Speaker Verification based on Deep Neural Network for Text-Constrained Short Commands

Heesu Kim*, Euntae Choi* and Kiyoung Choi*
* Department of Electrical and Computer Engineering
Seoul National University, Seoul, South Korea
E-mail: hkim@dal.snu.ac.kr, ideal_kar@dal.snu.ac.kr, kchoi@snu.ac.kr

***Abstract*—Speaker verification has been known to be a tough task especially under the condition of short utterances. Based on the observation that actual voice commands are composed of a few repeated words, we propose an effective approach for building and training a deep neural network to extract features with properties appropriate for tackling such condition. We demonstrate the effectiveness through experiments independently designed for each property. Our proposed approach achieves 5.89% equal error rate on word scale commands shorter than 1 second, and with a linear discriminative analysis, it decreases to 3.43%.**

Index Terms: speaker verification, short utterance, text-constrained, voice commands

I. Introduction

As electronic devices are extensively using voice for their control, Speaker Verification (SV) has become an interesting topic to numerous researchers. Unfortunately, however, it does not seem as robust as classical verification (e.g., text-password verification) because the voice signature of a person is very likely to have variability coming from both recording environment and utterances' lexical contents. Formally, SV is a task to verify whether the speaker has been enrolled or not by analyzing his or her voice input. SV systems are usually categorized into two classes in terms of flexibility of phrase choice and lexical variability. One is Text-Dependent Speaker Verification (TD-SV). Given a set of predefined pass-phrases (e.g., Ok Google or Hey Siri), it tries to verify not only speaker identity but also whether the speaker is uttering the correct pass-phrase. TD-SV system benefits from constrained lexical variability while having little flexibility in choosing pass-phrases. The other is Text-Independent Speaker Verification (TI-SV), where only the speaker identity is considered. However, it generally needs a more complex model and dataset to train the TI-SV system as it must deal with much wider lexical variability of the speakers' voice inputs. Also, TI-SV systems require the inputs to be relatively longer to obtain consistent performance [1].

For commercial products, they currently have a restricted command set which is supported by the products such as registering the schedules and alarms, sending a text message, and making a phone-call. Although actual commands are not identical to each other, there are common words occurring repeatedly in commands since they share the structure of sentence; examples of such common words include interrogatives, digits, and date. Accordingly, Text-Constrained SV (TC-SV) can be another condition where SV allows lexical variability to some extent but is constrained to a limited vocabulary. With this condition, TC-SV could obtain consistent and better performance with relatively simple model and dataset compared to TI-SV. At the same time, it acquires better flexibility in lexical variability than TD-SV. Some previous works [2][3] tried to improve SV performance in this condition and we focus on TC-SV in this paper.

Typical voice commands are not designed to have long duration (e.g., a few reserved keywords, commands (verbs), digit passwords). Poddar et el. [4] interpreted this condition as a trade-off between user convenience and SV performance. Therefore, short utterances (i.e., short commands) are worthy to be considered for SV. However, that incurs severe performance degradation in TI-SV [1] due to their insufficient phonetic information. Li et al. [5] handled it by clustering speech units and synthesizing the model for unseen speech unit classes to cover insufficient phonetic information in short utterances. Unlike it, we mitigate this problem by changing the condition to the text-constrained condition, which alleviates the insufficient phonetic information problem in the short utterances and maximally exploit the trade-off.

Probabilistic model-based approaches have been widely used and proven their effectiveness on SV. Gaussian Mixture Model - Universal Background Model (GMM-UBM) [6] represents the speaker and channel independent attributes over their Gaussian components. Therefore, the speaker-dependent model can be acquired via MAP adaptation from GMM-UBM. The Joint Factor Analysis (JFA) model [7] factorizes GMM-UBM supervector into the channel, speaker, and residual factors to deal with non-speaker related factors independently. The i-vector model [8] is a simplified one but outperforms the existing models. It trains the total variability matrix where the channel and speaker variabilities are compressed into low dimensions. The Linear Discriminative Analysis (LDA) and Probabilistic LDA (PLDA) are used to improve i-vector to discriminate speakers by reducing the within-class variation such as channel effect. The i-vector model has been known as the state-of-the-art model for SV. However, it suffers from sensitivity to lexical variability for short utterances [4]. In addition, Li et al. [9] demonstrated in detail that i-vector experiences dramatic performance drops for short utterances. Zhong et al. [2] also studied that problem and achieved a better result with reduced senone sets constrained

to digits. They utilized deep neural networks (DNNs) just to take an assistant role for selecting the senones and augmenting the input feature with a bottleneck feature. That looks similar to our work in a sense that we also utilize intrinsic properties of the features from a DNN model, but we use it primary features.

In the past few years, DNNs have shown great accomplishments on SV. DNNs have replaced some components of existing approaches based on probabilistic models. For example, Lei et al. [10] replaced the GMM which produces frame alignments with a DNN to calculate the sufficient statistics for training i-vector extractor. Due to its more precise and robust frame alignments, DNNs enable i-vector model to show a better performance than before. Instead of being a part of the existing processes, some approaches [9][11][12] use DNNs as the speaker-discriminative feature extractor (the feature vector is called d-vector in contrast to the i-vector). Heigold et al. [13] extend its role to classifier. They trained a DNN model end-to-end through the back-propagation algorithm. Li et al. [9] showed that d-vector can produce high-quality speaker features even with a very short utterance about 0.3 seconds and concluded that speaker traits are a kind of short time patterns that can be extracted in a short utterance. However, they used the model with a lot of parameters and a large dataset including 95,167 utterances of 5,000 speakers to achieve reasonable performance for text-independent scenario. Li et el. [3] alleviated the lexical variability from text-independent to text-constrained condition and they proposed phone-dependent training to handle it; they simply concatenated phone posteriors with acoustic feature. Unfortunately, however, it showed only a marginal improvement. In this paper, we investigate and exploit the potential of d-vector for SV with very short utterances under text-constrained condition, we then propose effective ways of extracting the d-vector under this condition.

II. Speaker Verification based on the D-vector

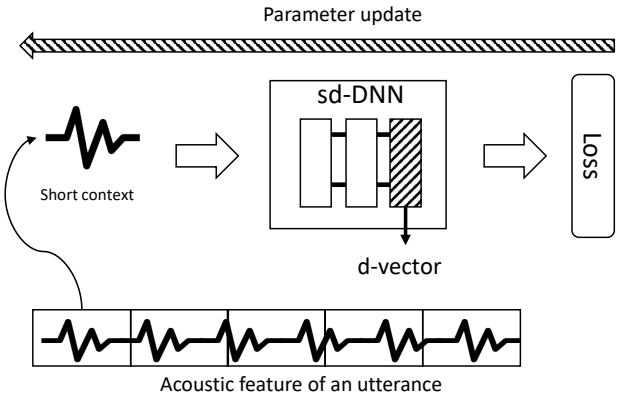

Fig. 1. Training process of the sd-DNN model. Note that the activations of the last layer make a d-vector.

To obtain d-vectors, *Speaker-Discriminant DNN (sd-DNN)* model should be trained first. DNN models can be trained to map inputs to a space where the target classes are discriminant, which is called representation learning [14]. As shown in Fig. 1, the sd-DNN model is trained to reduce cross-entropy loss which quantifies errors in classifying the speaker identities based on the presented utterances. After the training reaches a convergence, the activations of the last hidden layer are treated as the speaker-discriminative feature which is called the d-vector.

Before running the SV on test utterances, each speaker who wants to be enrolled should present several utterances during the enrollment stage like sign up process. Then all d-vectors of the presented utterances are averaged to make a speaker model (i.e., speaker id) belonging to the corresponding speaker. After enrollment stage is done, a test utterance is also transformed to a d-vector, and then the cosine distance of the test d-vector is measured with respect to the speaker model. Only when the measured distance exceeds a predefined threshold, an utterance is authenticated as the one from the enrolled speaker.

III. D-vector Properties

A. Local Pattern

Convolutional Neural Networks (CNNs) do not need hand-crafted features, instead, take raw input data and extract optimal features for classification. Therefore, they are trained to extract the most discriminative features specialized for the classification objective. For acoustic inputs, Mel-Frequency Cepstral Coefficients (MFCC) or Mel-scaled energy Filter banks (Fbank) are typically used and shaped into a single-channel 2D image-like data. Therefore, the superiority of CNNs [15] is still valid for such acoustic inputs [16]. In CNN models for SV, the filters extract different types of local patterns while sliding the filter windows over the inputs, and then they are transformed into more abstract features enough to represent the speaker identity. Note that the state-of-the-art CNN model is constructed with small sized filters [17] that effectively capture informative local patterns.

Unlike The probabilistic-model-based approaches [6][8] where the model extracts statistics frame by frame and then computes the utterance-level feature based on the statistical model such as Gaussian mixture model, the sd-DNN model does not have an explicit statistical model. Instead, it merely averages frame level features to generate an utterance-level feature. That property makes d-vector superior to i-vector for short utterances where the statistics can be easily biased [9].

B. Lexical Invariant Feature

As mentioned in Section II, the sd-DNN model is trained to correctly classify speakers without considering lexical variability of utterance; lexical variability is considered as noise during training. Wang et al. [18] evaluated various features for SV including d-vector. In their experiment, the use of d-vector failed for tasks that require lexical information such as detecting spoken terms and classifying word orders, but it showed the best performance for tasks that require

speaker-discriminative information. That means d-vector is somewhat invariant to lexical variability. Therefore, it is expected that d-vector is tolerable to lexical mismatch caused by the very short length of utterance. Of course, it is not always true; if an utterance is too short, it is not possible to discriminate the speakers due to its invalid local patterns.

IV. Training Strategies for The Speaker-Discriminative DNN

The loss used to train the sd-DNN model is not designed specifically for speaker verification since it does not fully represent the SV process explained in Section II. Therefore, even they have a positive correlation, the lower loss is not directly connected to the better SV performance. In short, we figure out that SV performance is closely related to the generalization performance which is opposite sign to overfitting. Because speakers seen during sd-DNN model training and speakers in SV process are different (i.e., Zero-shot learning). In that perspective, we propose training strategy for sd-DNN model.

There are many options for granularity of d-vector from a frame to entire frames of an utterance. We figure out that when the input granularity is excessively big, it can induce overfitting. Because with stacked layers of the sd-DNN model, their receptive fields are wider as the layer is deeper and eventually cover all portion of input at the penultimate layer where d-vectors are extracted. In that case, the sd-DNN model tends to overfit utterance-level feature especially seen in the training. As a result, utterances from the same speaker would not be verified to the same identity due to their lexical variability, thus resulting generalization performance degradation. Therefore, we propose to use a few frames (called short context) as a granularity of d-vector so, the sd-DNN model is trained to extract the feature from short context, which means the speaker traits are less dependent to lexical variability and length of utterance. Note that our sd-DNN model has the advantage of extracting local pattern due to their convolutional layers.

Another control knob that we must set is how to make a batch for training. According to [19], generating batch is a crucial part of feature vector quality (d-vector in here). There are two options; one is to change the selected utterances after a short context of each utterance is randomly batched, the other is to change the utterances after all short contexts are processed. We call the former "random-splice batch" and the latter "full-splice batch". The main difference is that the full-splice batch makes the sd-DNN update their parameters for the utterances (i.e., the speakers) continuously but the random-splice batch do it discretely. Of course, it is expected that the result of random-splice batch training will converge to that of full-splice batch after enough epochs. However, we figure out that random-splice batch incurs overfitting during training, which is not recovered shortly or permanently. Opposite to [20] where the context size is 3 seconds and random-splice batch helps avoid overfitting, it rather makes model parameters updated with insufficient or biased acoustic feature in case of short context size less than 0.5 seconds. Thus, we adopt the "full-splice batch" for the better generalization performance.

V. Text-Constrained

Since the text independence condition requires the model to cover wide lexical variability, the model must be large and complex (i.e., deep and wide neural network) and dataset for training the model should be large to cover wide lexical variability as in [20]. However, in the text-constrained condition, the model can focus on dealing with limited vocabulary, so it can perform well while keeping their structure less complex and more compact thereby reducing the computing resource requirements. Note that our text-constrained condition does not mean all utterances should contain only the words in the vocabulary. Instead we assume that the majority of words in the utterances is included in the vocabulary. And it also does not mean that our sd-DNN model cannot handle out-of-vocabulary problem at all. As we stated in Subsection III-B, our model has the lexical invariant property, and thus it can handle that problem to some extent as shown in Section VI.

VI. Experiments

TABLE I
Dataset Statistics used for Experiments.

	speakers	**words**	**utterance**
Speech commands	1,881	30	65,000
Development[1]	1,766	20[3]	37,148
Evaluation[2]	102	30	11,220

[1] For sd-DNN training [2] For SV evaluation [3] Seen words

A. Dataset

To validate our proposed idea that DNN-based SV is feasible for very short utterances composed of constrained vocabulary, we perform some experiments. The dataset we used in our experiments is *Speech Command* [21], which was released in August 2017 by Google Inc. The dataset contains 65,000 one-second long utterances uttering one of 30 short words by over a thousand different people. Although it was originally intended to evaluate keyword recognition, it has short duration and limited vocabulary that fit well with our target conditions (i.e., text-constrained and short utterance). Table I summarizes the dataset statistics. We split this dataset into two parts so that the first one is used to train our sd-DNN model (Development Set) and the other is used to evaluate the SV performance (Evaluation Set). They do not share any speaker. As the dataset provider did, we consider 20 words including 'yes', 'no', digits, and words for directions as seen words and the other 10 words as unseen words. We select 102 speakers for SV and each of them has 110 utterances consisting of 100 out of seen words and 10 out of unseen words. Next, 30 people are randomly sampled to be enrolled speakers and for each speaker, 40 utterances (2 per 20 seen

words) are picked as enrollment utterances. The other 72 speakers are considered as imposters.

Our input feature is the 40-dimensional Fbank, which is the intermediate result before DCT is applied in the way to generate MFCC. It is known to better keep local patterns intact than the MFCC because of the absence of DCT [22]. We use window length of 30ms and shift 10ms along the time. And we eliminate non-speech frames of utterances in the development set by Voice Activity Detection (VAD).

TABLE II
EQUAL ERROR RATE (EER, %) ACROSS DIFFERENT BATCH METHODS

Batch Sampling	cosine distance	w/ LDA
Full-Splice	5.89	3.43
Random-Splice	6.19	3.84

B. Model and Training

As we stated in Section V, the text-constrained condition alleviates the requirement of a complex model. Therefore, we decide to exploit a simple model, composed of only 4 blocks each of which contains a convolutional layer followed by batch normalization, ReLU activation, and max-pooling layers. The output neurons correspond to the speaker labels whose size is 1,766, which is the number of speakers in the development set used to train the model. For sd-DNN model training, we use the stochastic gradient descent algorithm with 0.9 momentum and 10^{-6} weight decay, and our learning rate and batch size are set to 0.01 and 64 respectively. We train it until there is no more improvement in validation accuracy. And we apply the two different batch methods (full-splice and random-slice). As shown in Table II, full-splice batch shows better equal error rate (EER)[1] values as we stated in Section IV. Our simple model archives 5.89% EER by cosine distance and it decreases to 3.43% when we apply the LDA on d-vectors.

C. Context size and Utterance Duration Scalability

TABLE III
EQUAL ERROR RATE (EER, %) AND SPEAKER IDENTIFICATION ACCURACY (SI ACC., %) ACROSS THE DIFFERENT CONTEXT SIZES AND NUMBERS OF WORDS IN AN UTTERANCE

		Number of Words			
Context size	SI Acc.	1	2	3	4
$100ms$	19.35%	**5.89**	**2.66**	**1.33**	**0.84**
$200ms$	30.65%	5.95	2.68	1.43	0.90
$500ms$	67.48%	10.62	5.31	3.19	2.21

We train the model while sweeping the context size for a d-vector. The result in Table III verifies our claim that training the sd-DNN model with short context helps avoid overfitting. Consequently, we use $100ms$ as the context size for all experiments if there is no explanation.

Until now, we have evaluated SV performance on word scale utterances. However, the duration of utterances can vary according to the number of words (i.e., complexity) in voice commands. To take this into account, we synthesize longer utterances by concatenating several words. Of course, it is not identical to the utterance recorded at a time, but it is still useful to estimate how increasing utterance duration affects the SV performance. We sweep the number of words from one to four. The results presented in Table III show that the extension of utterance duration improves the SV performance a lot. This experiment demonstrates that shortness of utterance is one of harsh obstacles for the SV and our model has the great scalability to utterance duration.

[1]Error rate at the point where false positive and false negative rate are equal

TABLE IV
EQUAL ERROR RATE (EER, %) WITH DIFFERENT FEATURES AND NUMBERS OF WORDS IN AN UTTERANCE

	Number of Words			
Feature	1	2	3	4
i-vector (cosine distance)	6.51	2.78	1.46	**0.66**
d-vector (cosine distance)	5.89	2.66	1.33	0.84
i-vector (w/ LDA)	7.70	3.99	2.36	1.14
d-vector (w/ LDA)	**3.43**	**1.88**	**1.19**	0.79

D. i-vector vs d-vector

In the i-vector model used for the experiment, the GMM-UBM has 1024 Gaussians and i-vector dimension is 256. We use the same development set for the training of the i-vector model and the same evaluation set for SV performance comparison. However, we use MFCC for the input feature, unlike the d-vector. The result is shown in Table IV. The d-vector shows better performance compared to i-vector in cases with fewer words (i.e., shorter utterances). However, in the four words case, i-vector shows better performance. As a result, we can conclude that d-vector is more suitable than i-vector for short utterances as we stated in Subsection III-A. When we apply LDA to both, the performance gap becomes larger. For i-vector, LDA does not seem to be helpful under short utterance condition.

TABLE V
EQUAL ERROR RATE (EER, %) ACROSS TEST UTTERANCES COMPOSED OF SEEN, UNSEEN, AND MIXED WORD SET INDEPENDENTLY

word set	Seen	Unseen	Mixed
cosine distance	5.75	6.75	5.89

TABLE VI
EQUAL ERROR RATE (EER, %) VARIATION ACCORDING TO SPEAKER MODEL CONFIGURATIONS

	(kind of words, # of each word)				
	(2, 2)	(4, 1)	(10, 2)	(20, 1)	(20, 2)
cosine distance	8.38	7.41	6.64	6.38	5.89
w/ LDA	6.37	4.93	4.29	3.60	3.43

E. Tolerance to lexical variability and Mismatch

As we mentioned in Table I, We reserve 10 words as unseen words, which are not included in the development set, thus our sd-DNN model has not seen those words during training. Therefore, SV on unseen words examines the tolerance of our model to the lexical variability. In Table V, The 'Seen' and 'Unseen' mean utterances are composed of only seen and unseen words, respectively and the 'Mixed' means no restriction on words. The results in Table V show a little worse EER for the unseen word set and nearly unchanged EER for the mixed word set with compared to seen words set. Note that the mixed word set is the most common case for real-life usages.

In addition, we also consider a case where speakers do not provide enough lexical variability during enrollment, which incurs lexical mismatch during testing. To evaluate SV performance for that case, we restrict the enrollment in two ways; one is the kind of words and the other is the number of each word. The kind of words determines the level of lexical mismatch between the speaker model and test utterances. As shown in able VI, as the total number of utterances increases, the quality of the speaker model improve accordingly. Although the larger total number of utterances consistently make the EER better, the kind of words is more sensitive factor given the same number of words. In particular, it becomes severe in the case of small amount of enrollment utterances. However, our model still works with (4,1) configuration (4 utterances). There is only 1.5%p drop compared to (20,2) configuration (40 utterances).

Those two experiments shows our model tolerates the lexical variability and the features from the model has tolerance to lexical mismatch between the speaker model and test utterances.

VII. Conclusions

We have shown that a DNN can generate powerful speaker-discriminative features from short utterances under the text-constrained condition. We have also proposed an effective way to train the model and extract the features applicable to our target conditions. The generalization performance of the model is the most important factor in the sd-DNN model training for speaker verification. To achieve this, we propose to use short context granularity and full-splice batch training. According to our extensive experiments, our model successfully learns to extract latent local patterns from short contexts in an utterance and shows better EER than the state-of-the-art technique using i-vector. Our proposed approach also shows tolerance to lexical variability and mismatch caused by unseen words and insufficient enrollment utterances.

Acknowledgment

This research was supported through an academic-industrial collaboration funded by Samsung Research, Seoul, Korea

References

[1] R. Vogt, S. Sridharan, and M. Mason, "Making confident speaker verification decisions with minimal speech," *IEEE transactions on audio, speech, and language processing*, vol. 18, no. 6, pp. 1182–1192, 2010.

[2] J. Zhong, W. Hu, F. Soong, and H. Meng, "Dnn i-vector speaker verification with short, text-constrained test utterances," *Proc. Interspeech 2017*, pp. 1507–1511, 2017.

[3] L. Li, D. Wang, Z. Zhang, and T. F. Zheng, "Deep speaker vectors for semi text-independent speaker verification," *arXiv preprint arXiv:1505.06427*, 2015.

[4] W. Li, T. Fu, H. You, J. Zhu, and N. Chen, "Feature sparsity analysis for i-vector based speaker verification," *Speech Communication*, vol. 80, pp. 60–70, 2016.

[5] L. Li, D. Wang, C. Zhang, and T. F. Zheng, "Improving short utterance speaker recognition by modeling speech unit classes," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 6, pp. 1129–1139, 2016.

[6] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[7] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[8] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[9] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 1542–1546, 2017.

[10] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[11] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[12] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[13] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

[14] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[18] S. Wang, Y. Qian, and K. Yu, "What does the speaker embedding encode?" in *Proc. Interspeech 2017*, vol. 2017, 2017, pp. 1497–1501.

[19] K. Sheng, W. Dong, W. Li, J. Razik, F. Huang, and B. Hu, "Centroid-aware local discriminative metric learning in speaker verification," *Pattern Recognition*, vol. 72, pp. 176–185, 2017.

[20] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *Proc. Interspeech 2017*, vol. 3, pp. 33–039, 2017.

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

[22] A. Torfi, N. M. Nasrabadi, and J. Dawson, "Text-independent speaker verification using 3d convolutional neural networks," *arXiv preprint arXiv:1705.09422*, 2017.