Combining Data Augmentations for CNN-Based Voice Command Recognition

Arian Azarang, *Student Member, IEEE*, John Hansen, *Fellow, IEEE,* and Nasser Kehtarnavaz, *Fellow, IEEE*

***Abstract*—This paper presents combining two data augmentation methods involving speed perturbation and room impulse response reverberation for the purpose of improving the generalization capability of convolutional neural networks when used for voice command recognition. Speed perturbation generates voice command variations caused by shorter or longer time durations of commands spoken by different speakers. Room impulse response reverberation generates voice command variations caused by reflected sound paths. The combination of these two augmentation methods is presented in this paper by examining a public domain dataset of voice commands. The experimental results based on the performance metric of word error rate indicate the improvement in voice command recognition rates when combining these data augmentation methods relative to using each augmentation method individually.**

Keywords—Combining data augmentation methods for voice command recognition, CNN-based voice command recognition, voice command human interaction systems

I. INTRODUCTION

Many human interaction systems involve voice command recognition. Due to the advances made in speech recognition, a wide range of voice assisted or voice command applications have been commercialized. Examples of such applications are motorized wheelchair control, voice controlled motors, voice activated appliances, and voice controlled smartphone apps [1-4]. Many papers have appeared in the literature on voice command recognition. For example, in [5], Mel Frequency Cepstral Coefficients (MFCCs) together with vector quantization were used to achieve voice command recognition. In [6], an end-to-end voice command recognition system was developed based on so-called capsule networks. In [7], MFCC features and Dynamic Time Warping (DTW) were utilized to carry out voice command recognition. In [8], Recurrent Neural Networks (RNNs) were used to perform voice command recognition.

More recently, due to the effectiveness of Deep Neural Networks (DNNs) for recognition tasks, DNNs and in particular Convolutional Neural Networks (CNNs) have been deployed for voice command recognition, e.g. [9-10]. In [11-12], it has been shown that CNNs provide effective models of the contextual information in the Mel domain. Considering that a considerable amount of voice data is needed in order to train CNNs, data augmentation methods have been considered to increase voice dataset sizes, e.g. [13-15]. More specifically, two data augmentation methods were presented in [16-17] for voice command recognition based on speed perturbation and Room Impulse Response (RIR) reverberation. In this paper, for convolutional neural network-based voice command recognition, the combination of these data augmentation methods is used and the outcome is compared with using each augmentation method individually.

The rest of the paper is organized as follows. In section II, the two previously used augmentation methods are briefly mentioned. The CNN architecture used for voice command recognition is presented in section III. Section IV describes the public domain dataset considered. The experimental results and their discussion are then stated in section V followed by the conclusion in section VI.

II. AUGMENTATION METHODS

A. Room Impulse Response (RIR)

The persistence of sound signals in a cavity like a room is called reverberation. In fact, reverberation is caused by many sound reflections. Mathematically, reverberation is modeled via the convolution of the original signal with RIR. As discussed in [18], for an original sound signal $s(t)$, the reverberation signal $x(t)$ can be expressed as

$$x(t) \approx \sum_{\tau_i=0}^{n} s(t-\tau_i) + n(t)$$

where τ_i s denote the delays of reflected signals, $i = 0$ corresponds to the direct signal path, and $n(t)$ represents environmental noise.

RIR is computed based on the Reverberation Time (RT) parameter. This time is defined as the time it takes for the sound intensity to drop 60 dB below the direct path sound, and is empirically expressed as follows [18]:

$$RT_{60} = \frac{24\ln(10)\,V}{c\sum_{i=1}^{6} A_i(1-\beta_i^2)}$$

where V denotes the room volume, A_i the surface area of the i^{th} wall, c the sound velocity, and β_i the reflection coefficient

Arian Azarang, John Hansen and Nasser Kehtarnavaz are with Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080, USA; email addresses: {azarang, john.hansen, kehtar}@utdallas.edu.

978-1-7281-3980-7/19/$31.00 ©2019 IEEE

of the i^{th} wall. Fig. 1 shows the effect of a reverberation time on two different command signals.

B. Speed Perturbation

In [17], speed perturbation was used to achieve data augmentation. This method involves expanding and compressing a command signal $x(t)$ by a factor α, i.e. $x(\alpha t)$. Speed perturbation produces a warped version of the command signal. The Fourier transform of the warped signal is given by $\alpha^{-1}X(\alpha^{-1}\omega)$ with ω denoting frequency. Fig. 1 also shows the effect of speed perturbation on the two sample command signals. Basically, the speed perturbation augmentation takes into consideration speed variations in the way speakers utter command words.

III. CONVOLUTIONAL NEURAL NETWORK USED

There are many types of deep neural networks. Convolutional Neural Networks (CNNs) [9] are most widely used. These networks are capable of learning voice commands in the presence of variability of voice command utterances. The input to the CNN used here is considered to be the commonly used Mel-frequency spectrogram derived from MFCCs.

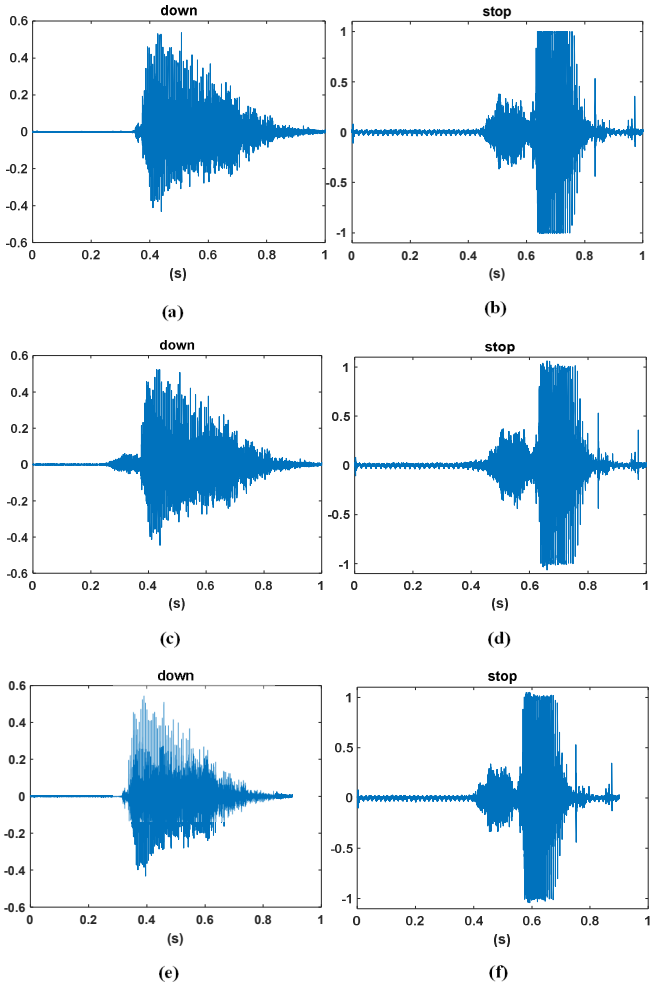

Fig 1. (a)-(b) Two original signals, (c)-(d) versions for a reverberation time of 0.4s, and (e)-(f) versions for a resampling rate of 0.9.

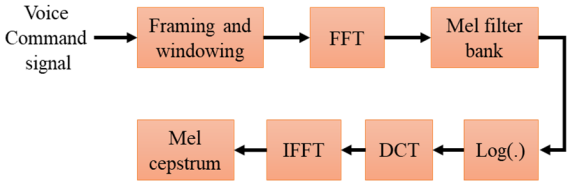

Fig 2. Steps to generate Mel-frequency spectrogram as the input to CNN.

Fig. 2 shows the steps involved in computing the Mel-frequency spectrogram and Fig. 3 shows a sample spectrogram. Since the performance of CNNs is highly dependent on the network architecture, different CNN architectures were examined and the CNN architecture that was found to be the most effective is indicated in Table 1. This architecture incorporates the following three types of processing element layers: convolution layer, max-pooling layer, and softmax layer. Readers are referred to [13] for details on these layers and CNN training.

TABLE 1. CNN ARCHITECTURE USED

Layers	Number of Filters	Description
Input Layer	---	(40, 98)
Conv1	12	(3,3) with (1,1) stride
Max-pooling1	---	(3,3) with (2,2)
Conv2	24	(3,3) with (1,1) stride
Max-pooling2	---	(3,3) with (2,2)
Conv3	48	(3,3) with (1,1) stride
Max-pooling3	---	(3,3) with (2,2)
Conv4	48	(3,3) with (1,1) stride
Max-pooling4	---	(3,3) with (2,2)
Conv5	48	(3,3) with (1,1) stride
Max-pooling5	---	(3,3) with (1,1)
Fully Connected	12	Connected to Softmax

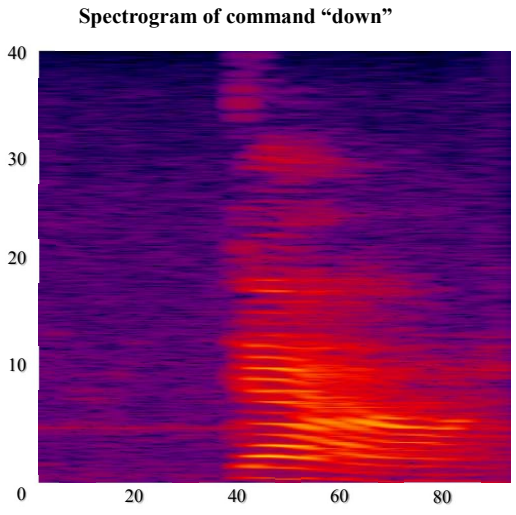

Fig 3. Sample Mel-frequency spectrogram as the input to CNN.

IV. DATASET

The experiments reported in the next section were conducted by considering the public domain Google voice command dataset [19]. This dataset consists of 105,000 audio file utterances of thirty different command words. Table 2 lists the number of different utterances for each of the command words. In addition to the voice command files, background noise or pure noise files are also included in this dataset. 10 words in this dataset were considered to be the Words or commands of Interest (WoI), which are {yes, no, up, down, left, right, on, off, stop, go}, and the remaining 20 words were considered to be the Words of Non-interest (WoN).

TABLE 2. NUMBER OF UTTERANCES FOR COMMAND WORDS IN THE GOOGLE DATASET [19].

Command	Number of Utterances
Bed	2,014
Bird	2,064
Cat	2,031
Dog	2,128
Down	3,917
Eight	3,787
Five	4,052
Follow	1,579
Forward	1,557
Four	3,787
Go	3,880
Happy	2,054
House	2,113
Learn	1,575
Left	3,801
Nine	3,934
No	3,941
Off	3,745
On	3,845
One	3.890
Right	3,778
Seven	3,998
Six	3,860
Stop	3,872
Three	3,727
Tree	1,759
Two	3,880
Up	3,723
Yes	4,044
Zero	4,052

V. EXPERIMENTAL RESULTS AND DISCUSSION

The spectrograms of the speed perturbed or/and reverberated signals were added to augment the spectrograms of the original signals. Then, both the original and the augmented datasets were randomly divided into three non-overlapping sets: training set (70%), evaluation set (10%) and test set (20%).

To evaluate the performance without and with augmentation, the Word Error Rate (WER) performance metric described by the following equation was computed [20]

$$WER = \frac{S + I + D}{N}$$

where *S, I, D,* and *N* indicate the numbers of substitutions, insertions, deletions, and in reference, respectively. The following four scenarios were examined: without augmentation, with augmentation using speed perturbation, with augmentation using reverberation, and a combination of the two augmentation methods. As noted in Tables 3 and 4, four different experiments or augmentations were done by changing the parameter values for the two augmentation methods. In case of speed perturbation, the resampling rates were considered in this range [0.9, 1.1] as in this range the perturbed signals still remained intelligible. In case of reverberation, the reverberation time was selected in this range [0.4, 0.9] as suggested in [16].

The average WERs across all the utterances for the test set are shown in the bar chart shown in Fig. 4 for different augmentation folds. As can be seen from this figure, both of the augmentation methods achieved lower WERs as compared to the scenario with no augmentation. Furthermore, the combined augmentation proposed in this paper outperformed each of the two previously proposed augmentation methods. As can be seen in this figure, the number of augmentation folds lowered the rates till 8 folds. Further increases in the augmentation folds beyond 8 folds had no or negligible impact on the WERs.

Table 5 shows the overall rates corresponding to an augmentation by 8 folds. As can be observed from this table, the combination of the speed perturbation and reverberation augmentations generated the lowest overall error. The confusion matrices for the no augmentation case and the 8-fold augmentation cases for individual and combined augmentations are shown in Figs. 5 through 8.

TABLE 3. SPEED PERTURBATION AUGMENTATION VERSIONS

Experiment Number	Resampling Rates
Experiment 1	0.9, 1.1
Experiment 2	0.9, 0.95, 1.05, 1.1
Experiment 3	0.9, 0.93, 0.95, 1.05, 1.07, 1.1
Experiment 4	0.9, 0.93, 0.95, 0.97, 1.03, 1.05, 1.07, 1.1

TABLE 4. REVERBERATION AUGMENTATION VERSIONS

Experiment Number	Reverberation Times (sec)
Experiment 1	0.4, 0.7
Experiment 2	0.4, 0.5, 0.6, 0.7
Experiment 3	0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Experiment 4	0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1

TABLE 5. EFFECT OF AUGMENTATION ON WORD RECOGNITION RATE

Augmentation Method	Overall WER (%)
Without augmentation	4.32
With speed perturbation augmentation only (by 8 folds)	3.05
With reverberation augmentation only (by 8 folds)	3.48
Combined augmentation (by 8 folds)	**2.67**

VI. CONCLUSION

In this paper, the data augmentation methods of speed perturbation and room impulse response reverberation have been combined in order to improve the generalization capability of convolutional neural networks for recognizing voice commands. A public domain dataset of voice command words made publicly available by Google was considered to examine the impact of the combined augmentation on the recognition error rate. The experiments conducted have indicated the proposed combined augmentation method is more effective than each of the previous augmentation methods when used individually.

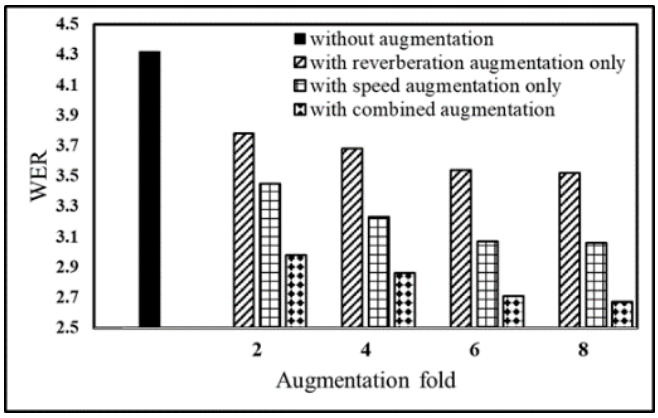

Fig. 4. WERs of the augmentation methods.

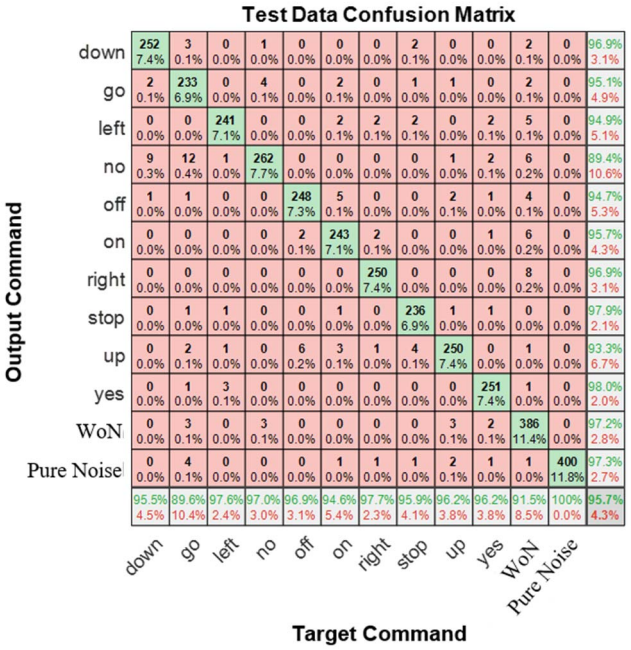

Fig. 5. Confusion matrix without augmentation.

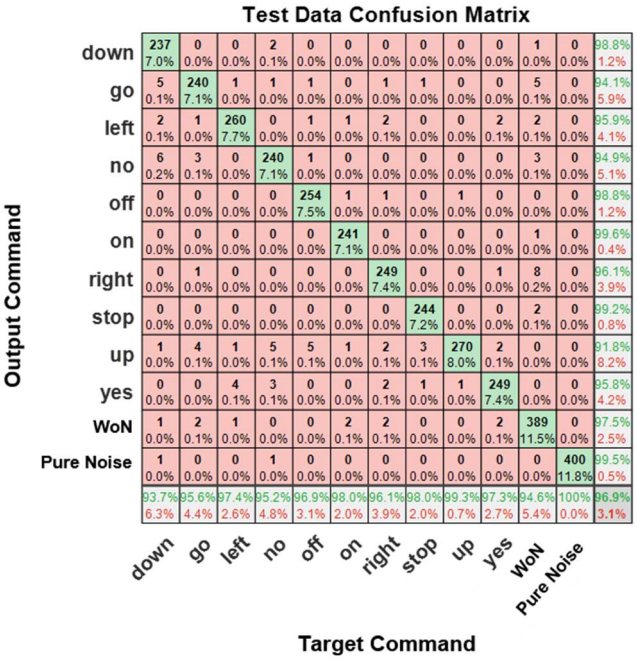

Fig. 6. Confusion matrix using speed perturbation augmentation.

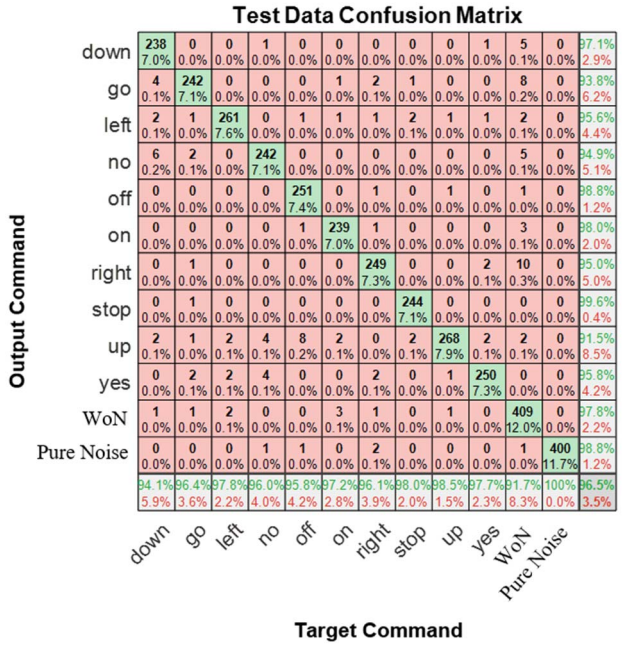

Fig. 7. Confusion matrix using reverberation augmentation.

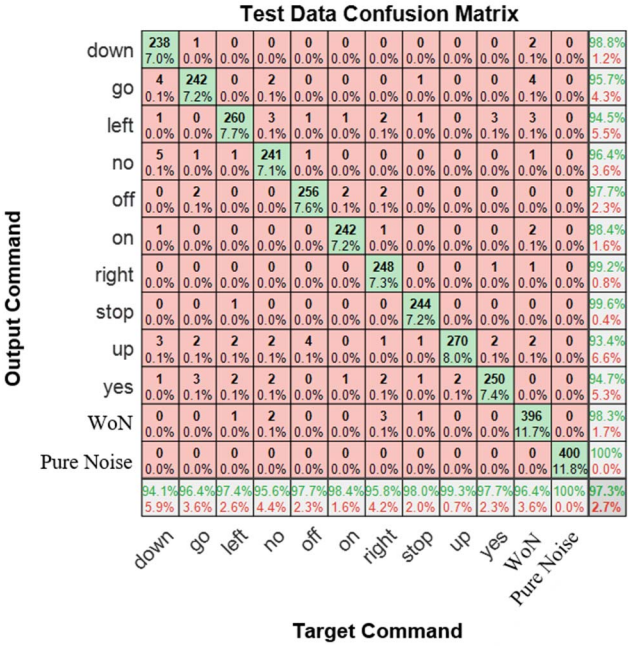

Fig. 8. Confusion matrix using combined augmentation.

VII. REFERENCES

[1] J. Kurtzberg, and J. Lew, "Voice-controlled motorized wheelchair with sensors and displays," US patent no. 6,108,592, 2000.

[2] G. Azam, and M. Islam, "Design and fabrication of a voice controlled wheelchair for physically disabled people," *Proceedings of International Conference on Physics Sustainable Development & Technology (ICPSDT)*, pp. 81-90, August 2015.

[3] C. D'Souza, C. D'Souza, S. D'Souza, and R. Rodrigues, "Voice operated control of a motor using LabVIEW," *Electrical and Electronic Engineering*, vol. 7, no. 2, pp. 60-64, 2017.

[4] A. Wilde, O. Ojuroye, and R. Torah, "Prototyping a voice-controlled smart home hub wirelessly integrated with a wearable device," *Proceedings of IEEE 9th International Conference on Sensing Technology (ICST),* pp. 71-75, December 2015.

[5] M. Shaneh, and A. Taheri, "Voice command recognition system based on MFCC and VQ algorithms," *World Academy of Science, Engineering and Technology*, vol. 57, pp. 534-538, 2009.

[6] J. Bae, and D. Kim, "End-to-end speech command recognition with capsule network," *Proceedings of Interspeech*, pp. 776-780, September 2018.

[7] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using Mel frequency cepstral coefficient and dynamic time warping techniques," *arXiv:1003.4083*, 2010.

[8] A. Corradini, and P. Cohen, "Multimodal speech-gesture interface for handfree painting on a virtual paper using partial recurrent neural networks as gesture recognizer," *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, August 2002.

[9] P. Vecchiotti, F. Vesperini, E. Principi, S. Squartini, and F. Piazza, "Convolutional neural networks with 3-d kernels for voice activity detection in a multiroom environment," *Multidisciplinary Approaches to Neural Computing*, vol. 69, pp. 161-170, 2018.

[10] H. Bae, H. Lee, and S. Lee, "Voice recognition based on adaptive MFCC and deep learning," *Proceedings of 11th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1542-1546, June 2016.

[11] W. Liu, W. Li, L. Sun, L. Zhang, and P. Chen, "Finger vein recognition based on deep learning," *Proceedings of 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 205-210, June 2017.

[12] H. Liang, X. Lin, Q. Zhang, and X. Kang, "Recognition of spoofed voice using convolutional neural networks," *Proceedings of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 293-297, November 2017.

[13] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep convolutional neural network acoustic modeling," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 4545-4549, April 2015.

[14] W. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," *arXiv:1707.06265,* 2017.

[15] J. Salamon, and J. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279-283, 2017.

[16] T. Ko, V. Peddinti, D. Povey, M. Seltzer, S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220-5224, March 2017.

[17] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," *Proceedings of Sixteenth Annual Conference of the International Speech Communication Association*, pp. 3586-3589, September 2015.

[18] E. Habets, "Room impulse response generator," Eindhoven University of Technology Technical Report, vol. 4, pp. 1-21, 2006.

[19] P. Warden, "Speech Commands: A dataset for limited-vocabulary speech recognition," *arXiv:1804.03209*, 2018.

[20] S. Murthy, D. Sitaram, and S. Sitaram, "Effect of TTS generated audio on OOV detection and word error rate in ASR for low-resource languages," *Proceedings of Interspeech*, pp. 1026-1030, September 2018.