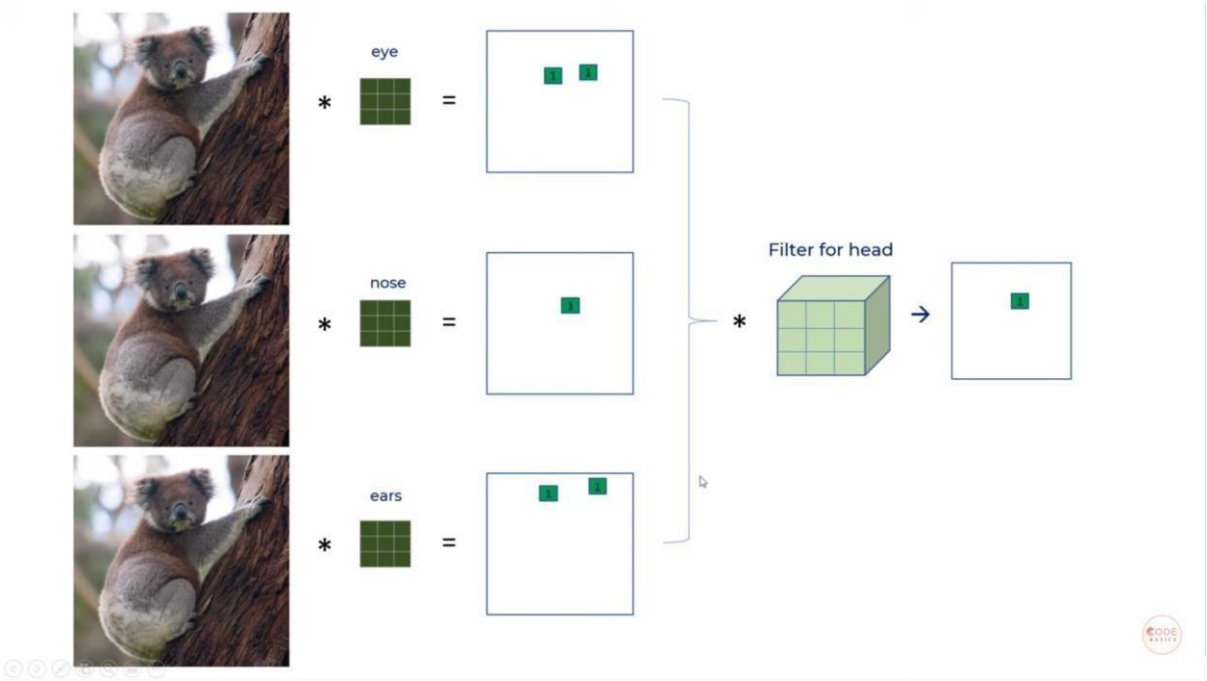


Fig 4.6 Applying 3-dimensional filter to input.

Filters can also be three dimensional. In the above picture three-dimensional filter used is to detect head. One dimension detects nose, second one detects nose, and third one detects ears. Three results combining detects the face of the Koala.

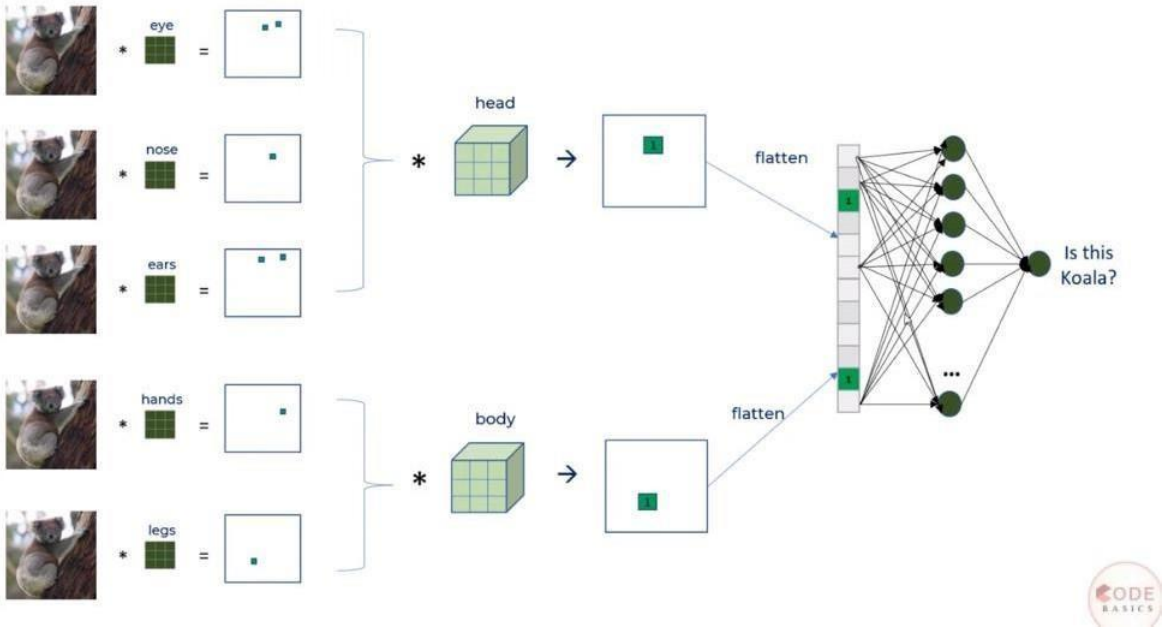


Fig 4.7 Fully Connected Layer of the given input.

Above picture is of Fully connected layer which has both feature extraction and classification. Input is sent through different filters generating feature maps. All the feature maps are stacked. This Feature Extraction will help in detecting the Koala features in any position. Koala may be sleeping, dancing, hanging. Classification is done by simple ANN. This Fully Connected Layer will be able to detect features.

4.1.1 BENEFITS OF CONVOLUTION:

- Connections Sparsity reduces over fitting.
- Conv + pooling gives the location in variant feature detection.
- Parameter sharing.

4.2 RECTIFIED LINEAR UNIT:

Rectified Linear Unit is acronym for Re Lu. It helps with making the model non- linear. Re Lu is a linear function that will output the input if it is positive, if the input is negative then output will be zero.

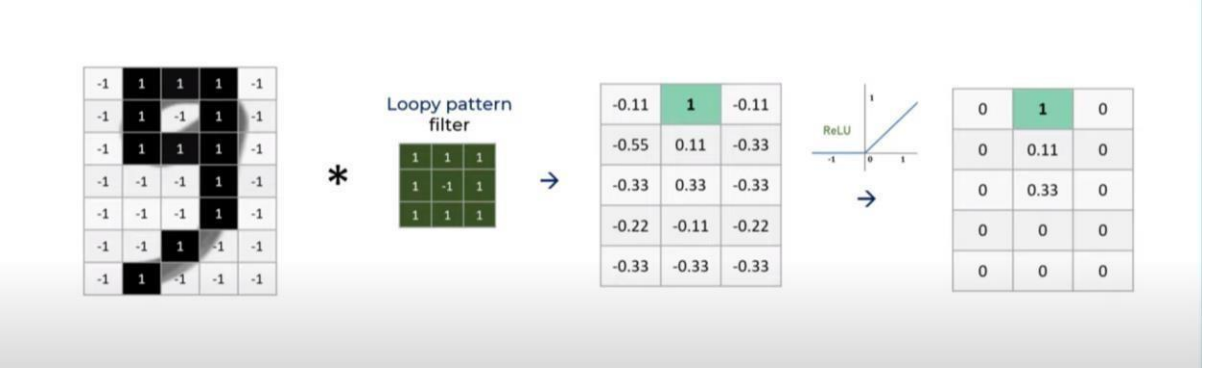


Fig 4.8 Applying Re Lu to input.

Feature map is generated by applying filter/Loopy pattern detector to input. Re Lu is applied to feature map where input is positive output will be same as input. If input is less than zero, then output will be zero.

4.2.1 BENEFITS OF USING Re Lu:

- Introducing nonlinearity.
- Speeds up training.
- Faster to compute.

4.3 POOLING LAYER:

Pooling layers provides down sampling the feature maps by summarizing the presence of features in patches of the feature map. There are two common pooling methods. One is Average pooling. Second is Max pooling. This pooling summarizes the average presence of a feature and the most activated presence of a feature, respectively.

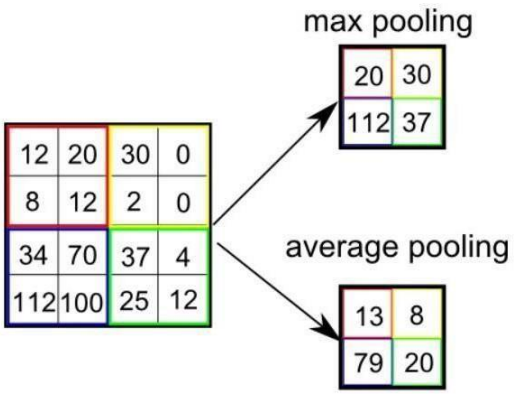


Fig 4.9 Applying Pooling Layer.

Theoretically, Pooling Layer reduces the size of the feature map generated from given input. Pooling process include selecting the window of size n by n and choosing the number one number from each window resulting feature map whose size much smaller than actual feature map.

4.3.1 TYPES OF POOLING LAYER:

For selecting the number from each window there are two methods, they are:

1. Max Pooling.
2. Average Pooling.

MAX POOLING:

Max pooling is a process used to reduce the dimensions of the feature map. In max pooling from each window maximum number is selected and new feature map is generated with selected numbers. Max pooling is most widely used.

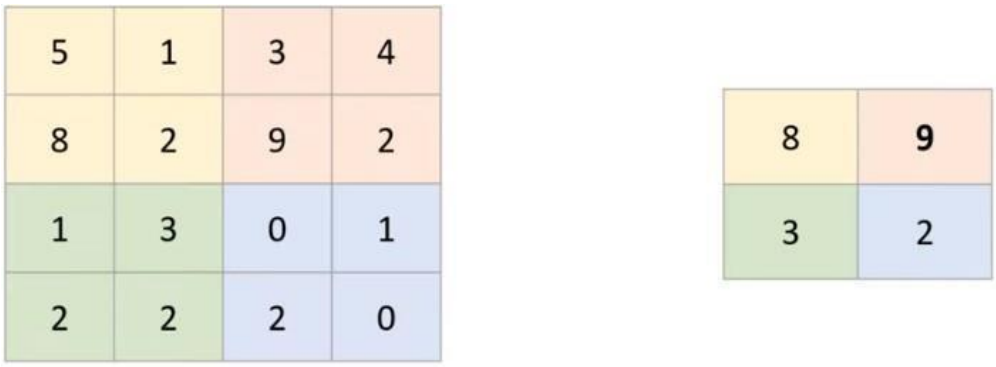


Fig 4.10 Max Pooling 2 by 2 filter stride = 2.

In the above example, the feature map of size 4 by 4 is reduced using max pooling layer. Here window size taken is 2 by 2 and stride is 2, stride can be of any no and window size can be taken based our requirement. From 2 by 2 window largest number among windowis selected. In the above example after max pooling 4 by 4 feature map is reduced to 2 by 2 feature map.

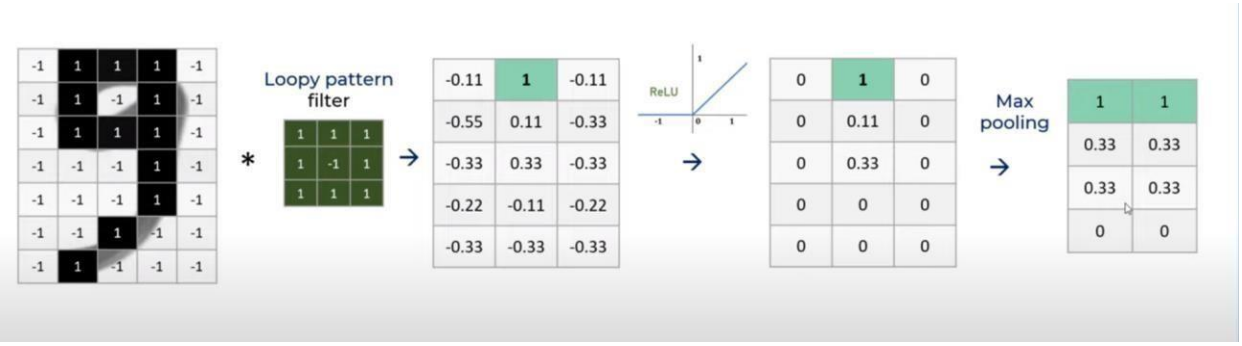


Fig 4.11 Applying Max Pooling on input.

AVERAGE POOLING:

Average pooling is another process used to reduce the dimensions of the feature map. In average pooling from each window average of numbers calculated and selected. New feature map is generated with selected numbers.

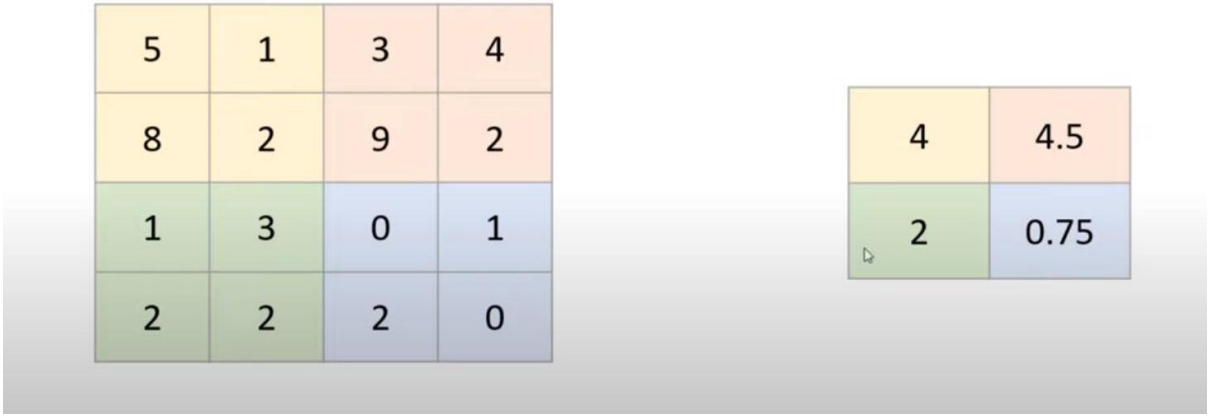


Fig 4.12 Average Pooling on Feature map.

In the above example, the feature map of size 4 by 4 is reduced using average pooling. Here window size taken is 2 by 2 and stride is 2, stride can be of any no and window size can be taken based our requirement. From 2 by 2 window average of all the numbers among window is selected. In the above example after average pooling 4 by 4 feature map is reduced to 2 by 2 feature map.

4.3.2 BENEFITS IN USING POOLING:

- Reduces the dimensions and computation.
- Reduce overfitting as there are less parameters.
- Model is tolerant towards variations distortions.

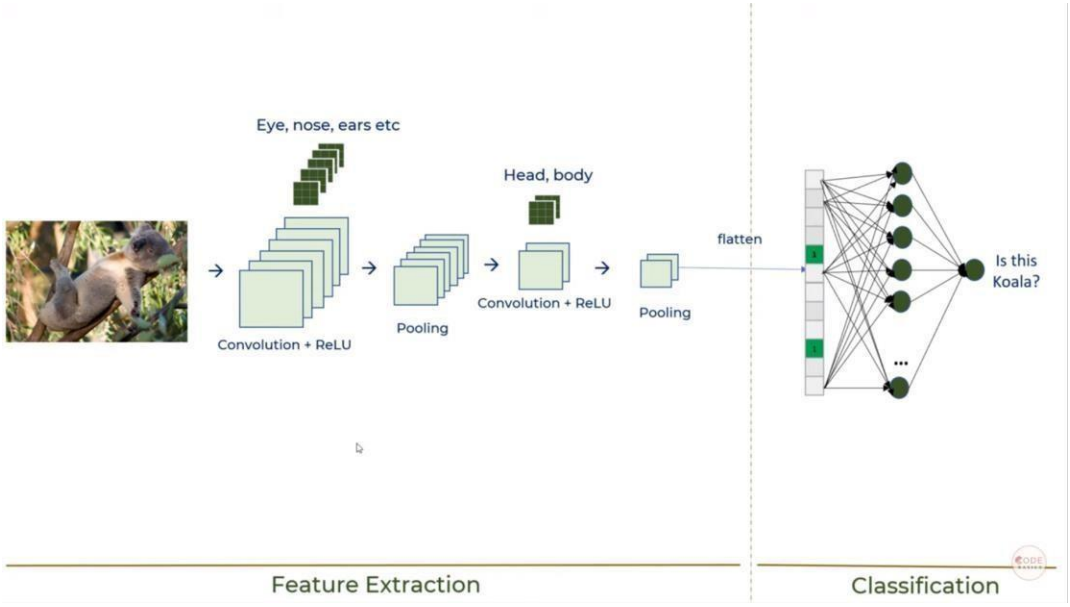


Fig 4.13 Complete CNN for an input.

Above image is complete CNN for an input. Typically, CNN is comprised of Conv + ReLu and Pooling. There can be many layers of Conv + ReLu and Pooling based on input. For above case First layer detects eye, nose, ears of the input and second convolution layer detects Head, body. Later max pooling reduces the dimensions of the generated feature map from input. Classification is done using simple CNN.

4.4 ADVANTAGES OF CNN:

- Local receptive fields.
- Sparse Connectivity.
- Parameter sharing.
- Equivariant representation and translation-invariant.
- Faster Computation.

CHAPTER 5: TOOLS USED

5.1 VISUAL STUDIO CODE:

The Visual studio code is a text editor for writing programs in different languages. It was developed and maintained by Microsoft. Apart from being only a text editor it also has many features like debugging tool, source control, testing, and an integrated terminal. With these features it becomes Integrated Development Environment (IDE). Apart from default features more functionalities can be added to visual studio code by searching for the required tool and adding them in the extensions section.

Fig 5.1 Visual Studio Code

5.2 PYTHON IDLE:

Python IDLE (Integrated Development and Learning Environment) is an open-source tool which is IDE in which python files can be run. Using IDLE we can create, modify, execute python files.

CHAPTER 6: DATA SET

6.1 ABOUT THE DATA SET:

The dataset used in this project is JSON (JavaScript Object Notation). JSON is a light weighted text-data. This is interchangeable format. It is used to store information in an organized manner. It is used to transmit the data from server to client and vice versa. JSON is Scalable. It is lightweight. It is easy to read and write. It is a text-based, human-readable data exchange format. JSON can be used for both Relational and Non-Relational databases. JSON uses less data overall, so you reduce the cost and increase the parsing speed.

The dataset I used for project consists of records of 85 most visited places all over Cleveland Each Record consists of Intents, Tag name, patterns, and responses and for each record data/context is stored in string format.

6.1.1 SAMPLE RECORD IN DATA SET:

```
Intents {
        {
                "tag": "tag name",
                "patterns": ["pattern name"],
                "responses": ["Responses to Query."]
        }
}
```

CHAPTER 7: METHODOLGY AND WORKFLOW

7.1 BLOCK DIAGRAM:

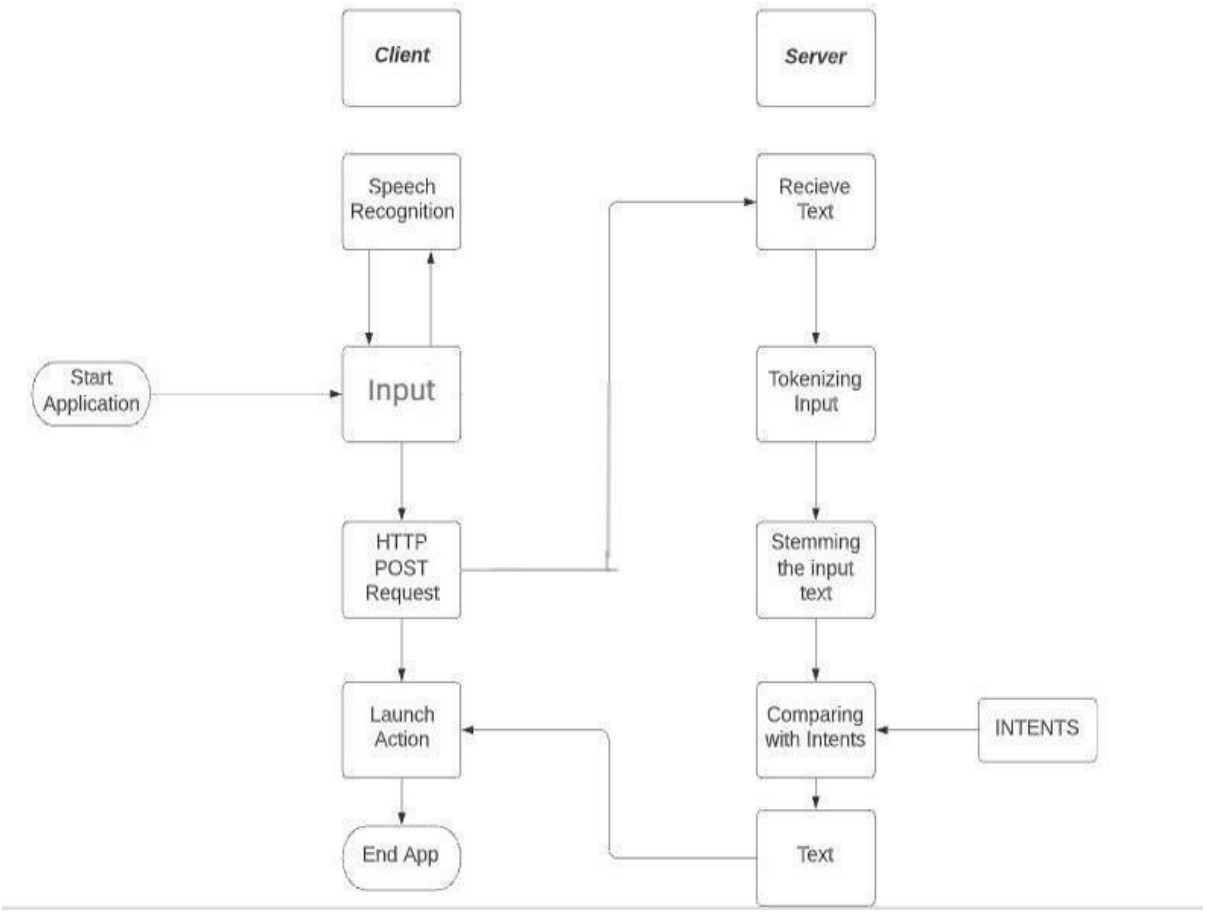


Fig 7.1 Block Diagram.

7.2 DESCRIPTION OF BLOCK DIAGRAM:

In this project both client side and server side are involved. Client side serves as user interface where user can give the input. Server side is where model is present. Model is responsible for processing input and extracting output.

Now let us discuss briefly about each step present in the above block diagram.

Input:

User gives input to the model using web page. User can enter the input in the field.

Speech Recognition:

User can enter input in text or voice. If user enter voice as input. Speech synthesis library detects the voice and converts into text for processing.

HTTP POST Request:

HTTP stands for “Hypertext transfer protocol”. After Speech input is converted to text. Client raises a request to server using HTTP POST Request.

Receive text:

Server receives text from client. Server consists of model which processes the input and sends the output to Client. To get output there are some steps to be followed.

Tokenizing Input:

Model uses NLP for tokenizing input. It is the process which converts sentences into tokens. For example: Consider the sentence “The Travel Planner can used for joyful trips”, In above sentence tokens are ‘The’, ‘Travel’, ‘Planner’, ‘can’, ‘used’, ‘for’, ‘joyful’, ‘trips’.

Stemming Input:

Model uses NLP for stemming the input. It is the process which converts words into base or root form. For example: Consider the words, ‘consulting’, ‘consultant’, ‘consultation’, ‘consultants’ and root word will be ‘Consult’.

Comparing the intents:

After tokenization and stemming the input next step is comparing the input with the intents present in the dataset and CNN algorithm tracks for the matching result.

Text:

Once matching result fetched, now model will send the output to Client which is sending Response to the HTTP POST Request.

Launch Action:

Client receives output from Server. Now client which is web page displays the output to the user. If user expects voice output speech recognition will read out the text displayed in the output field.

7.3 WORKFLOW:

7.3.1 IMPLEMENTING REQUIRED LIBRARIES:

To build the model we need some of the predefined libraries which offers some operations to do specific task.

```
# importing required libraries
import nltk
from nltk.stem.lancaster import LancasterStemmer
import numpy as np
import tflearn
import tensorflow as tf
import random
import json
from flask import Flask,jsonify, request, render_template,json
```

Fig 7.2 Implementing required libraries.

In this project I used libraries like 'nltk' stands for "natural language tool kit" it supports natural language processing. 'numpy' It is a library used to work with arrays. 'tflearn' it is used to run 'tensorflow'.

7.3.2 Loading the dataset of JSON format:

Dataset used in this project is JSON (JavaScript Object Notation), which is easy to handle and has better schema support. In this step dataset is uploaded to model.

```
# loading the dataset of JSON format
with open("intents.json") as jd:
    intents = json.load(jd)
```

Fig 7.3 Loading the dataset of JSON format.

7.3.3 Building the Model:

To build model I am going to use NLP and CNN algorithm. NLP takes the input and processing is done which includes steps like Tokenization, Stemming, tagging. The model is trained. Model fetches the output from given dataset.

```
# shuffle our features and turn into np.array
random.shuffle(training)
training = np.array(training)
# creating training list
train_x = list(training[:,0])
train_y = list(training[:,1])
# Building convolutional neural network
net = tflearn.input_data(shape=[None, len(train_x[0])])
net = tflearn.fully_connected(net, len(train_x))
net = tflearn.fully_connected(net, len(train_y[0]), activation='softmax')
net = tflearn.regression(net)
# Define model and setup tensorboard
model = tflearn.DNN(net, tensorboard_dir='tflearn_logs')
model.fit(train_x, train_y, n_epoch=1000, batch_size=8, show_metric=True)
model.save('model.tflearn')
```

Fig 7.4 Building the model.

7.3.4 Generating the HTTP POST Request:

This happens at client side when user enters the input. Client generates HTTP POST Request. When client generates the request. Server receives the input and later process takes place resulting in output.

```
$.ajax({
    type: "POST",
    url:server+appdir,
    data: JSON.stringify(op_nu),
    dataType: 'json'
})
```

Fig 7.5 Generating HTTP POST Request.

7.3.5 Sending the response to HTTP POST Request:

After the input is processed by the model and output is generated then server sends the output as response to HTTP POST Request.

```
@app.route('/op', methods=['POST'])
def sum_num():
    rf=request.form

    for key in rf.keys():
        data=key

    data_dic=json.loads(data)
    input_sent=response(data_dic['str'])

    resp_dic={'str':str(input_sent)}
    resp = jsonify(resp_dic)
    resp.headers['Access-Control-Allow-Origin']='*'
    return resp
```

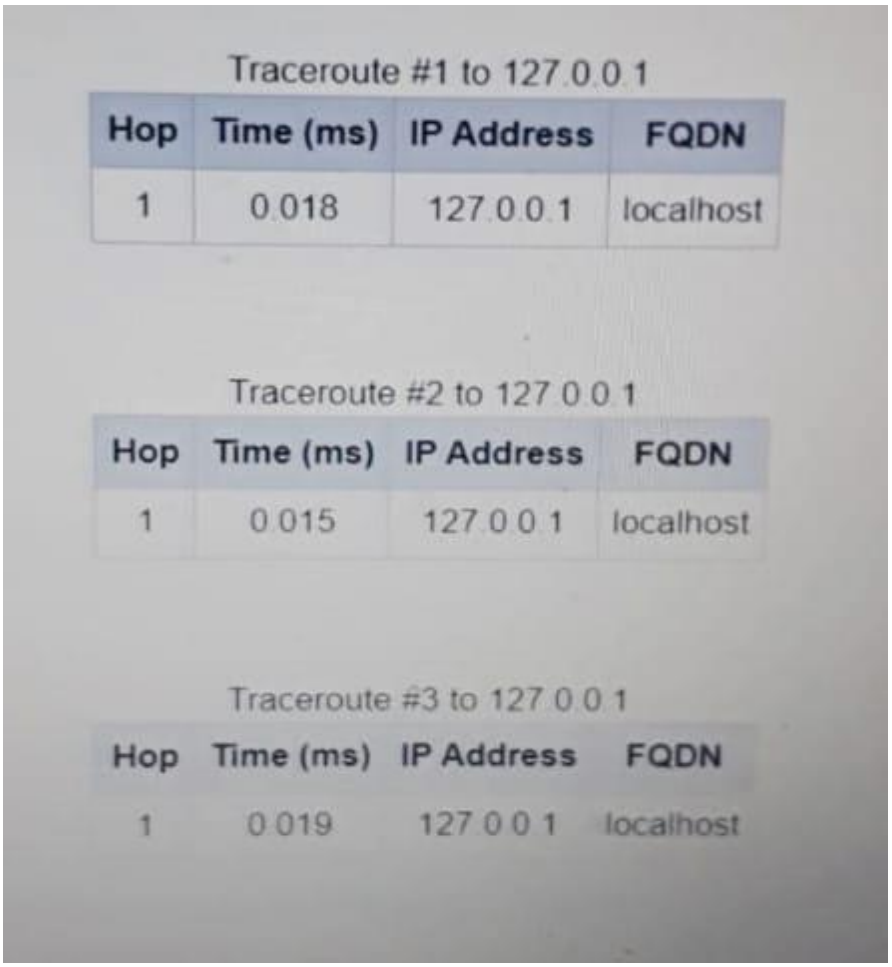
Fig 7.6 Sending the response to HTTP POST Request.

CHAPTER 8: RESULTS

8.1 RESULTS:

Input is given by the user in speech or text, it is processed, and output is displayed on screen. When the input is present in dataset output is displayed. If place name is not listed in dataset application displays “none”. I have calculated the Response Time of my Application.

Response time is the total time taken for an application to respond to request for a service. It is algebraic sum of the wait time and the service time. To measure the response time of my application I used webtestpage.com which reads the time at which user sends HTTP request and time at which output is generated by model. I have calculated response time for three random inputs and calculated the average of those results.


Traceroute #1 to 127.0.0.1

Hop	Time (ms)	IP Address	FQDN
1	0.018	127.0.0.1	localhost

Traceroute #2 to 127.0.0.1

Hop	Time (ms)	IP Address	FQDN
1	0.015	127.0.0.1	localhost

Traceroute #3 to 127.0.0.1

Hop	Time (ms)	IP Address	FQDN
1	0.019	127.0.0.1	localhost

Fig 8.1 Snapshot of results of response time for random inputs.

Input to Output	Response Time 1 (ms)	Response Time 2 (ms)	Response Time 3 (ms)
Text to Text	**0.018**	**0.015**	**0.019**
Speech to Speech	**0.018**	**0.015**	**0.019**

Table 8.1 Response time for random inputs.

Response Time: 0.01733 ms

8.2 SNAPSHOTS OF THE RESULTS:

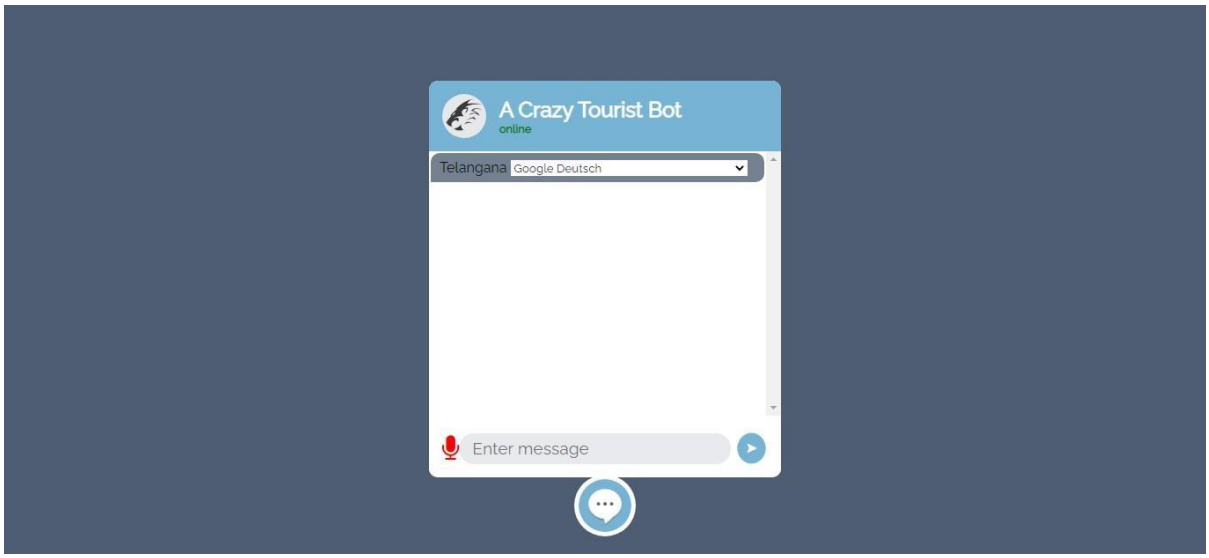
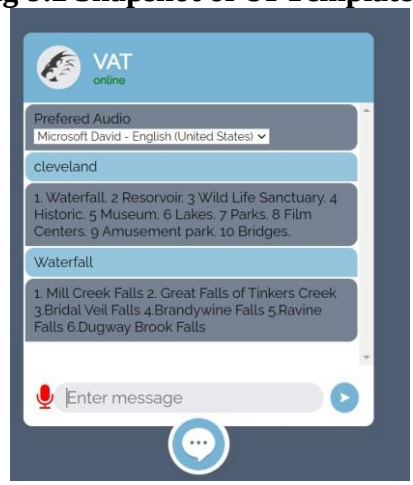


Fig 9.2 Snapshot of UI Template.



CHAPTER 9: CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION:

Virtual Assistant helps in responding to the queries of the users without any human interference. I built the assistant using deep learning algorithm, Convolutional Neural Networks. This CNN model is trained with Json data set consisting of records of 85 most visited places in Cleveland. When user enters the input, which is not listed in dataset, My Application displays “none”. User should enter input listed in dataset.

This Trained Algorithm is integrated with web page using docker and this web page serves as the user interface for end user and finally my project reduces the human effort in solving queries of users manually and makes profit to organization by reducing their expenses on customer executive service.

9.2 FUTURE SCOPE:

Virtual Assistant can be further powered by Artificial Intelligence which will increase the speed of response and reduces the response time. Using a non-relational database, we can also add images and videos that can displayed as output with context to enhance the user experience. Dataset can be upgraded with more no of records it enhances the performance.

We can use API (Application Programmable Interface) of google maps and integrate maps with my model so that user can directly use navigation on my web page itself.

This Project can also be used for medical, educational, entertainment, ecommerce, and education purposes.

WEBSITE LINKS:

1) https://code.visualstudio.com/docs
2) Welcome to Colaboratory - Colaboratory (google.com)
3) 3.9.5 Documentation (python.org)
4) Welcome to Flask — Flask Documentation (2.0.x) (palletsprojects.com)
5) API Documentation | TensorFlow Core v2.5.0
6) Natural Language Toolkit — NLTK 3.6.2 documentation
7) Convolutional Neural Network (CNN) | TensorFlow Core
8) NumPy Documentation
9) pandas documentation — pandas 1.2.4 documentation (pydata.org)
10) Working with JSON - Learn web development | MDN (mozilla.org)
11) SpeechRecognition · PyPI
12) HTML documentation — DevDocs
13) CSS documentation — DevDocs
14) W3.CSS Home (w3schools.com)
15) HTML Tutorial (w3schools.com)
16) JavaScript | MDN (mozilla.org)
17) Introduction · Bootstrap (getbootstrap.com)
18) Bootstrap 3 Tutorial (w3schools.com)
19) Simple explanation of convolutional neural network | Deep Learning Tutorial 23 (Tensorflow & Python) - YouTube
20) [DL] 8. CNN 1(Convolutional Neural Network Basics) | by Jun | jun-devpBlog | Medium