**RESEARCH PAPER**

**ON**

**VOICE ASSISTANT**

**USING**

**NEURAL NETWORK IN PYTHON**



**GUIDED BY**

**DR. VIKAS PANDEY**

**(ASSISTANT PROFESSOR)**

**SUBMITTED BY**

**NAME : TRILOK DEWANGAN**

**ROLL : 21072165**

**ENROLL: GGV/18/5075**

**ABSTRACT**

A VOICE ASSIATANT, also called VIRTUAL ASSISTANT is the application that accept user input in the form of voice commands and understand this voice commands, and the perform task for the user according to that correspondence voice inputs. The popular and advance virtual assistant currently include Amazon Alexa, Apple Siri, GoogleAssistant, Microsoft Cortana. In my project I have created an A.I. virtual assistant system using neural network with the help of python programming language and NLP(natural language processing) technique. The Virtual Assistant system which is able to perform all the task given by users in the form of voice commands like google search, wikipedia search, youtube search, open any application in windows, web scrapping, date, time, day, year etc. In addition to that here I have added a chatbot system that understands users emotions and give response according to the voice inputs.

**INTRODUCTION**

The virtual assistant is the implementation of Neural Network algorithm and various python module for helping the end user. Using this virtual assistant we can reduce the human efforts in various field like army, industrial area, chemical plants powerplants etc. There are lots of popular virtual assistant in the market like Amazon Alexa, GoogleAssistant, Microsoft Cortana. They have some limitation. The next evaluation can be done with the help of this voice assistant application like in household “turn on the air condition when temprature rises above 30”, “if there is any motion detects by security camera after 11pm than call Trilok”, “turn on or off the fan, light” . Most of the Virtual Assistants work basically on voice as communication. It focuses on processing of audio signal into the system, converting them to text and performing the required task. In general, speech processing consists the following: A Speech To-Text Module that converts speech signals to text, A Parser that extracts the semantic context, A Dialog Manager that determines system response through machine learning algorithms, An Answer Generator that provides the system response in text and A Speech Synthesizer that converts text to the speech signal. The voice assistant that I have made, firstly its train the dataset intent file that contains various tags, patterns, response in the form of list and dictionary. After training we get trained data. After that The voice assistant takes voice input from user, this voice assistant match this input with pattern of trained data if match get found then its generate response according to that user inputs. If input are found communicational type, so the response are directly gives to the user eg. input: “how are you” response: “I am good sir what about you”. And if inputs are found task related type then the respose are send to the task related file, here task function are called by Voice Assistant corresponding to the user input and response and task are perfomed by various task function. So this way our Voice Assistant system works.

**LITERATURE REVIEW**

The Speech Recognition Model is one of the most important part of a Virtual Assistant. Considering the various Neural Networks that are required for building up of a speech recognition system, it was necessary to survey the models that provided the insight by determining the accuracy and other factors of each Model. It was observed that High Accuracy and less Validation Accuracy was achieved for Convolutional Neural Network (CNN) model as compared to Basic Neural Network. Thus, proving that CNN is a better choice for speech recognition systems. Considering the Limitations for the Model, other parameters such as Word Error rate, throughput of the system was not taken into consideration.

Various Machine Learning Algorithms are used for speech recognition. It was found that on application of Auto- WEKA on various algorithms, determined Random Forest as the best algorithm which is useful for learning the dataset based on the training set. However in this survey, Speech samples consisting of noise was not tested for determining the scalability and robustness of the models[3]. In the Survey of scaling speech recognition using CNN, following metrics were taken into consideration:

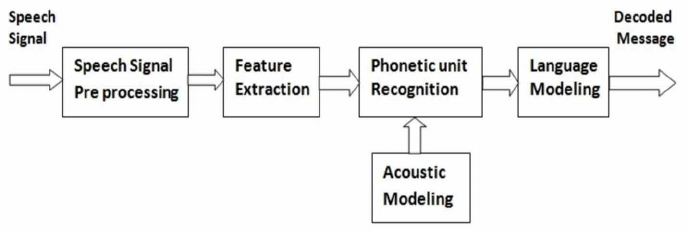
1. throughput,
2. Real-Time Factor(RTF) and latency, and
3. Word Error Rate (WER)

for the overall framework, helped in achieving an efficient model. But due to the increase in the number of the layers the implementation of the same was difficult. Other Algorithms such as the Long Short-Term Memory (LSTM) is very powerful in speech recognition and Hybrid model of Hidden Markov Model (HMM) and Gaussian Mixture Models (GMM) can give excellent results. In various Projects of developing a Virtual Assistant it was observed that the platform failed to support various other languages of the countries including China, Japan, India, etc. The survey paper provided with detailed study on the Recurrent Neural Networks (RNNs) that can be used for Speech Recognition System but with more research to be carried out on the same. However, the survey focused more on the Supervised Learning Models and less importance was given to the Unsupervised Learning Models. A Survey included the detailed comparison of the Personal Voice-Based Assistants available in the market namely, Google Assistant, Cortana, Alexa and Siri. It concluded that Google assistant gave good results in VR and HFI by achieving 60% accuracy. Siri achieved 44% accuracy in VR and HFI. Cortana was observed with decrease in accuracy close to 30%. Other results included that Alexa wasn’t suitable with simple questions whereas Cortana was poor in basic voice recognition. . The illustration to use AI-enabled content analysis has been discussed in one of the survey paper. The system can examine text of leadership speeches, content related to a specific organization. However, Only one type of content was analysed with limited samples and a Pre-defined Coding Scheme was used for the Project. Since, the survey of this paper also included the comparison of IBM Watson and Google DialogFlow, various Projects carried on these platforms were studied before arriving at the conclusion for a better Platform for the Project. A Project was based on successful implementation of IBM Watson in developing an application for health care purpose. This Project thus provided a base for building up of an AI Application with the help of IBM Watson. Various other Projects used IBM Watson as their platform for building the system which processed various queries with the help of its in-built Natural Language Processing (NLP) and Natural Language Understanding (NLU) Algorithms. Projects based on successful implementation of the Google DialogFlow for an Organization were studied. It provided an insight to the various technologies like the Google Cloud Platform, Google Cloud Vision API for integrating detection features in the system and Firebase Real Time Database for developing the Application. The system ensures security of database with the help of OAuth Authentication for accessing the system. However, most of the actions carried out with the help of Google services required Internet Connectivity while accessing the system and thereby failed to service the queries offline. Other Project, aimed to design a system for Educational purpose using Google DialogFlow. The proposed methodology consists of two main phases: Knowledge Abstraction and Response Generation. The methodology studied the deep learning model, The Decision Tree, that is been used in implementing the Dialogflow.

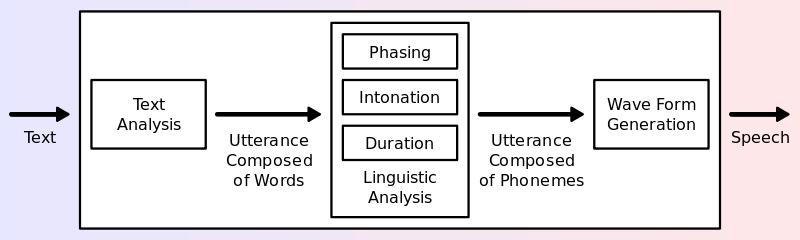
**RELATED WORK OR METHODOLOGY**

**(A)creating base**

* **Listen.py Module:** I used speechRecognition library to implement the Listen Module. Using this module assistant takes voice input from users using microphone. To takes input from user it uses google’s recognize\_google() api.



* **Speak.py Module:** To implement the speak module I used pyttsx3 library. Using this library assistant convert the text data to the audio format. It helps to respose to the users for any spacific query.



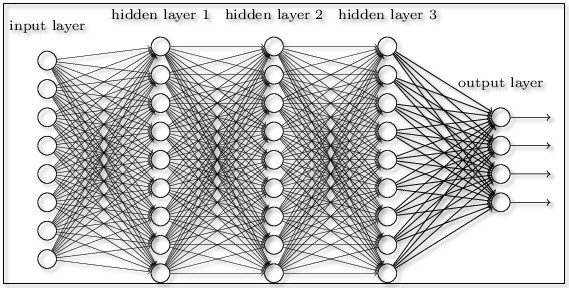
* **Intent.json:** It is a collection of list and dictionary that contains various pattern and responses. We have made it manually for doing some spacific task and give proper response.
* **Task.py Module:** This module contains various task function that can be demanded by the user. We have implemented all the task like google search, wikipedia search, youtube search, web scrapping etc. To implement this task we used lots of python library like webbrowser, pywhatkit, datetime, beautifulSoup, requests, os, translator etc.

**(B).model training**

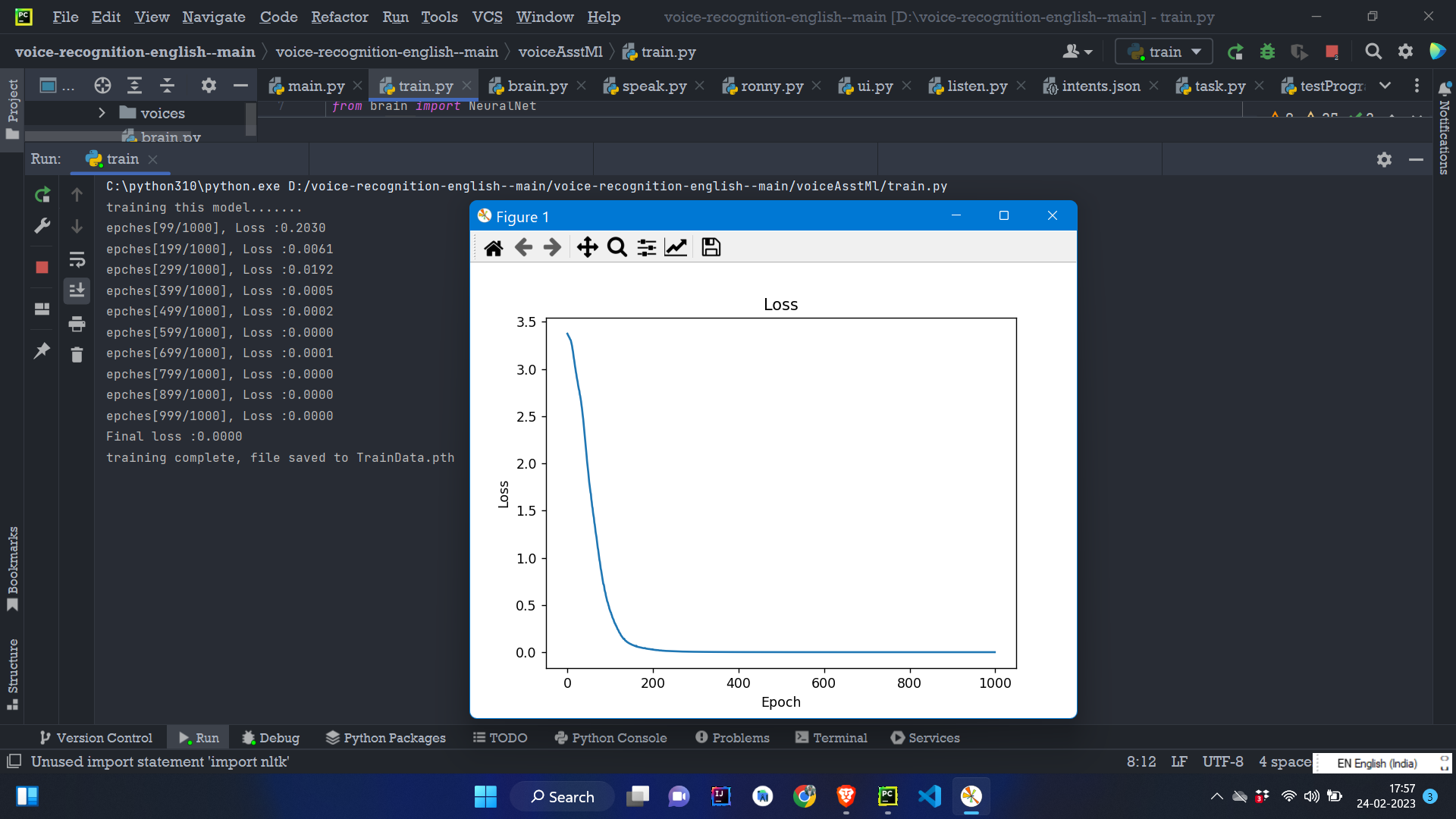
* **Brain.py:** this is the most important module in our voice assistant system, here we implemented NeuralNetwork architecture using **torch** python library for train our datasets. Here we used Linear model of Neural Network that contains one input layer, one output layer and three hidden layer. In this we used **Multilayer feed forward Neural Network** And alos used **relu** activation function for output computation.

**🡪Relu Activation function:** ReLU stands for rectified linear activation unit and is considered one of the few milestones in the deep learning revolution. It is simple yet really better than its predecessor activation functions such as sigmoid or tanh. { f(x)=max(0,x) }

ReLU function is its derivative both are monotonic. The function returns 0 if it receives any negative input, but for any positive value x, it returns that value back. Thus it gives an output that has a range from 0 to infinity.



* **NeuralNetwork.py:** In this module we have implemented tockenize(), bag\_of\_word(), stem() method that is very useful for finding the pattern in our input data. The tockenize() method breaks the sentence and find the word patterns. The stem() method reduce the redundency for same word frequency. And the bag\_of\_word() breaks the words and find the patterns between letters.
* **Train.py:** In this module we have implemented our training phase, here we load intent.json dataset fetch input and outputs, preprocessig it using NeuralNetwork module and fit it to the NeuralNetwork model. In our model we apply training iteration for 1000 epoches and we get 0 error after 600 epoches.



After training we perform performance evaluation to moniter the efficiency of the model for various iteration.

* **TrainData.pth:** this is the converted dataset file that extension are changed from **.json to .pth** to get fast access of response. This contains unreadable forms of data, it is only understandable by the computers.

**(C).GUI CREATION**

* **UI.py:** in this module we implemented the graphical user interface for better interaction with our Voice Assistant**.** Here we use pyQt5 python library for make gui. This UI conatains buttons, text-browser, labels etc. for interacting with the users.

**(D).Main Entry Module**

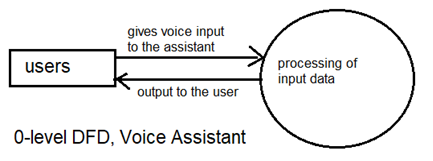
* **Ronny.py:** this is the very important module from where our voice assistant starts works. In this module we use method**(takeCommand())** of listen module**,** using this method the assistant starts taking input from user. Here also used **say()** method of speak module to say anything.Basically in this module our voice assistant takes input from user and find the pattern, after that this pattern are matching with the dataset that is already available on the **TrainData.pth** file. If matching are found or not found than it will gives response accroding to the input pattern. If any task related query are found than control are sent to the **task.py** module to perform spacific task. In this module we used lots of module that we have discussed above.
* **Speak.py**
* **Task.py**
* **TrainData.pth**
* **Listen.py**
* **NeuralNetwork.py**
* **Torch(built in module)**
* **Json(build in module)**
* **Main.py:** this is the starting point of our voice assistant where we import **gui.py** module and used its methods and classes**.** By using this module we start run our gui element. We have imported ronny.py module in this module. If we click on start button it call the funciton of ronny.py module and starts working. And after clicking quite button it will terminate and close the voice assistant application.

**TECHNOLOGY USED**

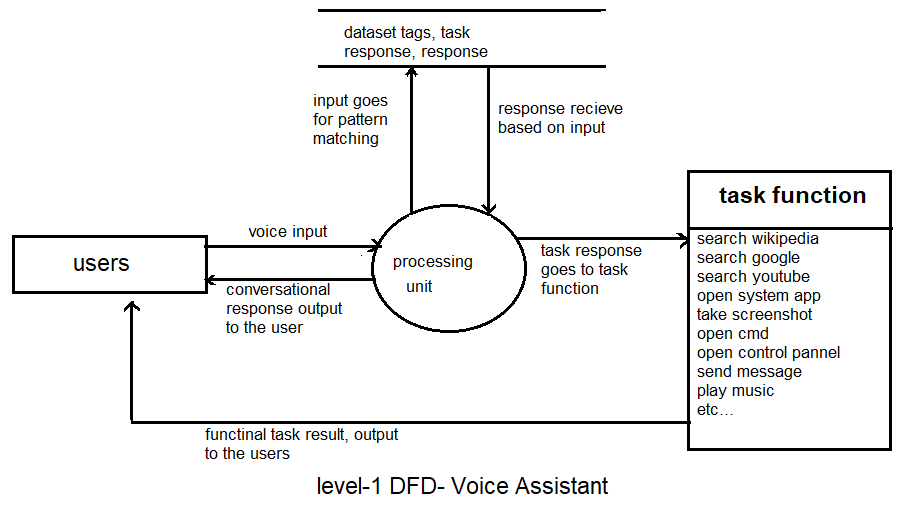
To create this application we have used **python** programming language and **pycharm** ide and various python library. QtDesigner for creating gui with pyqt5.

**4.DATA FLOW DIAGRAM/ WORK FLOW**

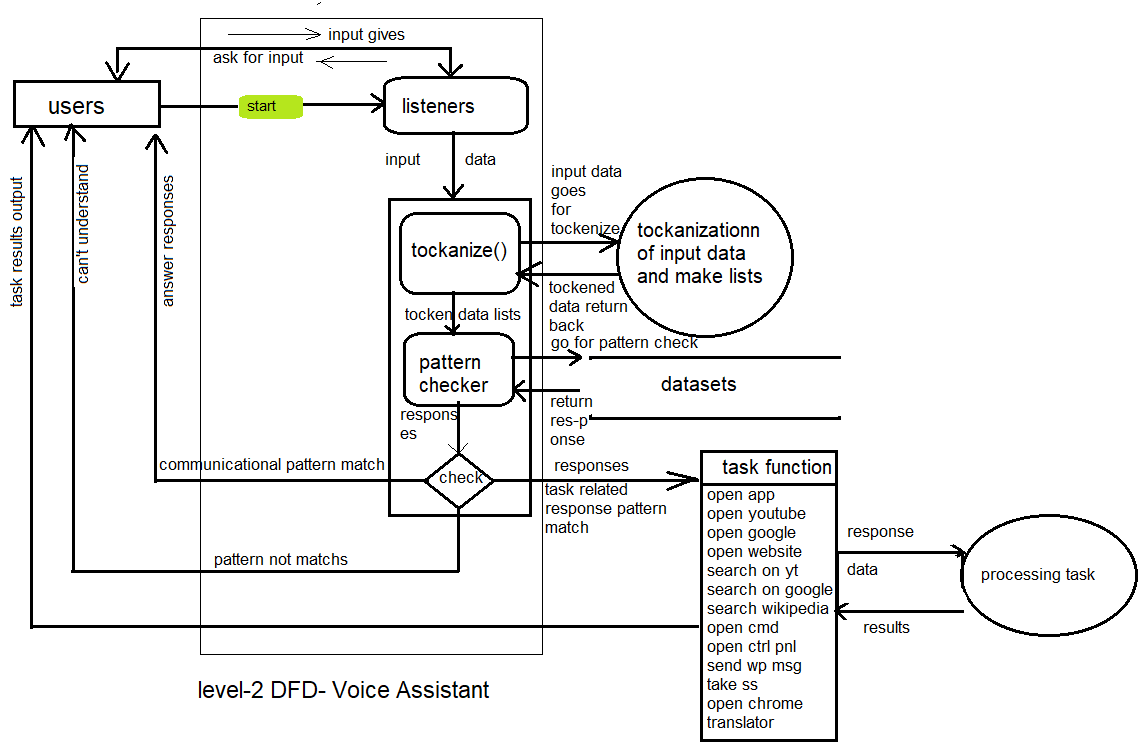
0-LEVEL DFD



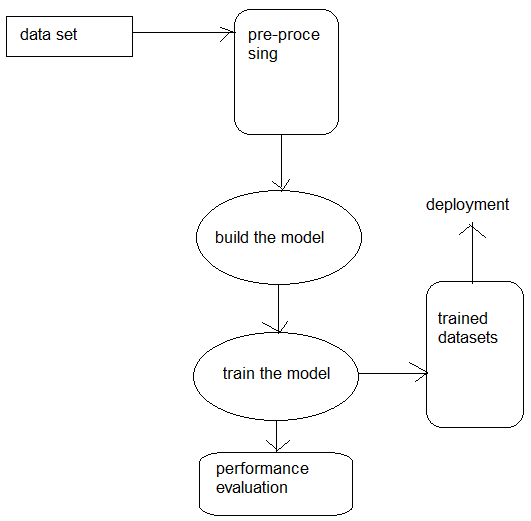
1-LEVEL DFD



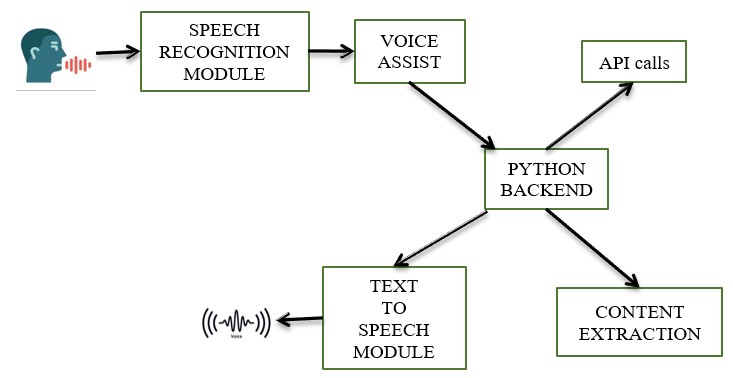
2-LEVEL DFD



FOR MODEL TRAINING DIAGRAM



**RESULTS AND DISCUSSION**

****

In this discussion we will briefly discuss the whole process of our voice assistant. Firstly we train the dataset that contains various input and response in the form of list and dictionaries labelles as responses, patterns, tags. To train this data set we used multilayer feed forward neural netowork and relu activation function. We have train the model for 1000 iteration/ epoches for better efficiency. After training process we get trained dataset.

Our voice assistant firstly starts taking input in the form of voice using microphone this voice input are cleaned and decoded by the machine and converts it into the text format and applying various preprocessing steps like tockenize(), stem(), bag\_of\_word() for pattern evaluation. After that this text input are matched with the trained dataset input and machines decides that which type of response and task have to done. If any emotional patterns match found than machine response directly to the user in the form of voice. And if the patterns are matching like any task related, than control goes to task.py module here various task are performed by task function. To perform any task the python code call the particular api, and using this api machine extract information and provided to the users.

**LIMITATIONS**

* It does not have any big dataset for responsing every kind of query.
* It does not have all type of task function that satisfies all the users.
* It does not have any user identification technique.
* It does not have any encryption method.

**ENHANCEMENT IN FUTURE**

* We can include more input and output pattern in dataset for all kind of query of all type of users.
* It can be added all type of task funcion that fully control our system.
* It can be added user signup and login page for user identificaiton.
* It can be added encryption method for security camera.
* We can implement it in the self driving cars or normal cars to control.
* We can use this application in robotics in the industrial, army, powerplant area. It will very helpful as like a living person where we have to only ask for doing any task.
* It can be added home automation system machanism.
* This can be make greate evaluation in arm forces for any country, by using this application we can control our arm robots by giving voice command from anywhere. It can save lots of life of soldiers by using robotics.
* Can be add more language.

**CONCLUSION**

The main objective of this paper is the AI Assistant can perform all the operations such as search engine tasks, setting remainders ,date, time, location, answering to the questions, playing youtube videos and also finds the emotion of a person by analysing the speech of a person. The scope of this paper is most of the people are busy in their work life they don’t even find time to relax because of which their mental pressure can increase , so this AI Assistant can help them as this assistant is designed to talk with people and perform their desired tasks and also whenever people get bored or stressed they can communicate with this AI as this assistant can understand their emotion and talk with people in a friendly way.

**REFERENCES**

* AI System With Voice Modulation Using Neural Network M. Roshini1, S. Nandhika2, S.Prathipa3, Dr. Rajasekar Velswamy4 1, 2, 3, 4Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.
* Official python documentaion <https://docs.python.org/3/>
* Youtube.com
* Geeksforgeeks.org
* <https://help.imsl.com/c/6.0/stat/default.htm?turl=multilayerfeedforwardneuralnetworks.htm>
* <https://www.geeksforgeeks.org/speech-recognition-in-python-using-google-speech-api/>