Speech Bot Voice based Search Engine

A Project Report Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Bachelor of Technology (Hons.)
in
Computer Science and Engineering

by

Sai Manoj Konidana
(Roll No. 2017BCS0030)

to

DEPARTEMENT OF COMPUTER SCIENCE
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
KOTTAYAM-686635, INDIA
April 2021

DECLARATION

I, **Sai Manoj Konidana** (**Roll No: 2017BCS0030**), hereby declare that, this report entitled **"Speech Bot Voice based Search Engine"** submitted to Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology** in **Computer Science and Engineering** is an original work carried out by me under the supervision of **Dr.Shajulin Benedict** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Kottayam-686635 **Sai Manoj Konidana**

April 2021

CERTIFICATE

This is to certify that the work contained in this project report entitled **"Speech Bot Voice based Search Engine"** submitted by **Sai Manoj Konidana** (**Roll No: 2017BCS0030**) to Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology** in **IIIT kottayam** has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Kottayam-686635	Dr.Shajulin Benedict
April 2021	Project Supervisor

ABSTRACT

While the Internet is adapting from a text based medium to a diverse multimedia platform, there is a need to enable queries based on the multimedia content. This project focuses on building an optimised and robust model(neural network) trained on massive audio data set for speech recognition that is used in enabling audio based search querying to achieve voice enabled search for multimedia content on one or more search engines. We propose an architecture for building the speech recognition model to use and build transcriptions for audio data to be used in an efficient search engine. We introduce a convolutional recurrent network with focus on speech command recognition that identifies transcripts and words in noisy audio files.

Contents

1 Introduction **1**

1.1 How are audio based query retrieval systems better ? 2

1.2 Solution : 2

2 LITERARY SURVEY **4**

2.1 A neural attention model for speech command recognition [5] 4

2.2 Audio-visual speech recognition using deep learning : [11] . . . 6

2.3 Deep Speech: Scaling up end-to-end speech recognition [6] . . 7

2.4 A P2P Audio Search Engine Based on Keyword Spotting [7] . 8

2.5 A Music Search Engine Built upon Audio-based and Web-based Similarity Measures [9] 9

3 Proposed Architecture : **11**

3.1 1.Transcoder : 12

3.2 Speech Recognizer : 12

3.3 Library : 13

3.4 Indexer : 13

3.5 Architecture of Speech recognition model : 13

3.5.1 Summary of the neural network : 14

3.5.2 Libraries used for training : 15

3.6 Alternate approach : 15

4 RESULTS AND DISCUSSION 16

4.1 Setup : 16

5 Future Works : 22

6 CONCLUSION 23

7 References 24

Chapter 1

Introduction

While the popularity and usage of graphical and audio based content on the internet grows, specifically, large pools of audio and video streams on various content creation platforms , robust and optimal ways to automatically find the related segments of the multimedia content has become a necessity. However, conventional search engines are mostly limited to text and image indexing and many online documents, especially video and audio, are often omitted by the classical query retrieval systems.

Most of the audio data archived on the Web are simple lists of links to audio files of various sizes. Any sort of indexing doesn't exist to achieve accurate searches on these multimedia. A searchable index would make these archives of multimedia much more accessible for users interested in a particular search.

1.1 How are audio based query retrieval systems better ?

1.With voice search, web page admins can easily provide results that the users need, which increases traffic to their landing pages.

2.Short-tail keywords: Have diminished importance and conversational search is bound to decrease their prominence even further.

3.People don't use voice search the same way they type a query into a search engine. They ask more direct queries, to get more relevant answers.

4.This is where long-tail keywords in your content come in handy. Using these keywords helps increasing the chances of the relevant content getting prioritised in voice search engine result pages.

5.Focus content on answering FAQs

6.Unlike text search, voice search can enable colloquial too considering how people generally speak, to develop content to match their tone.

1.2 Solution :

A simple and conventional approach to solve this problem is to generate the transcriptions automatically for all the audio(or video) content using a speech recognition system with a wide range of vocabulary. However, speech recognition technology is currently inaccurate, particularly when the audio data is noised and degraded due to bad recording conditions, unwanted compression parameters etc.

Therefore, in this phase we focus on building an efficient automatic speech recognition model that is robust against noise first.

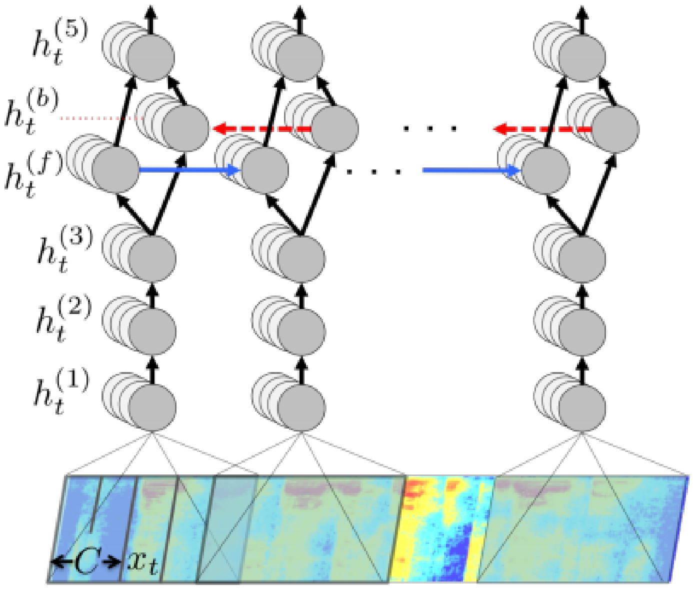
Several attempts have been made in the past to build these recognition models like, Audio-visual speech recognition, peer to peer keyword spotting systems, Recurrent neural networks that don't require phoneme dictionaries for training, ensemble neural networks etc.

In the following section we take a closer look at each of these researches and extract critical points on pros and cons of these various models and build a fine and optimal architecture for Automatic Speech recognition.

Chapter 2

LITERARY SURVEY

2.1 A neural attention model for speech command recognition [5]



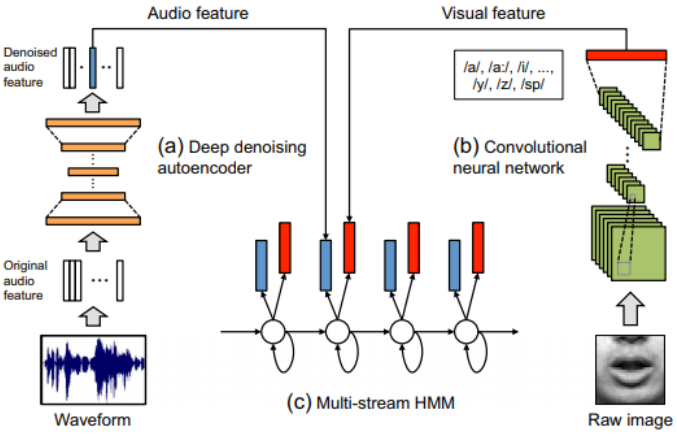
Audio files might contain a word or a command that can be used anywhere in the length of the file, it can be assumed that any model that is able to classify an audio should also be trained on accurate region of interest. Therefore, the attention mechanism has been proposed in this paper to do this particular task.

This model starts by computing the mel-scale spectrogram of the audio using non-trainable layers implemented using the kapre library. The input to the model is the noisy audio data with a preset sampling rate. In this approach Mel-scale spectrogram is computed using 80-band mel scale, 1024 discrete Fourier transform points and hop size of 128 points.

After the spectrogram computation, a set of convolution and bidirectional layers are applied to the spectrogram output only in the time dimension to extract local relations in the audio file. A set of two bidirectional LSTM units is used to capture both forward and backward dependencies in the audio file.

And then, output vectors of the final LSTM layer is evaluated, projected with a dense layer and used as query vector to identify the most relevant part of the audio file to the label. The median vector of the LSTM output is chosen as the voice command is expected to be centered in the audio files.

Finally, the weighted average of the output is fed into 3 fully connected layers for classification.



2.2 Audio-visual speech recognition using deep learning : [11]

This paper is based on training a hidden markov model with audio inputs and also the visual inputs of mouth area images to build an efficient multi-modal neural network.

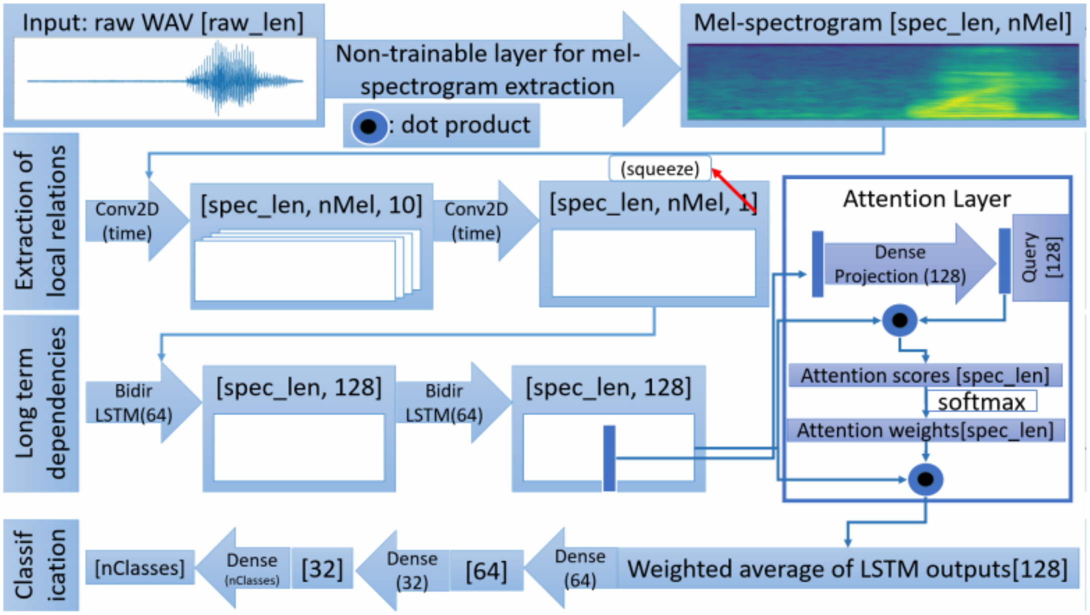
This study introduces a hidden Markov model network for noise-free Automatic speech recognition. Initially, a denoising variational autoencoder is used for extracting noise-free audio features.

By making the training data for the network with consecutive multiple steps of damaged audio features and the corresponding denoised features, the NN is trained to produce clean audio features from the corresponding inputs.

Then, a CNN is used to extract visual features from raw mouth area images. By preparing the training data for the CNN as pairs of raw images and the corresponding transcript label outputs, the network is trained to predict transcript labels from the corresponding mouth area images.

Finally, a multi-stream HMM is implemented for integrating the acquired audio and visual HMMs independently trained with the respective features.

2.3 Deep Speech: Scaling up end-to-end speech recognition [6]



This paper presents a speech recognition system developed again using peer too peer deep learning. This architecture is significantly simpler than traditional speech systems, that depend on rigorously engineered processing pipelines. These Conventional systems produce quite inaccurate outputs when used in noise filled environments.

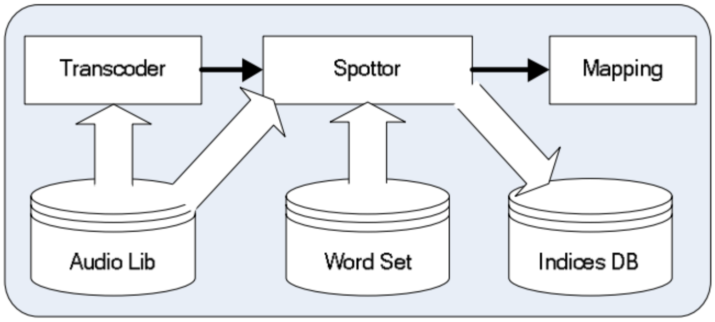
In contrast, this system does not require specially designed components to reduce background noise, resound, multiple speakers or speaker variation, but instead directly learns features that are resistant to such effects. It doesn't require a phoneme dictionary, nor even the concept of a phoneme.

This approach is a quite optimized using the Recurrent neural network training system which uses multi-programming accelarators as well as conventional data production techniques that allow to efficiently obtain data with a large amount of variance for training.

2.4 A P2P Audio Search Engine Based on Keyword Spotting [7]

While searching for audio content on the web, an efficient content-based audio search engine is required. In this paper, it has been demonstrated about an audio search engine called 'ASEKS' based on keyword spotting technology in a peer-to-peer network.

The indexing model identifies data in local audio documents and generates indexing for user queries. The P2P networks distributes these query and gathers the results and produce to the user.



In ASEKS, machines are divided into two classes: leaf nodes and hub nodes.

Leaf nodes are the most common nodes in the network. They have limited resources like processor, memory, hard disk or bandwidth. They are only allowed to be connected to the hubs. On the other hand, hub nodes form an important and active part of the network infrastructure, organizing surrounding nodes, filtering the traffic.

keyword spotting and indexing are attained within each node during indexing process. Each node processes the audio files saved on its disk and maintains a database of indices. During query retrieval, the initiator sends its query to the hub connected directly with it. Hubs distributes the query to other hubs,according to a query forwarding algorithm.

If a Peer receives a query it has not processed before, it rerun the indexing process for the keyword of this query and adds the result into database. Peers that receive and finish the query then return their responses to the initiator. Query initiator collates the responses and gives the result.

2.5 A Music Search Engine Built upon Audio-based and Web-based Similarity Measures [9]

This approach to automatically builds a search engine for large-scale audio documents that can be queried through natural language. While other conventional approaches heavily depend on explicit manual annotations or labels as target variable data and meta-data assigned to the individual audio chunks, this system automatically derives descriptions by making use of methods from web retrieval and audio information retrieval.

Based on the ID3 tags of a collection of mp3 files, this retrieves relevant web pages through google queries and use the contents of these pages to characterize the audio pieces and represent them by term vectors.

By including coexisting information about acoustic similarity it is able to both reduce the dimensionality of the vector space and improve the performance of query retrieval, which indirectly reflects in the quality of the results.

Based on the different observations made in the above works, an architecture that covers most of the critical points extracted has been proposed in the next section.

Chapter 3

Proposed Architecture :

I.Build an audio Recognition system that has a wide range of vocabulary, that can identify words or transcripts in noisy audio data and acoustic models in various languages.

II.Generate a transcription automatically using the speech recognition system.

III.After achieving a satisfactory model for indexing audio from the Web, fetch the audio documents from the Web and build an index from the generated data.

IV.Keep a link to the original documents , similar to traditional search engines.

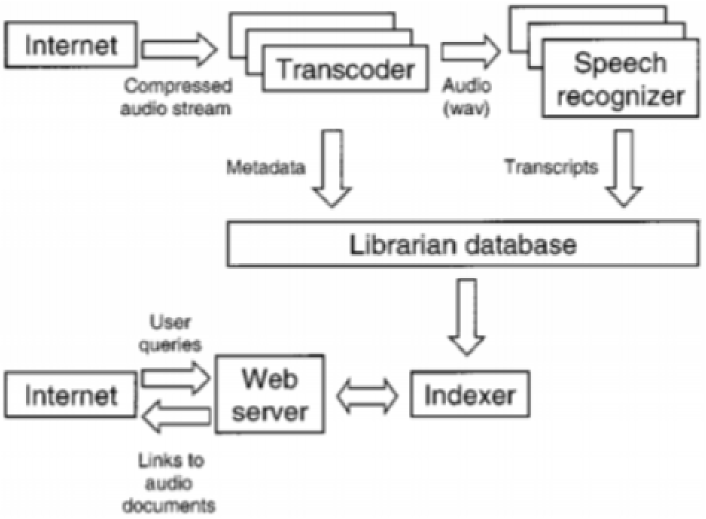
The system is divided into 4 major modules :

1.Transcoder

2.Speech Recognizer

3.Library

4.Indexer



3.1 1.Transcoder :

Fetch and decode video and audio files from the web. For each element, it extracts the meta-data, download the files to a temporary local repository.

This meta is later used by the Librarian database to identify and track the document while the user's query gets processed by the engine.

3.2 Speech Recognizer :

1.Either build a robust convolutional neural network or get a pre-trained model and use transfer learning to build the recognition system.

2.Loading the data set (For this project we have used Google Speech Command Datasets).

3.3 Library :

Stores meta-data and other information required for producing the results corresponding to a user's query.

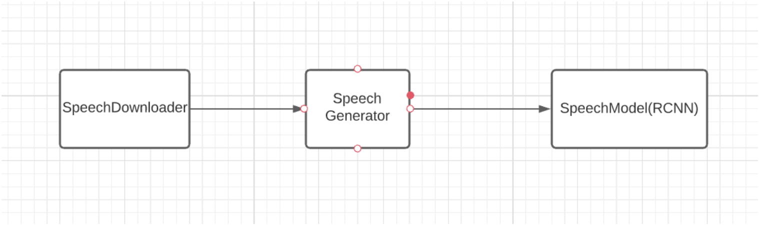
3.4 Indexer :

The indexer provides a list of audio files based on the transcription produced by the speech recognizer.

Supplies the user interface with a list of documents that match a user's query.

The indexer also retrieves the location of these matches within the documents.

3.5 Architecture of Speech recognition model :



1.Speech downloader :

Loads the Google Speech Command Datasets for training, performs preprocessing, and converts audio data into numpy arrays as training inputs.

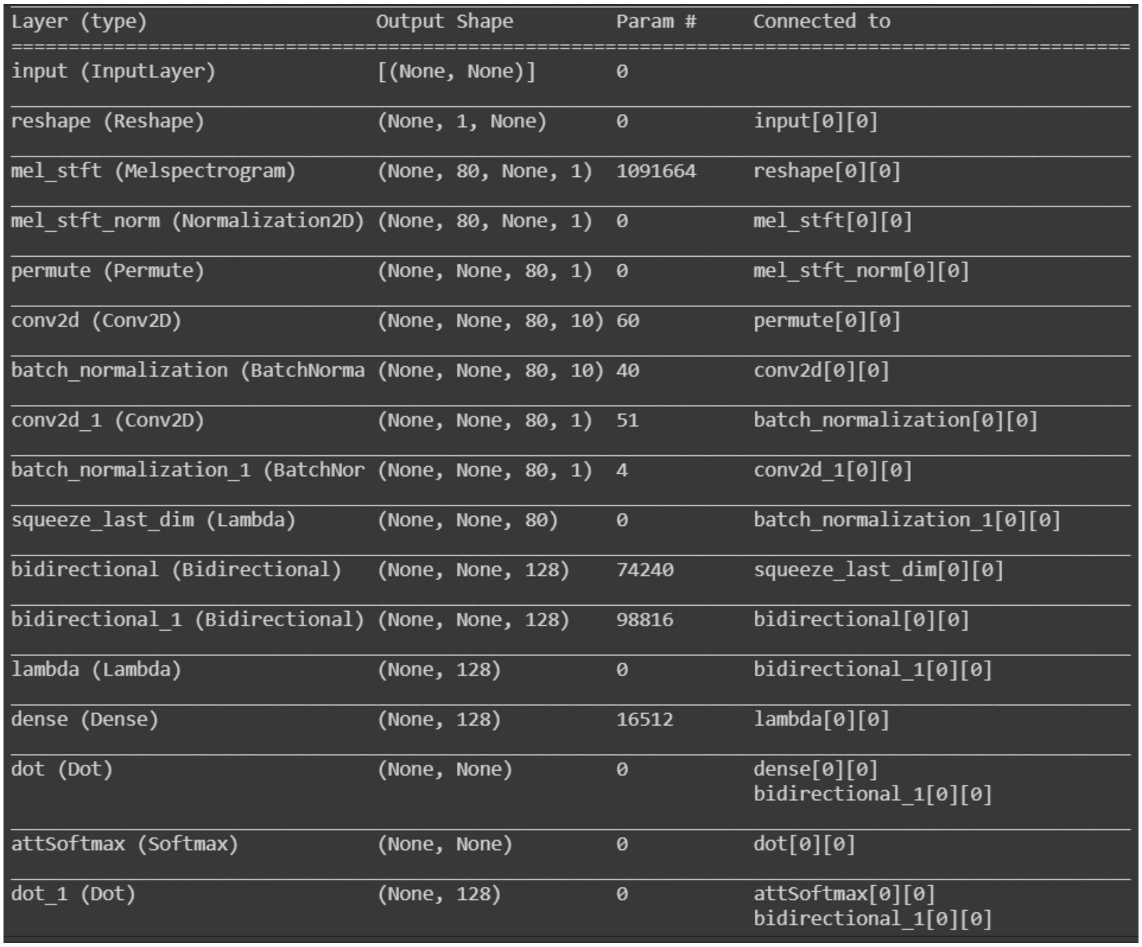
2.Speech Generator :

Loads the target variables i.e, the labels or the words that exist as transcripts in the audio files that we have taken as input.

3.Recognition Model:

The Neural Network that has multiple Permutational, Convolutional, bidirectional layers added as hidden layers.

3.5.1 Summary of the neural network :



```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input (InputLayer)              [(None, None)]       0
__________________________________________________________________________________________________
reshape (Reshape)               (None, 1, None)      0           input[0][0]
__________________________________________________________________________________________________
mel_stft (Melspectrogram)       (None, 80, None, 1)  1091664     reshape[0][0]
__________________________________________________________________________________________________
mel_stft_norm (Normalization2D) (None, 80, None, 1)  0           mel_stft[0][0]
__________________________________________________________________________________________________
permute (Permute)               (None, None, 80, 1)  0           mel_stft_norm[0][0]
__________________________________________________________________________________________________
conv2d (Conv2D)                 (None, None, 80, 10) 60          permute[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma (None, None, 80, 10) 40          conv2d[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, None, 80, 1)  51          batch_normalization[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, None, 80, 1)  4           conv2d_1[0][0]
__________________________________________________________________________________________________
squeeze_last_dim (Lambda)       (None, None, 80)     0           batch_normalization_1[0][0]
__________________________________________________________________________________________________
bidirectional (Bidirectional)   (None, None, 128)    74240       squeeze_last_dim[0][0]
__________________________________________________________________________________________________
bidirectional_1 (Bidirectional) (None, None, 128)    98816       bidirectional[0][0]
__________________________________________________________________________________________________
lambda (Lambda)                 (None, 128)          0           bidirectional_1[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 128)          16512       lambda[0][0]
__________________________________________________________________________________________________
dot (Dot)                       (None, None)         0           dense[0][0]
                                                                 bidirectional_1[0][0]
__________________________________________________________________________________________________
attSoftmax (Softmax)            (None, None)         0           dot[0][0]
__________________________________________________________________________________________________
dot_1 (Dot)                     (None, 128)          0           attSoftmax[0][0]
                                                                 bidirectional_1[0][0]
```

3.5.2 Libraries used for training :

1.Tensorflow

2.Kapre

3.Pandas

4.Librosa

5.Tqdm

6.Matplotlib

3.6 Alternate approach :

1.After recognizing the voice input successfully an alternate way is proposed to search the query.

2.The recognized word sequence will be searched for on existing search engines like google and the search results will be checked to find out if they match the user query or not.

3.If they did not match, an alternative word sequence will be checked

4.And this process continues so on to find the right one Once relevant result is found the system will peep to inform the user about the results.

Chapter 4

RESULTS AND DISCUSSION

4.1 Setup :

1.We connect to remote resources using Google Colab. Set the CPU memory limit to 256 MB. Set the GPU to a memory limit of 11 GB.

2.Import tensorflow and set up version 2.0.

3.Import all the required libraries :

Librosa : A python audio library to convert input audio files to their power spectrogram and show results.

Numpy : For converting and normalizing training inputs into arrays.

Matplotlib : For data visulisation.

4.Import the Google speech command dataset containing massive amount of audio files in WAV format and their target labels (the word that is contained in the corresponding audio file).

In our experiment we imported 84,849 audio files for training and testing.

5.We have 36 output classes(36 probable words that are contained in all the audio files).

6.Pre-process and Convert the WAV files into numpy arrays.

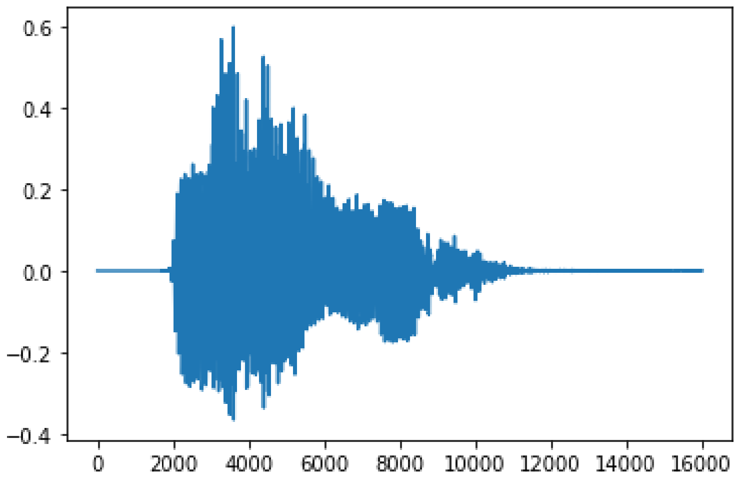


Figure 4.1: Frequency vs Time graph for 16,000 audio inputs

7.Build the Mel-spectrogram for the training inputs.Mel spectrogram does to audio inputs is similar to what a convolution does for image inputs.It extracts the features from WAV or any other audio files by passing the raw audio waveforms through filter banks.

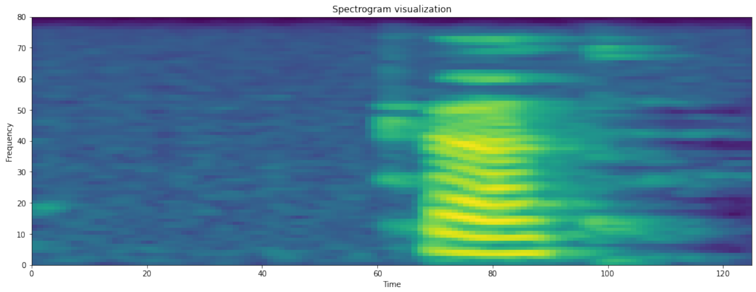


Figure 4.2: Mel spectrogram visualisation

8.Build the Neural Network with melspectrogram, convolutional, normalization and bidirectional layers.

9.Initial learning rate is set to 0.01 and is dropped for each epoch by e,0.4 depending on the obtained accuracy and loss of every epoch.

10.In our current experiment, model is trained for 50 epochs and is stopped earlier if convergence is reached.

11. Training Accuracy per each category :

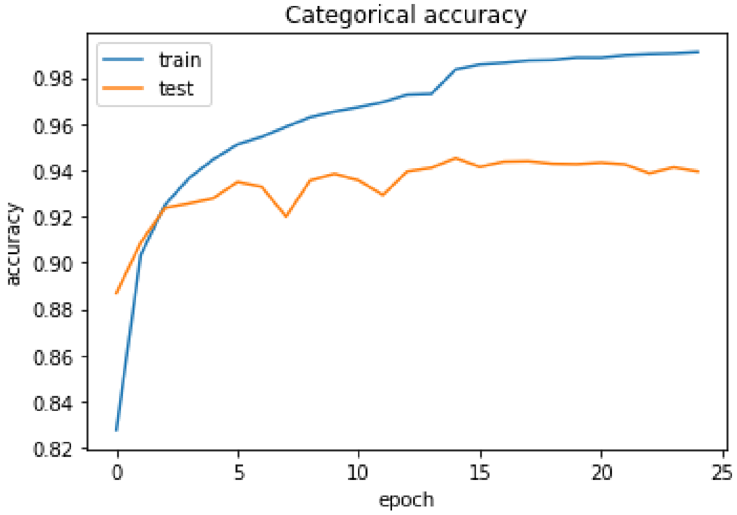


Figure 4.3: Training accuracy

In our case accuracy has converged by 25 epochs.

12.Loss :

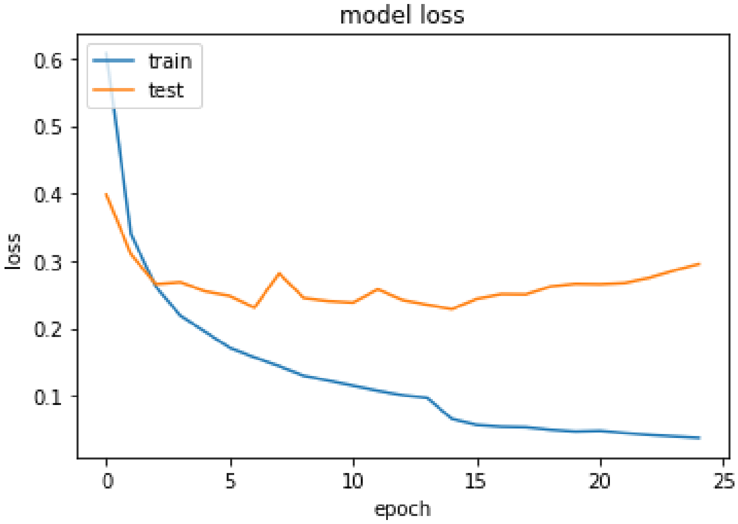


Figure 4.4: Loss

13.We test the trained model on the testing dataset which we initially separated from google speech command data set.

We evaluated the loss and categorical accuracy on these different sets of data and results are as shown :

Train: [0.05517520383000374, 0.9866559505462646] Validation: [0.227983385324478l5, 0.9456390738487244] Test: [0.22142711281776428, 0.9436619877815247]

14.Attention Weights : we use attention weight scores to evaluate the correlation of a certain testing element (an audio input in our case) to each of the output classes.

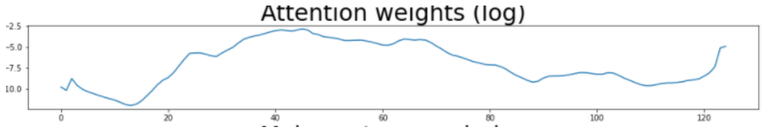


Figure 4.5: Logarithmic values of attention weights

15.Build a vector for predicted values and compare them with expected labels(the words that are actually expected to be in the test audio files) using confusion matrix.

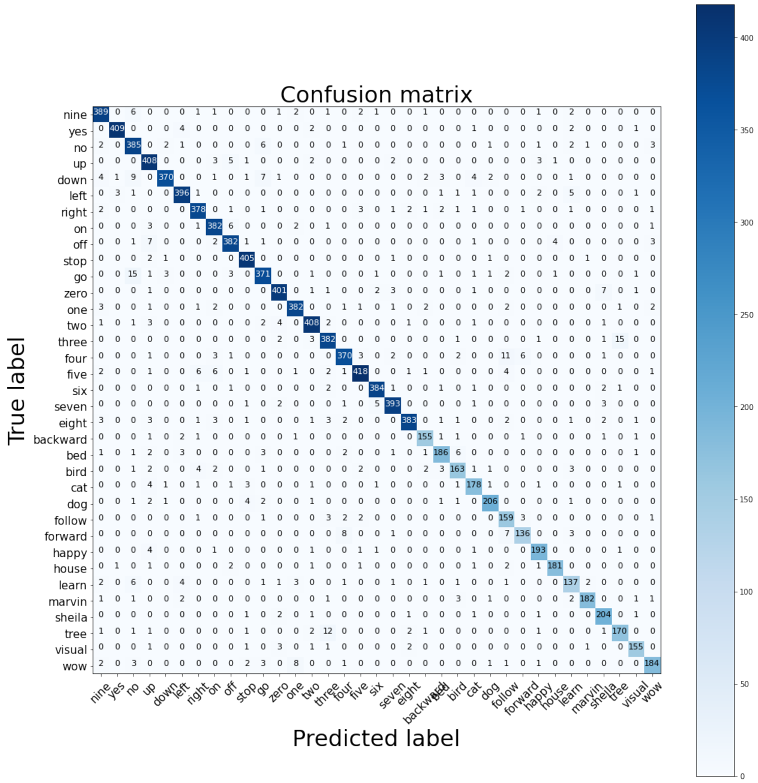


Figure 4.6: Confusion Matrix for predicted values against expected values

Chapter 5

Future Works :

Even though this report covers the the entire architecture of speech recognition, indexing and query retrieval, implementation has been limited to building an efficient speech recognition model only.

This project can be resumed by building the whole search engine system.

Speech diariazation can be added in a multi-model speaker which can make the searches possible while containing the noise and having multiple users at a time

Chapter 6

CONCLUSION

This is an audio based search engine based on indexing and retrieval system for the Web. It incorporates speech recognition technology and therefore is able to operate both with and without transcriptions.

It has given us a unique opportunity to test a spoken data retrieval system on a large scale.

Acceptable retrieval accuracy can be achieved despite high-recognition error rates which shows that indexing of audio documents on the Web is feasible given the current level of recognition.

Chapter 7

References

[1] Van Thong, J.-M., Moreno, P. J., Logan, B., Fidler, B., Maffey, K., Moores, M. (2002). an experimental speech-based search engine for multimedia content on the web. IEEE Transactions on Multimedia, 4(1), 88–96. doi:10.1109/6046.985557

[2] International Journal of Computer Applications (0975 – 8887) Volume 90 – No.3, March 2014 40 Smart Voice Search Engine Shahenda Sarhan Faculty of Computers and Information, Mansoura University Mansoura,Egypt.

[3] Voice Based Search Engine And Web Page Reader. 1,Ummuhanysifa U 2, Nizar Banu P K 1,2,B.S. Abdur Rahman University Chennai.

[4] [Dragon Naturally Speaking. DRAGON Naturally Speaking legal white paper, march 2009.

[5] A neural attention model for speech command recognition, Douglas Coimbra de Andrade, Sabato Leo, Martin Loesener Da Silva Viana, Christoph Bernkopf.

[6] Deep Speech: Scaling up end-to-end speech recognition, Awni Hannun,

Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. NgBaidu, Research – Silicon Valley AI Lab.

[7] Ye, Ruizhi Yang, Yingchun Shan, Zhenyu Liu, Yiyan Zhou, Sen. (2007). ASEKS: A P2P audio search engine based on keyword spotting. 615 - 620. 10.1109/ISM.2006.33.

[8] Cross modal audio search and retrieval with joint embeddings based on text and audio, Benjamin Elizalde, Shuayb Zarar, Bhiksha Raj †Microsoft Research, ?Carnegie Mellon University.

[9] Knees, Peter Pohle, Tim Schedl, Markus Widmer, Gerhard. (2007). A music search engine built upon audio-based and web-based similarity measures. 447-454. 10.1145/1277741.1277818.

[10] Evaluating deep learning architectures for Speech Emotion Recognition Haytham M. Fayeka, Margaret Lecha, Lawrence Cavedonb, School of Engineering, RMIT University, Melbourne VIC 3001, Australia, School of Science, RMIT University, Melbourne VIC 3001, Australia.

[11] J. Huang and B. Kingsbury, ”Audio-visual deep learning for noise robust speech recognition,” 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 2013, pp. 7596-7599, doi: 10.1109/ICASSP.2013.6639140.

[12] Ensemble Deep Learning for Speech Recognition, Li Deng and John C. Platt, Microsoft Research, One Microsoft Way, Redmond, WA, USA.

[13] Audio-visual speech recognition using deep learning, Kuniaki Noda · Yuki Yamaguchi ·Kazuhiro Nakadai · Hiroshi G. Okuno · Tetsuya Ogata.

[14] Taylor Francis, Hidden markov models for speech recognition. infor-

maltd registered in england and wales registered number: 1072954 registered office: mortimer house, 37-41 mortimer street, london w1t 3jh,uk, 30 december 201. michigan state university.

[15] hui jiang li deng gerald penn ossama abdel-hamid, abdel rahman mohamed and dong yu. convolutional neural networks for speech recognition. ieee/acm transactions on audio,speech,and language processing,vol.22,no.10. october , 2014.

[16] Iao1in hu yue zhao, xingyu lin. recurrent convolutional neural network for speech processing.

[17] Gilles boulianne3 luk as burget4 5 ondrej glembek4 nagendra goel 6mirko hannemann4 petr motl i cek7 yanmin qian8 petr schwarz4 jan silovsk y9 georg stemmer10 karel vesel y4 daniel povey1,arnab ghoshal2. algebraic analysis of many valued logics.