**Speech Recognition Using Python**

Abstract:.

Speech recognition technology is one of the fast growing engineering technologies. It has a number of applications in different areas and provides potential benefits. About 20% of the world's people suffer from various disabilities, many of them blind or unable to use their hands effectively. Some people lose their hands in accidents. In those special cases, the speech recognition system provides them an important help, so that they can operate the computer through voice input and share information with people. The project is capable of recognizing speech and converting input audio into text.

# **INTRODUCTION**

Speech recognition is a technique that enables a computer to capture the words uttered by a human with the help of a microphone. These words are later recognized by the speech identifier and finally, the system outputs the recognized words. The process of speech recognition consists of various stages, which will be discussed in the following sections. Speech recognition is used for two main purposes. The first dictation is in the context of that speech recognition, which is the translation of words into text and the second is to control the computer, to develop software that is probably capable enough of the world to allow the user to access various applications by voice Authorize to operate. If we write with voice, our writing speed will be doubled or tripled. Speech recognition is an alternative to the keyboard. Using speech recognition a person who is unable to write or does not want to type can do almost anything that can be done with the keyboard. Apart from all the benefits and benefits, we are unable to develop a hundred percent correct speech recognition system. There are many factors that reduce the accuracy and performance of a speech recognition program. The speech recognition process is easy for humans, but it is a difficult task for a machine, speech recognition programs are less intelligent when compared with the human mind, this is because of the human mind's ability to think, understand and react. Naturally, while a computer program is complex, it needs to understand spoken words in relation to their meanings and to create an adequate balance between words, noise, and spaces. A human has the ability to filter out noise from a speech, while a machine requires training, a computer needs assistance to separate sound from noise.[1][2]

**2. RELATED WORK**

**2.1** The first speech recognition system, Audrey, was developed in 1952 by three Bell Labs researchers. Audrey was designed to recognize only digits

**2.2** 10 years later, IBM introduced its first speech recognition system IBM ShowAbox, which was able to recognize 16 words including digits. It can identify commands like "five plus three plus eight plus six plus four minus nine, total" and will print the correct answer, ie 17

**2.3** In the 1970s, the Defense Advanced Research Projects Agency (DARPA) contributed greatly to speech recognition technology. DARPA funded a program called Speech Understanding Research for about 5 years 1971–76 and, eventually, Harpy was developed that was able to recognize 1011 words. This was a huge achievement at that time.

**2.4** In the 1980s, the Hidden Markov Model (HMM) was implemented for speech recognition systems. The HMM is a statistical model that is used to model problems involving sequential information. It has a very good track record in many real-world applications including speech recognition.

**2.5** In 2001, Google introduced a voice search application, which allowed users to search for queries by speaking from a machine. It was the first voice-enabled application that was very popular among people. This made interaction between people and machines much easier.

**2.6** As of 2011, Apple launched Siri, offering a real-time, fast, and easy way to interact with Apple devices using your voice. By far, Amazon's Alexa and Google's Home are the most popular voice-command-based virtual assistants that are widely used by consumers worldwide. [3]

# **DATASETDESCRIPTION**

The dataset used in this project work has been taken from the Kaggle.com available at (<https://www.kaggle.com/c/tensorflow-speech-recognition-challenge>).It is a speech command datasheet issued by Tensorflow. The dataset has 65,000 long pronunciation of 30 short words by thousands of different people.

## **Data Preprocessing**

In the dataset few recordings are less than 1 second and sampling rate is very high. To solve these problems we perform resampling on the dataset from sample rate 16,000 to 8,000 and we remove the commands which are less than 1 second.

# **SYSTEM DESCRIPTION**

**4.1** To build this system, we have used various modules like Keras, TensorFlow, Librosa, Sounddevice, os, numpy, matplotlib , IPython.

**4.1.1 Keras** runs on top of open source machine libraries such as Tensorflow, Theano, and Cognitive Toolkit (CNTK). **TensorFlow** is the best-known symbolic mathematics library used to create neural networks and intensive learning models. TensorFlow is very flexible and the primary advantage is distributed computing. Theano and TensorFlow are very powerful libraries but difficult to understand for building neural networks. Keras is based on a minimalist structure that provides a clean and easy way to create intensive learning models based on TensorFlow or Theano. Keras is designed to quickly define intensive learning models. Well, Keras is an optimal choice for deep learning applications. [4]

**4.1.2 Convolutional Neural Networks** (**ConvNets** or **CNNs**) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. ConvNets have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self driving cars, and seeing these efficiency and effectiveness of CNN, we have used them to build our audio to text conversion project. [5] We have used following layers of the Convolution Neural network-

**4.1.2.1 Dense -** A dense layer represents a matrix vector multiplication. [6]

**4.1.2.2 Dropout -** The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. [7]

**4.1.2.3 Flatten -** The role of the Flatten layer in Keras is super simple: A flatten operation on a tensor reshapes the tensor to have the shape that is equal to the number of elements contained in tensor non including the batch dimension. [8]

**4.1.2.4 Conv1D -** This layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs. If use\_bias is True, a bias vector is created and added to the outputs. Finally, if activation is not None, it is applied to the outputs as well.[9]

**4.1.2.5 Input -** The input layer takes a shape argument that is a tuple that indicates the dimensionality of the input data. It is used to instantiate a Keras tensor. A Keras tensor is a TensorFlow symbolic tensor object, which we augment with certain attributes that allow us to build a Keras model just by knowing the inputs and outputs of the model. [10]

**4.1.2.6 MaxPooling1D -** The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. [11]

**4.1.2.7 Librosa**-We have also used Librosa which is a python package for music and audio analysis. It provides building blocks necessary to create music information retrieval systems. [12]

**4.1.2.8 Matplotlib**.**pyplot-** Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. [13]

**4.1.2.9 NumPy**-NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. [14]

**4.2** **Building Model**

**First Input layer –** It has given one argument “shape = (8000, 1)”. It is used to reshape input data into 1D column for 8000 elements.

**Second Conv1d layer 1 –** In this layer the dimensionality of the output space (i.e. the number of output filters in the convolution) is 8 and kernel\_size which specifying the length of the 1D convolution window is equal to 13.

**Third Conv1d layer 2 –** In this layer the dimensionality of the output space (i.e. the number of output filters in the convolution) is 16 and kernel\_size which specifying the length of the 1D convolution window is equal to 11.

**Fourth Conv1d layer 3 –** In this layer the dimensionality of the output space (i.e. the number of output filters in the convolution) is 32 and kernel\_size which specifying the length of the 1D convolution window is equal to 9.

**Fifth Conv1d layer 4 –** In this layer the dimensionality of the output space (i.e. the number of output filters in the convolution) is 64 and kernel\_size which specifying the length of the 1D convolution window is equal to 7.

**Sixth Flatten layer** – Flatten layer prepares a vector for the fully connected layers.

**Seventh Dense Layer 1 –** For this layer value of argument units = 256 which represent dimensionality of the output space.

**Eighth Dense Layer 2 –** For this layer value of argument units = 128 which represent dimensionality of the output space.

**Ninth Output Layer –** A dense layer is used as output layer with units = length of labels which is dimensionality of the output space.

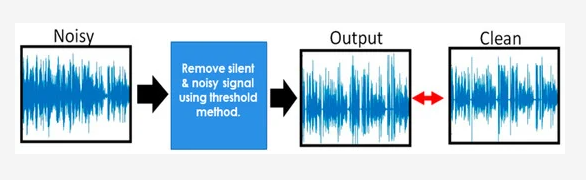


FIG-1

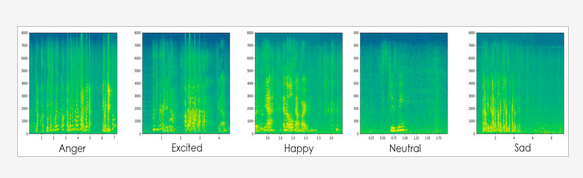
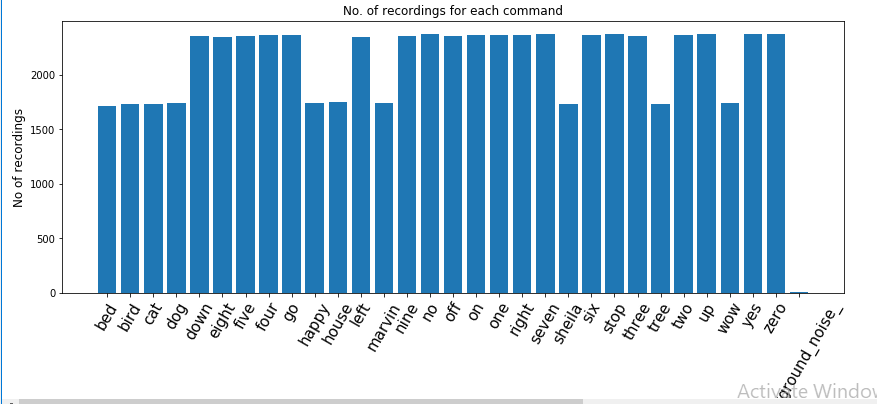


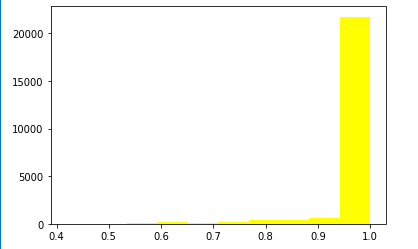
FIG-2

**5. Results and discussions:**

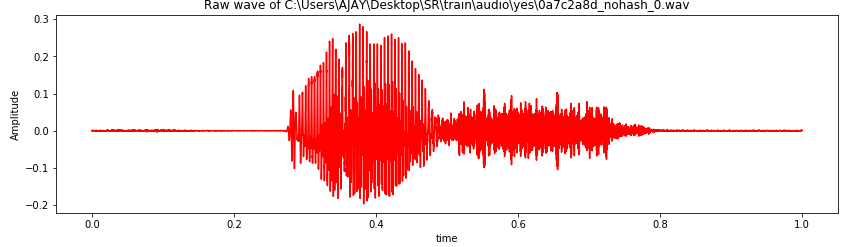
In this project, we have used lots of voice commands and, this is the output, which displays the number of recordings of each command.



Here, this is the output of the code where we have checked the duration of the all audio commands. Through this histogram it is clear that maximum voice commands are of one second.

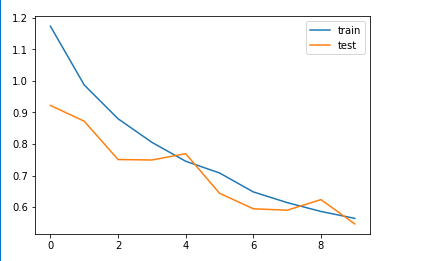


Data Exploration and Visualization helps us to understand the data as well as the preprocessing steps in a better way. This is the output of the visualization of Audio signals in time domain series.



Next, we have the output of the code where we have trained our model and tested it.

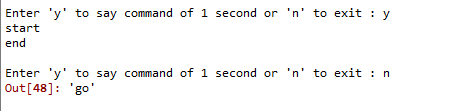
Model predict the voice\audio command with the accuracy of 0.8765. In this graph ,Blue line shows the training part and orange line shows the testing.



This is the final output of our code ,where it asks from the user to speak for one second.

At start user speaks and at end command user stops ,and then the model predicts the voice and convert that audio into text.

‘This model successfully predicts my command ‘go’ and converted into text’



**6. CONCLUSION**

In this project we have trained speech recognition model which can few basic commands from speech to text in real time. It can recognize commands such as "yes", "no", "up", "down", "left", "right", "on", "off", "stop", "go", etc. This project has been executed in two stages. The first stage include training and testing of model based on deep learning using keras library consist of nine layers of neural network including input layer and output layer. The second stage mainly includes taking one second audio input from user and predicting its value.

**REFERENCES**

[1] https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition.htm

[2] Lawrence Rabiner, Biing-Hwang Juang - Fundamentals of Speech Recognition-Prentice Hall (1993).

[3]https://sonix.ai/history-of-speech-recognition#:~:text=The%20first%20speech%20recognition%20systems,to%2016%20words%20in%20English.

[4]https://keras.io/

[5]https://keras.io/api/layers/convolution\_layers/

[6] https://keras.io/api/layers/core\_layers/dense/

[7]https://www.tensorflow.org/api\_docs/python/tf/keras/layers/Dropout

[8] https://keras.io/api/layers/reshaping\_layers/flatten/

[9] https://www.tensorflow.org/api\_docs/python/tf/keras/layers/Conv1D

[10]https://keras.io/api/layers/core\_layers/input/

[11]https://www.tensorflow.org/api\_docs/python/tf/keras/layers/MaxPool1D

[12]https://pypi.org/project/librosa/

[13]https://matplotlib.org/[12]https://pypi.org/project/librosa/

[14]https://www.w3schools.com/python/numpy\_intro.asp#:~:text=NumPy%20is%20a%20python%20library,you%20can%20use%20it%20freely.