INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

A Voice-Activated Switch for Persons with Motor and Speech Impairments: Isolated-Vowel Spotting Using Neural Networks

Shanqing Cai[1], *Lisie Lillianfeld*[1], *Katie Seaver*[1], *Jordan R. Green*[1,2,3], *Michael P. Brenner*[1,3], *Philip C. Nelson*[1], *D. Sculley*[1]

[1]Google Research, USA
[2]MGH Institute of Health Professions, USA
[3]Harvard University, USA

{cais, lisie, mbrenner, pqnelson, dsculley}@google.com, jgreen2@mghihp.edu

Abstract

Severe speech impairments limit the precision and range of producible speech sounds. As a result, generic automatic speech recognition (ASR) and keyword spotting (KWS) systems fail to accurately recognize the utterances produced by individuals with severe speech impairments. This paper describes an approach in a simple speech sound, namely isolated open vowel (/**a**/), is used in lieu of more motorically-demanding utterances. A neural network (NN) is trained to detect the isolated open vowel uttered by impaired speakers. The NN is trained with a two-phase approach. The pre-training phase uses samples from unimpaired speakers along with samples of background noises and unrelated speech; then the fine-tuning phase uses samples of vowel samples collected from individuals with speech impairments. This model can be built into an experimental mobile app to act as a switch that allows users to activate preconfigured actions such as alerting caregivers. Preliminary user testing indicates the vowel spotter has the potential to be a useful and flexible emergency communication channel for motor- and speech-impaired individuals.

Index Terms: Speech impairment, keyword spotting, augmentative and alternative communication

1. Introduction

Individuals with severe speech and motor impairments have limited options for communicating and accessing life-improving technologies. Although automatic speech recognition (ASR) technologies would be of great use to these individuals, it is not a feasible option when speech is limited to the production of only a few vocalizations [1]. Assistive devices that require physical activation may also be unsatisfactory when movement is severely restricted, e.g., during the advanced stages of neurological diseases such as amyotrophic lateral sclerosis (ALS). In these cases, alternative modes of switch activation have been explored, such as camera- and electromyography-based approaches [2-3], as well as approaches based on non-speech vocalization [4-6].

Voice-activated interfaces that recognize vocalizations from a distance do not occupy the eyes or hands and are, therefore, attractive for individuals with limited speech, mobility, and dexterity [7-8]. Recent advances in speech technology have made voice-activated interfaces more ubiquitous. Users' utterances of keywords such as "OK Google" are detected by continuously-running detectors against background noise [9-11]. In this study, we describe a speaker-independent, voice-activated switch that is triggered by the isolated open vowel /**a**/. We test the efficacy of the model for activating call-for-help signals. The open vowel (/**a**/ in phonological notation) sound is chosen for its motoric simplicity. /**a**/ is easy to produce through simple jaw opening movements with minimal demands on other articulators such as the tongue, lips, and velum. Previous studies showed dysarthric speakers produce open vowels with greater precision than mid and closed vowels such as /**i**/ [12-13]. A final reason for choosing the vowel /**a**/ is its relatively high sound intensity among vowels and voiced consonants (e.g., /**i**/ and /**m**/), which leads to a higher signal-to-noise ratio and benefits the quality of signal detectors.

Despite the simplicity of /**a**/, we cannot use a detector trained on the samples from unimpaired speakers directly on impaired speakers, due to the phonatory abnormalities often present in impaired speakers (e.g, [14]). We describe the approach in which an NN is trained to achieve signal-detection qualities for impaired voice inputs when faced with a limited amount of data samples from the target speaker population.

Compared to studies aimed at improving the accuracy of ASR systems on impaired speakers (e.g., [15]), the current study explores the alternative approach of using a less motorically demanding target utterance, in order to expand the accessibility of the resultant communication aid to a wider range of speech-impaired users. Compared to vocalization detectors that require sensors placed on the user's body [5-6], the vowel detector described in the current study is both more convenient in that it requires no special sensor placement or user adaptation and more accessible due to its use of the commodity smart-phone platform.

2. Methods

2.1. Two-phase training and dataset

KWS models for typical speech are trained with a large number of positive examples (on the order of 10k-100k in [9-11]). Due to the difficulty in recording from a larger number of impaired speakers, the number of positive examples (/**a**/) available in this study is much smaller (<1k; see below). To train a high-quality detector for the impaired /**a**/ sound, we use a two-phase approach wherein the first phase trains a base model on a larger corpus of positive examples from typical

http://dx.doi.org/10.21437/Interspeech.2021-330

speakers and the second phase fine-tunes the base model on a smaller corpus of positive examples from impaired speakers. This approach is similar to using a pre-trained model to generate embedding vectors for sound, which has been shown to be more data-efficient than starting from scratch in training custom keyword classifiers [16].

As Table 1 shows, the pre-training phase uses the words “on” and “up” from the Speech Commands dataset [17]. These words are selected because they are the most similar to **/a/** in the Speech Commands dataset. In addition, we use 30 text-to-speech voices from five English dialects and both genders implemented with Tacotron [18] to synthesize samples of monosyllabic sounds similar to /**a**/. Including these synthesized samples significantly improves the validation accuracy on the impaired speech samples after fine-tuning. Using synthesized speech samples is not novel in the area of custom KWS models [16]. Both types of positive examples are augmented through mixing with noise examples at signal-to-noise ratios of 30, 20, and 10 dB, in addition to pitch shifting in the range of -2.5 to +2.5 semitones using the librosa library [21]. This leads to a total of 6.69 hours of positive examples for the pre-training phase.

Table 1: *Composition of the data used for pre-training.*

Dataset name	**Description**	**Amount of data train : validation : test (hours)**
Positive subset		
Speech Commands [17] subset	Words “on” and “up”	4.52† : 0.17 : 0.19
Synthesized speech	Words “ah”, “uh”, “um”, “I” and “R” of 30 different speech synthesizer voices	2.17† : 0.07 : 0.064
Positive subtotal		**6.69 : 0.24 : 0.25**
Negative subset		
LibriSpeech	Read speech [19]	40.0 : 5.0 : 5.0
Chime6	Conversation (“dinner party scenario”) [20]	13.1 : 1.6 : 1.6
Speech Commands subset	Other words from Speech Commands [17] (i.e., not “on” and “up”)	19.0 : 2.4 : 2.3
Cafeteria noises	Cafeteria noises	5.5 : 0.68 : 0.68
Other noises	Background noises	10.5 : 1.3 : 1.3
Music	Non-vocal music audio of various genres	34.0 : 2.0 : 1.4
YouTube audio	Non-speech audio tracks of various YouTube videos	209.2 : 12.3 : 10.8
Negative subtotal		**331 : 25.3 : 23.1**

†: Includes augmented data

The composition of the negative dataset for pre-training is shown in the lower section of Table 1. It includes excerpts of fluent, read speech from the LibriSpeech dataset [19] and casual, spontaneous speech from the Chime6 dataset [20]. No effort is made to filter sounds that resemble “ah” from these negative speech samples. This is based on the rationale that the trained NN should learn to ignore such sounds in connected speech by using the surrounding acoustic context. In addition, the remaining words from the Speech Commands dataset are included in the speech subset of the negative set. The non-speech examples of the negative dataset consists of excerpts from five different datasets of environment sounds and other types of non-speech sounds. These include recordings made in noisy cafeterias, indoor and outdoor environments, music sounds of various genres, and non-speech audio tracks of 200+ hours of YouTube videos. In addition to being used for pre-training, this negative dataset is also joined with the dataset from atypical speakers to subserve the fine-tuning phase.

As shown in Table 2, the fine-tuning phase uses a dataset of the 717 examples (0.20 hours) of the /**a**/ vowel and similar isolated monosyllabic utterances containing open vowels (/**ʌ**/, /**ʌm**/, /**ʌp**/, /**ɑ~**/) collected from 138 adults with speech impairments (age: 49±12; gender: 45F, 92M, 1 unknown) in collaboration with Project Euphonia. Data is collected only after explicit consent has been granted by individuals to provide speech samples for the purpose of research and improving speech-related technologies. We provide users with clear information on the purpose of the data collection and scope of research. The severity of the speech impairments of these 138 individuals are rated by three certified speech-language pathologists (SLPs). Two of the speakers are rated as profound, 47 severe, 45 moderate, 38 mild, and the remaining six close-to-typical. Hence a majority of these speakers (68.1%) have a severity rating of moderate or higher.

Table 2: *The underlying diseases of the impaired speakers used for fine-tuning.*

Underlying disease	**# of speakers (vowel samples)**	**Underlying disease**	**# of speakers (vowel samples)**
ALS	48 (292)	Cerebral palsy	28 (175)
Parkinson’s disease	16 (18)	Down syndrome	14 (17)
Speech in hearing loss	6 (37)	Muscular dystrophy	4 (40)
Frederich's Ataxia	4 (27)	Vocal fold paralysis	3 (12)
Spinal muscular atrophy	2 (20)	Cleft palate	2 (14)
Stroke	2 (7)	Ataxia telangiectasia	1 (18)
Primary lateral sclerosis	1 (12)	Childhood apraxia of speech	1 (12)
Brain tumor	1 (11)	Vagal nerve stimulator usage	1 (1)
Idiopathic familial torsion dystonia	1 (1)	Multiple sclerosis	1 (1)
Traumatic brain injury	1 (1)	Neuromuscular disorder (unclassified)	1 (1)
		Total	**138 (717)**

No additive-noise or pitch-shift augmentation is performed on this relatively small positive set. To compensate for the approximately 10x smaller amount of positive data in the fine-tuning phrase relative to that in the pre-training phrase, the positive examples from impaired speakers are oversampled by a factor of 10 in the fine-tuning phrase. All training examples are cropped into non-overlapping 1-second snippets consistent with the Speech Commands dataset [17]. The 1-second audio snippets are converted to a log-Mel spectrogram with 43 temporal bins (46.4-ms wide Blackman window with 23.2-ms stride) and 80 Mel frequency bins covering 20 - 5000 Hz. This time window, which is slightly longer than what is used in typical KWS systems [9-11], is selected based on the static articulatory gesture and the resulting slow-varying spectrum of the target open vowels. Each log-Mel spectrogram is normalized to a mean of 0 and standard deviation of 1.

2.2. Neural-Network Architecture and Training

Figure 1 is a schematic representation of the convolutional NN (CNN) trained to classify the 43x80-shaped spectrogram inputs. The convolutional architecture has been used successfully for KWS [11] and audio event detection [30]. The model is a small CNN containing four 2D convolution layers and associated max pooling layers as the feature extractor at the bottom and a multilayer perceptron (MLP) at the top for binary classification. Batch normalization [22] and dropout layers [23] are included in the CNN to counter overfitting. The 2D convolution and Dense layers use the exponential linear units (elu) activation [24]. The model is written in TensorFlow 2's Keras API [25-26]. The training of the model uses TPUv2 2x2 configuration based on synchronous gradient updates [27] under the ADAM optimizer [28] with an initial learning rate of 5e-4 and a decay of 0.95 every 5000 steps.

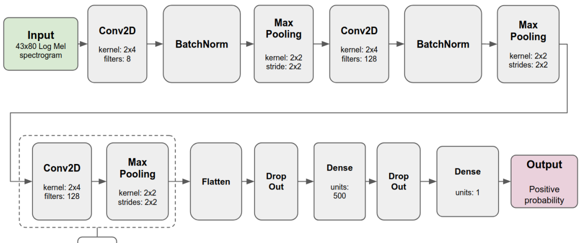

Figure 1: *The architecture of the convolutional NN.*

The objective function of training is the binary-cross-entropy loss calculated on the predicted and actual labels. However, the datasets in both phases of training are highly imbalanced, with a negative-to-positive ratio of 50 or greater, which makes unweighted binary-cross-entropy loss a poor metric of the actual signal-detection quality of the model. Therefore, we measure the accuracy of the model by using false positive rate (FPR) at a false rejection rate (FRR) of 0.1. The operating point of FRR=0.1 is chosen based on testing with testers with motor speech disorders. For reporting, we convert FPR to false positives per hour (FPPH) to better reflect the real-world performance of the model. Google Vizier [29] is used to tune the hyperparameter of the base model based on the FPPH metric. The tuned hyperparameters include the numbers of filters in the Conv2D layers, the size of the hidden Dense layer, dropout rates, the type of nonlinear activation, and the initial value and decay rate of the learning rate. For each set of hyperparameters, a total of 50 training epochs is performed. At the end of the 50 epochs, the epoch with the lowest (best) FPPH metric is chosen. The final model has 528k weights.

Due to scarcity of the positive dataset in the fine-tuning phase, we join the positive examples from impaired speakers with the negative data and randomly split the data into 4 folds for cross-validation. The folds contain non-overlapping sets of speakers. Hyperparameters including the fine-tuning initial learning rate and learning-rate decay are tuned with Vizier on the cross-validation sets. The result of the tuning is then evaluated on the negative test set joined with the positive folds. The resulting ROCs from these test folds are averaged to give rise to the final results (Figures 2-3).

2.3. On-device Inference and Android App

In order to turn the NN-based vowel detector into a useful call-for-assistance mechanism for users with disabilities, we create a cost-effective and portable solution based on an Android app. Once set up by a caregiver, the app runs continuously so the user can use it independently. The trained model is converted to the TFLite format and performs on-device inference at a rate of 4 Hz.

Each detections of the positive event (/**a**/) triggers a preconfigured action such as messaging a caregiver to call for attention. To reduce the effective false-positive rate, the app can be configured to activate actions only when two positive events happen within a 10-second time window. Quantitative testing results and qualitative feedback is obtained from held-out test speakers under their informed consent.

3. Results

3.1. Signal-Detection Quality

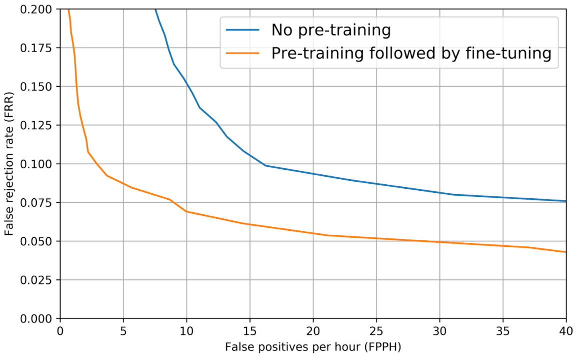

Figure 2: *Test ROC curves for detecting* /**a**/ *produced by impaired speakers, with and without pre-training on data from typical speakers.*

Compared with training the /**a**/ detector from scratch on the relatively small number of positive examples from impaired speakers, performing training from a base model pre-trained on a larger corpus of positive /**a**/-like examples from typical speakers leads to substantial improvement. As Figure 2 shows, the FPPH at the operating point of FRR=0.1 from pre-training followed by fine-tuning is 2.9, which is 5.5x lower (better) compared to training from scratch (16.0). This underscores the importance of starting from properly initialized weights through pre-trained checkpoint under the scenario of limited

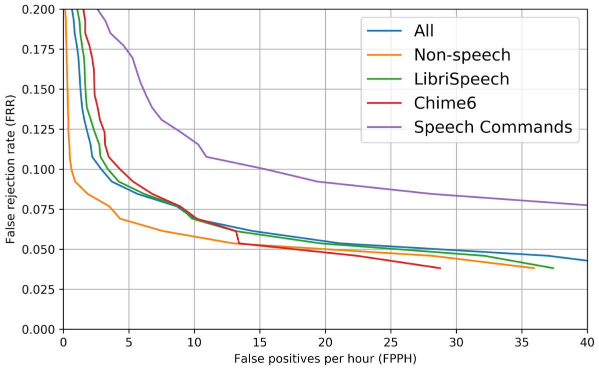

Figure 3: *ROC curves for the trained /ɑ/ detector evaluated on different negative subsets. The non-speech subset includes all categories without discernible speech in Table 1 (combined category of cafeteria, sound of silence, music, and YouTube audio.) The other subsets include different styles of speech, including LibriSpeech, Chime6 and Speech Commands. Each curve is an average from the 4-fold cross-validation.*

training data, an observation consistent with previous results of training custom KWS models [16].

To analyze the false-positive rates under different types of acoustic background, we plot the ROCs curves for different subsets of the negative examples in Figure 3. When only the non-speech negative examples (cafeteria noise, environment sounds, music, and YouTube audio tracks) are evaluated, the FPPH is substantially lower (0.61) compared to the FPPH computed on all negative data (2.9). Among the three speech-type negative subsets, the LibriSpeech, i.e, fluent read speech shows the least confusability with the positive examples (FPPH=3.4). In comparison, Chime6 (casual speech in household settings) and Speech Commands (short isolated words) show much greater confusability. Speech Commands shows the highest FPPH (15.4). Error analysis indicates that isolated monosyllabic words with vowels that resemble **/ɑ/** from the Speech Commands dataset (e.g., "house" and "wow") contributed the highest portion of the false positives.

3.2. On-Device Inference and User Testing

After converting to the TFLite format, the model runs at a sufficiently short latency (mean±SD: 26.4±0.1 ms) for real-time inference on a Pixel 4a phone. Five adults (four with ALS and one with CP, 1F4M) who had moderate to severe dysarthria but retained the ability to produce the vowel **/ɑ/** participated in the testing sessions. The data from these test speakers was not used in the training or tuning of the model. Each participant is introduced to the concept of the **/ɑ/** detector and its potential uses. The experimenter performed a quick demonstration of the **/ɑ/** sound (<2 minutes). Then each participant practiced the **/ɑ/** sound a few times. Then 20-30 utterances of the **/ɑ/** vowel are collected with the phone placed 0.5 m in front of the user. The FRR of the model on the test data from these speakers had a median of 0.103 (mean±SD: 0.168±0.065) at the operating point of FRR=0.1 calibrated on the validation data. Each testing session lasted for 20-30 minutes, during which a small number of false positives occurred due to the experimenter uttering filler words such as "uh". No other background noises triggered false positives during the test sessions.

The use case of the app is to activate call-for-help signals through text messages or HTTP requests to an in-house home-automation server. Due to the relatively high rate of false positives, the app runs under a mode that requires confirmation through repeated activation of the model. Specifically, the user must use the **/ɑ/** sounds to trigger two positive detections by the model within a time window of 10 seconds in order to activate the output action. A detection without an ensuing detection within 10 s was ignored. All five test participants are able to activate the output actions through repetition successfully. One of the test participants is currently using this app as a call-for-help mechanism daily on a voluntary basis at the time of this writing and has been doing so over three months.

4. Discussion

We trained a CNN to detect isolated open vowels produced by individuals with speech impairments. We are able to improve the signal-detection quality of our model through pre-training on a large corpus of typical voices followed by fine-tuning on impaired samples. Like typical KWS systems, this model has the benefit of speaker independence, working for users without requiring further tuning. User testing in a real-world setting demonstrates that this vowel detector enables communication such as calling for attention from caregivers. Compared with camera-based devices (e.g., [2]), such a voice-triggered device is less prone to misplacement and less sensitive to lighting conditions. For the call-for-assistance use case, this is particularly important at nighttime.

The signal-detection specificity of the model is 1-2 orders of magnitude lower compared to KWS systems trained on large corpora of typical speech [9-11]. The reason for this difference is two-fold. First, in order to reduce the difficulty of articulation, we choose a sound less complex and less specific compared to typical keywords (e.g., "OK Google"). These open vowels occur in natural conversational speech as filler words. This is a tradeoff we made for the accessibility of such a detection system at the cost of specificity. Second, compared with the data used to train KWS systems for typical speech, we have two orders of magnitude less training data. Thus, it is not meaningful to directly compare the signal-detection quality of our open-vowel detector and typical KWS sytems.

Future directions include training detectors for other simple sounds producible by severely-impaired speakers, such as **/m/**. A multi-class vowel detector can enable use cases such as coding different actions using sound sequences. Individuals with certain conditions often rely on assisted breathing devices such as BiPap, which significantly alters voice quality. Training a model tolerant to such alterations will make the system accessible to these individuals. In the meantime, it is also worth exploring the adaptation of KWS with typical keywords such as "OK Google" for less-impaired speakers.

5. Acknowledgements

Leonard Florence Center for Living assisted with tester recruiting. Pan-Pan Jiang, Marilyn Ladewig, and Lizzie Docel assisted with data collection. Kevin Wilson provided helpful comments on the manuscript. Jordan Green was partially supported by NIH-NIDCD under Grants K24DC016312.

6. References

[1] J. R. Duffy. Motor Speech disorders-E-Book: Substrates, differential diagnosis, and management. Elsevier Health Sciences. 2013

[2] B. Leung and T. Chau. A multiple camera tongue switch for a child with severe spastic quadriplegic cerebral palsy. Disability and Rehabilitation: Assistive Technology, vol. 5, no. 1, pp. 58-68. 2010.

[3] B. García-Conde and C. J. James. “On the development of a low-cost EMG switch for communication using minimal muscle contractions.” In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 1668-1671). IEEE, Aug. 2016.

[4] S. Harada, J. O. Wobbrock, J. Malkin, J. A. Bilmes, and J. A. Landay. “Longitudinal study of people learning to use continuous voice-based cursor control.” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 347-356), Apr. 2009.

[5] N. Alves, T. H. Falk, and T. Chau. “A novel integrated mechanomyogram-vocalization access solution.” Medical engineering & physics, vol. 32, no. 8, pp. 940-944, 2010

[6] M. Lui, T. H. Falk, and T. Chau. “Development and evaluation of a dual-output vocal cord vibration switch for persons with multiple disabilities.” Disability and Rehabilitation: Assistive Technology, vol. 7, no. 1, pp. 82-88, 2012.

[7] G. Azam and M. T. Islam. “Design and fabrication of a voice controlled wheelchair for physically disabled people.” In International Conference on Physics Sustainable Development & Technology (ICPSDT-2015) (Vol. 1, pp. 81-90), Aug. 2015.

[8] A. Pradhan, K. Mehta, and L. Findlater “Accessibility Came by Accident, Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities.” In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (pp. 1-13), Apr. 2018.

[9] S. Adya, V. Garg, S. Sigtia, P. Simha, and C. Dhir. “Hybrid Transformer/CTC Networks for Hardware Efficient Voice Triggering.” arXiv preprint arXiv:2008.02323, 2020.

[10] R. Alvarez and H. J. Park. “End-to-end streaming keyword spotting.” In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6336-6340), IEEE, May, 2019.

[11] T. N. Sainath and C. Parada. “Convolutional neural networks for small-footprint keyword spotting.” In Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[12] J. Lee, H. Kim, and Y. Jung. “Patterns of Misidentified Vowels in Individuals With Dysarthria Secondary to Amyotrophic Lateral Sclerosis.” Journal of Speech, Language, and Hearing Research, vol. 63, no. 8, pp. 2649-2666, 2020.

[13] J. Lee, E. Dickey, and Z. Simmons. “Vowel-specific intelligibility and acoustic patterns in individuals with dysarthria secondary to amyotrophic lateral sclerosis.” Journal of Speech, Language, and Hearing Research, vol. 62, no. 1, pp. 34-59, 2019.

[14] A. K. Silbergleit, A. F. Johnson, and B. H. Jacobson. “Acoustic analysis of voice in individuals with amyotrophic lateral sclerosis and perceptually normal vocal quality.” Journal of Voice, vol. 11, no. 2, pp. 222-231, 1997.

[15] J. Shor, D. Emanuel, O. Lang, O. Tuval, M. Brenner, J. Cattiau, ... and Matias, Y. “Personalizing ASR for dysarthric and accented speech with limited data.” arXiv preprint arXiv:1907.13511, 2019.

[16] J. Lin, K. Kilgour, D. Roblek, and M. Sharifi. “Training keyword spotters with limited and synthesized speech data.” In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7474-7478). IEEE, May 2020.

[17] P. Warden “Speech commands: A dataset for limited-vocabulary speech recognition.” arXiv preprint arXiv:1804.03209, 2018.

[18] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, N., … and R. A. Saurous. “Tacotron: Towards end-to-end speech synthesis.” arXiv preprint arXiv:1703.10135, 2017.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. “Librispeech: an asr corpus based on public domain audio books.” In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5206-5210). IEEE, Apr. 2015.

[20] S. Watanabe, M. Mandel, J. Barker, and E. Vincent. “Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings.” arXiv preprint arXiv:2004.09249, 2020.

[21] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. “librosa: Audio and music signal analysis in python.” In Proceedings of the 14th python in science conference, vol. 8, pp. 18-25, July 2015.

[22] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” In International conference on machine learning, pp. 448-456, PMLR, June 2015.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting.” The journal of machine learning research, vol. 15, no. 1, pp. 1929-1958, 2014.

[24] D. A. Clevert, T. Unterthiner, and S. Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus).” arXiv preprint arXiv:1511.07289, 2015.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, ... & X. Zheng. “Tensorflow: A system for large-scale machine learning.” In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp. 265-283, 2016.

[26] F. Chollet. Deep learning with Python (Vol. 361). New York: Manning, 2018.

[27] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, ... and D. H. Yoon, D. H. “In-datacenter performance analysis of a tensor processing unit.” In Proceedings of the 44th annual international symposium on computer architecture (pp. 1-12), Jun. 2017.

[28] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization.” arXiv preprint arXiv:1412.6980, 2014.

[29] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley. “Google vizier: A service for black-box optimization.” In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1487-1495), Aug. 2017.

[30] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, ... & K. Wilson. CNN architectures for large-scale audio classification. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 131-135). March 2017.