

Design and Implementation of Speech to Text Conversion on Raspberry Pi

A. Pardha Saradhi, A. Sai Kiran, A. Dileep Kumar, B. Srinivas, M. V. Nageswara Rao

Abstract: It is proposed to implement a portable speech to text conversion system on a raspberry pi using neural networks and transfer of the predicted text to a remote receiver via simple mail transfer protocols.

Index Terms: Kaggle, softmax function, argmax function, ASR(Automatic Speech Recognition).

I. INTRODUCTION

Auditory impairment has been a known issue to the society. According to WHO, about 5% of the world's population suffer from disabling hearing impairments. Several methods have been introduced over time to establish a communication medium, such as sign language and some of them were proven effective. Recent developments in deep learning allowed us to build effective Automatic Speech Recognition (ASR) systems with minimal human intervention. Raspberry pi is an easily portable, pocket sized computer weighing 45 grams. Raspberry pi's specifications were eventually improved to accommodate computation of much complex tasks. It has an inbuilt GPU, thus facilitating faster parallel processing. The raspbian OS comes with an inbuilt python IDE. Since python is one of the most used and popular languages, porting python projects to pi is a lot easier than doing so to other platforms.

II. LITERATURE SURVEY

J.C. Junqua has proposed their work to gain an understanding about the Lombard effect with the prospect of improving performance of automatic speech recognizer [1]. Wouter Gevaert et al. discussed the implementation of a discrete speech recognizer using multilayer feedforward networks and radial basis function network [2]. Tara N. Sainath proposed the deep convolutional neural networks perform comparatively better against deep neural networks. Its experimental results depict comparative performance improvements of CNNs over DNNs, GMMs/HMMs [3]. R.L.K. Venkateswarulu et al. suggested that recurrent neural networks are better at speech recognition when compared to a network of multi-layer perceptrons [4]. Awni Hannun presents a continuous speech recognition system using recurrent neural networks. Connectionist temporal classification loss function and Nesterov's accelerated gradient method were

used to optimize the model during the training period [5]. Arpita Gupta has proposed supervised and unsupervised approaches at speech recognition using recurrent neural networks with LSTMs and restricted Boltzmann machine respectively in [6]. Manjutha Mhas made a literature review on automated speech recognition. History of ASR, classification and methodologies of speech recognition were discussed in [7]. Ying Zhang has proposed the usage of connectionist temporal classification with CNNs without the inclusion of RNNs. This type of model can be computationally efficient with significant accuracy [8]. Aditya Amberkar has presented a study on RNNs, LSTMs and their performance contributions to speech recognition systems in [9]. Paresh M. Chauhan has presented an in depth study and implementation of MFCC feature extraction in [10] and performed speaker classification using neural networks in noisy environments.

III. PROPOSED METHOD

This paper presents a speech to text conversion system using neural networks with the following methods and algorithms to perform feature extraction, text prediction and model improvisation. It is proposed to port this STT model to a Raspberry pi for the ease of portability to the user. The predicted text is transferred to the user's email address to facilitate ease of usage. Its implementation process is described below in detail.

3.1 MEL FREQUENCY CEPSTRAL CO-EFFICIENT

The audio signal $X(n)$ to be processed is dissected into multiple overlapping frames and power spectral density for each frame is computed. An equally spaced triangular filterbank is designed in mel scale and converted to frequency scale.

To convert from mel to frequency scale:

$$M^{-1}(m) = 700(e^{\frac{m}{1127}} - 1)$$

To convert from frequency to mel scale:

$$M(f) = 1127 \ln\left(1 + \frac{f}{700}\right)$$

$X(n)$

Revised Manuscript Received on April 07, 2019.

A. Pardha Saradhi, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP.

A. Sai Kiran, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP.

A. Dileep Kumar, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP.

B. Srinivas, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP.

Dr. M.V. Nageswara Rao, Professor, Department of ECE, GMR Institute of Technology, Rajam, AP, India.

Design and Implementation of Speech to Text Conversion on Raspberry Pi

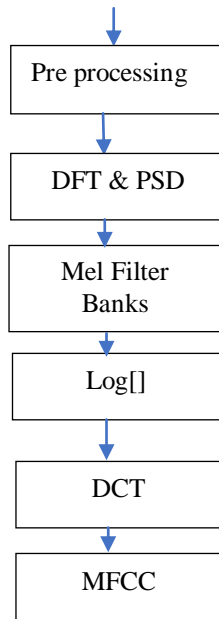


Figure 1: Steps to extract MFCC features

The melfilter bank is applied to the signal power spectrum and resulting energies in each filter are summed up. This energy matrix is applied with logarithm, followed by discrete cosine transform to obtain the mel frequency cepstrum coefficient features of the input audio. Figure 1 depicts a pictorial representation of the feature extraction process.

3.2 CATEGORIAL CROSS ENTROPY

Categorical Cross Entropy is a loss function. It requires one hot encoded sequence of all labels and softmax probabilities of all labels. The loss value is given by the equation below.

$$\text{loss} = - \sum \text{label_onehot} * \log(\text{softmax output})$$

Each label has a loss of " $-\log(\text{softmax output})$ " as only one element in its one-hot is unity and others are zeroes. The respective losses of each label back propagate through the network and manipulate the weights of each neuron in its path in proportion to the learning rate set. This process is iterated for every epoch trained.

3.3 BATCH NORMALIZATION

Some input variables might be proportionately larger or smaller compared to others. This might lead to extremely huge gradients as we traverse across the network. This in turn makes it harder for the network to learn on the given input parameters (dataset) or might even lead to an exploding gradient case. So, the goal is to normalize the input data between the range of 0 and 1 so that, even if the gradients cascade, the difference is comparatively smaller, allowing a smoother training.

$$\text{Normalized param}(x) = \frac{\text{parameter} - \text{mean}}{\text{standard deviation}}$$

3.4 MAX POOLING

To decrease the training load of a network, output of a layer is down sampled. This works as a second filter for

feature extraction. Input matrix of order $a \times b$ is disintegrated into a new $\left(\frac{a}{c}\right) \times \left(\frac{b}{d}\right)$ matrix. The newly formed matrix is the downsampled form of initial matrix. This method reduces the workload, saves the time while still maintaining the composition of the input features.

3.5 ADAM

ADAM is an algorithm used to change weights depending on the observed loss while back propagating through the network. It adjusts the weight values so as to reduce the loss and optimize the model. ADAM adapts its learning rate based on the mean and variance of the gradients and controlling the decay rate of the exponential moving averages.

3.6 SIMPLE MAIL TRANSFER PROTOCOLS

SMTP is a stack of application layer protocols. It is used by a sender to communicate with a remote receiver on a TCP/IP network. After establishing a successful connection, the SMTP server will be live till the connection is terminated. This paper makes use of SMTP to transfer the predicted text to the hearing impaired ones.

IV. ARCHITECTURE

The implementation of Speech to Text Conversion Architecture is shown in fig.2. It consists of four convolutional layers, each followed by a 2d max pooling layer and a dropout of 5% are used. The output is passed to a set of four feed forward layers with 5% dropout. It is followed by four simple recurrent neural network layers with no dropout. The fourth RNN layer's output is passed to a softmax function. Adam optimizer is used with categorical cross entropy loss function to update weights at each instance.

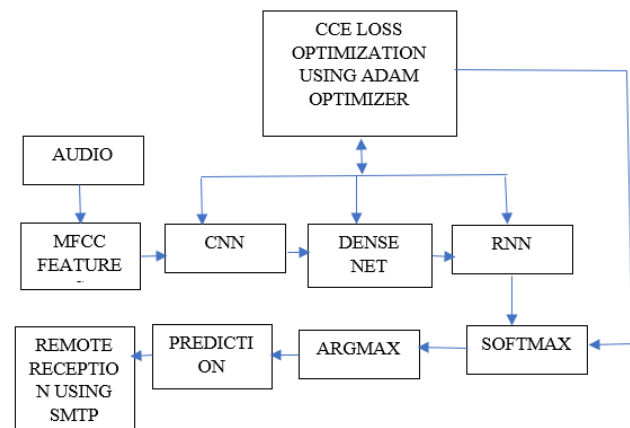


Figure 2: STC Architecture

V. IMPLEMENTATION

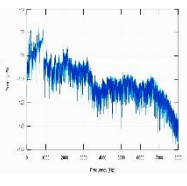
A set of individual audio files consisting of distinct words spoken by various persons is extracted from Kaggle’s datasets and used as the dataset for this implementation. The dataset consists of 35 labels, 105,829 audio samples, split into a train set of 63,497 samples and test set of 42,332 samples and are one hot encoded. Audio samples are compressed to 16KHz. Mel frequency cepstral co-efficient features are extracted, reshaped and saved to .npy files for each word. A convolutional neural network is designed to be trained on the extracted features. This output is fed into a softmax function, followed by an argmax function to predict the appropriate utterance. Keras python library with Tensorflow backend is used for ease of implementation. Google’s SMTP server “smtp.gmail.com” is used to transfer the prediction. Once the server is initialized, user’s Gmail credentials are used to request a server login. The predicted text is embedded into a message variable and mailed to the receiver’s address. The receiver at the other end will be able to view the prediction using Gmail’s GUI.

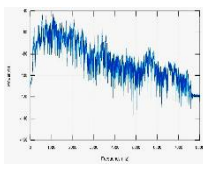
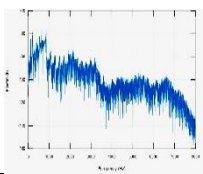
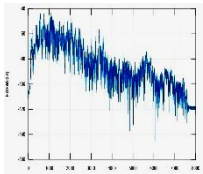
Once the above process is successfully executed on a PC/laptop, it is to be ported to Raspberry pi. A raspberry pi 3, 5V-2A adapter for power supply, RJ45-RJ45 ethernet connector, external memory card, USB microphone are the hardware requirements for this implementation. Download the debian based Raspbian OS from pi’s official website and extract to the memory card used. Insert the memory card into pi and power it up with the adapter. Connect the pi to PC/laptop using the ethernet connector and configure network settings to share internet among both the devices. MobaXterm software is used to simulate pi’s monitor on the PC/laptop screen. Install Librosa (1.6mb), Keras with Tensorflow, Backend (<100mb), Numpy (<15mb) through “pip install” commands in the terminal. Copy the model and associated scripts to pi and execute to record and predict the text.

IV. EXPERIMENTAL RESULTS

The model is trained for 15 epochs with a batch size of 100 units, on a train set of 63,497 samples. This model has an approximate size of 5mb and utilizes RAM of 300mb. After testing on a test set of 42,332 audio samples, a validation accuracy of 81.25% of is obtained. The below table depicts the original audio, it’s visual spectrogram and the predicted text at the receiver’s end,

Table 1.

LABEL NAME	SPECTROGRAM ESTIMATE	RECEIVED TEXT THOUGH SMTP
BED		(no subject) Inbox x project ece <project.25823@gmail.com> to me v bed

DOWN		(no subject) Inbox x project ece <project.25823@gmail.com> to me v down
ONE		(no subject) Inbox x project ece <project.25823@gmail.com> to me v one
EIGHT		(no subject) Inbox x project ece <project.25823@gmail.com> to me v eight

VI. CONCLUSION

In this work an end to end speech to text conversion model using neural networks is implemented. Techniques such as max pooling and batch normalization are used to further optimize the model and boost its accuracy. The process of porting the trained model to a Raspberry pi is explained. The usage of these kind of neural network models is confined to the labels used in the dataset. Better datasets with more labels and inclusion of various accents improve the application efficiency.

REFERENCES

1. J.-C.Junqua. “The Lombard reflex and its role on human listeners and automatic speech recognizers”.Journal of the Acoustical Society of America, 1993.
2. WouterGevaert, GeorgiTsenov, Valeri Mladenov, Senior Member, “Neural Networks used for Speech Recognition”.IEEE, 2010.
3. Tara N. Sainath1, Abdel-rahmanMohamed2, Brian Kingsbury1, Bhuvana Ramabhadran1,“Deep Convolutional Neural Networks for LVSCR”, IEEE, 2013
4. Dr.R.L.K.Venkateswarulu, Dr.R.Vasanthakumari, G.VaniJayasri, “Speech Recognition by Using Recurrent Neural Networks”, International Journal of Scientific & Engineering Research Volume 2, Issue 6, ISSN 2229-551, June-2011.
5. AwniHannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng, “Deep speech: scaling up end-to-end speech recognition”,Baidu Research – Silicon Valley AI Lab, Dec 2014.
6. Arpita Gupta and Akshay Joshi, “Speech Recognitionusing Artificial NeuralNetwork”, IEEE,2018.
7. Manjutha M,Gracy J, Dr P Subashini, Dr M Krishnaveni“Automated Speech Recognition System – A Literature Review”,IJETA-V4I2P9, April 2017.
8. Ying Zhang, Mohammad Pezeshki, Phil’emonBrakel, Saizheng Zhang, C’esar Laurent Yoshua Bengio1, Aaron Courville,“TowardsEnd-to-End Speech Recognition with Deep Convolutional Neural Networks”, IEEE, Jan 2017.
9. Aditya Amberkar, Gaurav Deshmukh, ParikshitAwasarmol, Piyush Dave, “Speech Recognition using RecurrentNeural Networks”, IEEE, 2018.



Design and Implementation of Speech to Text Conversion on Raspberry Pi

10. Paresh M. Chauhan, Nikita P. Desai, "Mel Frequency Cepstral Coefficients (MFCC) based speaker identification in noisy environment using wiener filter", IEEE, March 2014

AUTHORS PROFILE



A. Pardha Saradhi, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP. Interested in signal processing and neural networks



A. Sai Kiran, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP.



A. Dileep Kumar, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP.



B. Srinivas, B.Tech, ECE Department, GMR Institute of Technology, Rajam, AP.



M. V. Nageswara Rao, received Ph.D from Andhra University 2013. Presently, he is professor, Department of ECE, GMR Institute of Technology, Rajam, A.P; India. His research interests are VLSI and signal processing techniques.

’ ’ ’