Word-level Embeddings for Cross-Task Transfer Learning in Speech Processing

Pierre Beckmann[1]
Swiss Federal Institute of Technology Lausanne
Lausanne, Switzerland
pierre.beckmann@epfl.ch

Mikolaj Kegler[1]
Imperial College London
London, United Kingdom
mikolaj.kegler16@imperial.ac.uk

Milos Cernak
Logitech Europe S.A.
Lausanne, Switzerland
milos.cernak@ieee.org

***Abstract*—Recent breakthroughs in deep learning often rely on representation learning and knowledge transfer. In recent years, unsupervised and self-supervised techniques for learning speech representation were developed to foster automatic speech recognition. Up to date, most of these approaches are task-specific and designed for within-task transfer learning between different datasets or setups of a particular task. In turn, learning task-independent representation of speech and cross-task applications of transfer learning remain less common. Here, we introduce an encoder capturing word-level representations of speech for cross-task transfer learning. We demonstrate the application of the pre-trained encoder in four distinct speech and audio processing tasks: (i) speech enhancement, (ii) language identification, (iii) speech, noise, and music classification, and (iv) speaker identification. In each task, we compare the performance of our cross-task transfer learning approach to task-specific baselines. Our results show that the speech representation captured by the encoder through the pre-training is transferable across distinct speech processing tasks and datasets. Notably, even simple applications of our pre-trained encoder outperformed task-specific methods, or were comparable, depending on the task.**

***Index Terms*—Speech processing, deep learning, transfer learning, feature extraction**

I. Introduction

Deep learning frameworks for computer vision and natural language processing often rely on representation learning and knowledge transfer [1]. The goal of transfer learning is to build up domain-specific knowledge on one task and transfer it to another downstream task [2]. Currently, three main transfer learning approaches can be distinguished: (i) feature extraction, whereby the pre-trained model provides compact representations of domain-specific data [3], (ii) fine-tuning, whereby the knowledge captured by a pre-trained model can be adjusted (i.e. *fine-tuned*) to a particular task or dataset [4], and (iii) computing feature losses, whereby representations, obtained through the pre-trained feature extractor, are used to compute losses for training deep learning systems [5].

Following numerous successful applications of deep learning in speech processing, learning representations of speech became the next focus in the field [6]. In particular, learning unsupervised audio representations and evaluating them on downstream classification tasks has recently shown promising results [7]–[9]. Similarly, one of the recent trends in Automatic Speech Recognition (ASR) is an application of unsupervised [10], [11] or self-supervised [12], [13] speech representations, as a model pre-training followed by fine-tuning, or as auxiliary speech embedding features.

Recent studies investigating neural encoding of spoken language suggest multi-scale parsing of incoming information into units of the appropriate temporal granularity [14], roughly at a segmental (such as phonetic) and supra-segmental (such as syllabic) timescales. We argue that existing unsupervised and self-supervised speech representations emphasize segmental features (i.e., acoustic models), and when used with a subsequent language model can facilitate ASR. Here, we explore the complementary case, where the acoustic model is represented on the supra-segmental level. Such word-level representation of speech, used without a language model, may be therefore more suitable for a wider range of distinct non-ASR tasks.

In this paper, we introduce a spoken word encoder for learning task-independent representation of speech. In contrast to most current transfer learning approaches, our method was designed to be flexible, versatile and applicable across a range of distinct speech and audio processing tasks. Our example encoder presented here adopts VGG-16 architecture [15], similar to VGG-like models applied previously in audio classification [16] and speaker identification [17]. Importantly, the proposed methodology does not rely on this particular model, and can be easily employed with other architectures.

We hypothesize that the spoken word encoder's successive layers capture hierarchically organized generalized representations of speech at the intersection of acoustics and linguistic information. Notably, Speech2vec [18] encodes similar spoken word representation, however, it was applied only *within-task* for word similarity experiments, but not other, different downstream tasks. We thus evaluated our encoder in a *cross-task* and *cross-dataset* configuration using four distinct speech and audio processing problems: (i) speech inpainting [19], (ii) language identification [20], (iii) speech, noise and music classification [21] and (iv) speaker identification [22]–[24].

The paper is organized as follows: Section II introduces the spoken word encoder, its pre-training and its capability for transfer learning. Section III presents four applications of the pre-trained encoder in audio and speech processing tasks, results and comparisons to relevant baselines. Section IV concludes the paper and outlines avenues for future research.

[1]-These authors contributed equally to this work. We would like to thank Logitech & Imperial College London for supporting this study.

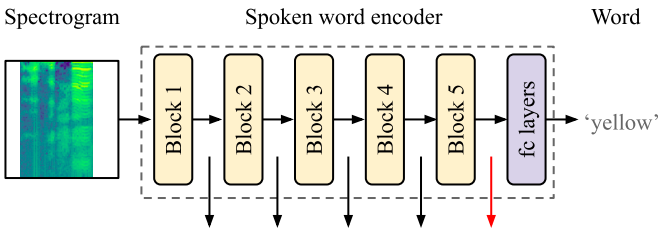

Fig. 1. SpeechVGG spoken word encoder and knowledge acquisition via the word classification task. Vertical arrows represent the activation at pooling layers, which reflect the hierarchical representation of speech features captured by the model. The output from the last block provides the most compact representation of speech features (red).

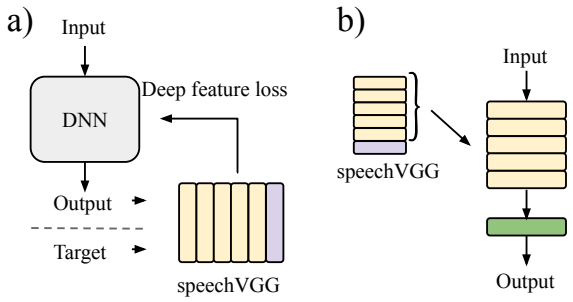

Fig. 2. Applications of the spoken word encoder in transfer learning. The pre-trained model can be used to: (a) compute deep feature losses or (b) to transfer knowledge to a new task as a feature extractor (i.e. fixed weights) or as a fully fine-tuneable module.

II. Spoken word encoder

Here we proposed a word encoder based on VGG-16 architecture, thus from now on denoted as *speechVGG*. Importantly, the proposed methods can be paired with other architectures and applied to transfer knowledge between various speech processing tasks beyond those considered here.

A. Encoder architecture

Diagram illustrating the architecture of our encoder and the pre-training task is presented in Fig. 1. The model adopts the VGG-16 architecture [15]. Specifically, the network is built out of five main blocks (Fig. 1, yellow), each composed of stacked convolution layers followed by ReLU activation and concluded by a max-pooling layer. The output from the last block is subsequently processed through two fully-connected linear layers followed by a softmax output layer (Fig. 1, purple). Note that depending on the task to which the model is deployed, the final fully-connected and output layers of the model may be modified (Fig. 2b).

B. Dataset & encoder pre-training

We used LibriSpeech dataset [25] to train the speechVGG. We used all the available training data to build sets of 100 (*train-clean-100*), 460 (+ *train-clean-360*) and 960 hours (+ *train-other-500*) of speech material and used them to train the encoder. We used *test-clean* as a validation set during the model training and *dev-clean* as a separate subset of data to evaluate the performance of the fully trained model. We trained the encoder on the word classification task using different training dictionaries extracted from the LibriSpeech transcriptions. We considered three dictionaries containing 1000, 3000 & 6000 most frequent, at least 4-letters-long, words from the available data. Together, all considered dictionary and training data sizes made up nine possible training configurations.

For each training setup, we obtained word boundaries (the start and end frames) using forced-alignment from Kaldi LibriSpeech setup [26], and extracted the corresponding segments from the data. We computed log-magnitude spectrograms for each extracted segment by taking absolute values of a complex short-time Fourier transform (STFT, 256 samples window with 128 samples overlap, 128 frequency bins) and then applying natural logarithm. Each frequency channel of the log-magnitude STFT was normalized using mean and standard deviation obtained from the corresponding training dataset.

Each training sample was augmented using SpecAugment [27] to improve the model's generalization capacity. The augmentation was applied by replacing random blocks of time and frequency bins (no more than 50% in each dimension) in the spectrograms with mean values. To address the varying duration of words, each time-frequency representation of a word was randomly padded with zeros to a size of 128 x 128, corresponding to a 1024-ms-long segment. Such a combination of zero-padding and augmentation facilitated the extraction of speech features in the model. We hypothesize that the zero-padding allows the model to learn to identify parts of the input containing speech, while augmentation makes the learned representations generalized.

Each configuration of the encoder was trained via cross-entropy loss for 30 epochs using ADAM optimizer with a learning rate set to 5×10^{-5}. For all of the considered training configurations the model yielded over 92% accuracy in the word classification task (on held-out data), therefore indicating successful training and knowledge acquisition.

C. Applications in transfer learning

Our spoken word encoder was designed to extract features from up to 1024-ms-long samples of audio. Features of longer samples can be obtained by averaging the representations from several windows. Due to the hierarchical, modular architecture, each block of the model emphasizes distinct features of the input. The highest-level features, obtained from the last max-pooling in the model, provide the most compact and informative representation of the input (Fig. 1, red).

The pre-trained encoder can be applied to extract features in a range of speech processing tasks. They can be employed directly as feature extractors in different downstream tasks or used to train deep learning systems via (deep) feature losses [19] (Fig. 2a). In the latter case, the extractor pooling layers' activation provides rich representations of both training output and the target. The direct loss computed between these

two representations (for example L_1) can then be used to train the main system.

The pre-trained encoder can be also employed to provide a hot start for learning a brand new task. This can be accomplished by replacing the final layers of the extractor or attaching the other system to the output of the encoder's final block (Fig. 2b). Such a system benefits from the knowledge already captured by the pre-trained model and can be furthermore fine-tuned. In the calibration process, the extractor's weights can be either fixed (i.e. *frozen*) or fine-tuned with the rest of the system. In both cases, a set of generalized speech-specific weights facilitates the (re-)training process and the trained system's overall performance.

III. Transfer learning experiments

A. Speech inpainting & benchmarking training setups

In our previous work [19], we employed the speechVGG pre-trained using 1000 words from 100 hours of speech recordings to train a deep speech inpainting system for reconstructing missing or distorted parts of the time-frequency representation of speech. Here, we explored how different speechVGG pre-training configurations (section II-B) influence the speech inpainting performance to determine the optimal setup.

We adopted the exact same framework for deep speech inpainting as introduced in [19]. In particular, we used *train-clean-100* from the LibriSpeech dataset to train the inpainting framework and *dev-clean* as an independent dataset for model evaluation. All the speech material was chunked into 1024-ms-long segments and preprocessed in the same way as for the speechVGG pre-training (see section II-B for details). Each log-magnitude spectrogram was then distorted using random time & frequency masks, similar to those applied in SpecAugment [27]. The masks removed from 10% up to 40% time and frequency bins from the input STFTs. Such samples, along with their mask (i.e. the position of the intrusion was known) were processed through the network to reconstruct the original time-frequency representations of speech. Waveforms were obtained directly from the reconstructed STFT magnitudes using the locally weighted sums algorithm [28].

The speech inpainting system was trained using nine different configurations of speechVGG specified in section II-B. Pre-trained speechVGG was each time used as a feature extractor with fixed weights and applied to compute feature losses for training the inpainting system (Fig. 2a). Specifically, each reconstructed training sample and the corresponding target were processed through the pre-trained speechVGG. The deep feature loss was obtained by computing L_1 loss between activation of the speechVGG's pooling layers and used to train the speech inpainting model. The inpainting performance of the such trained model was quantified via the short term objective intelligibility (STOI) [29] and perceptual evaluation of speech quality (PESQ) [30] between the reconstructed and actual speech samples from the held-out dataset (*dev-clean*).

Results: Improvements of STOI & PESQ scores through speech inpainting, with respect to the unprocessed, distorted case, are reported in Fig. 3. SpeechVGG pre-trained to classify 3000 words extracted from 460 hours of speech recordings was the optimal setup leading to the largest improvements in STOI & PESQ scores. Notably, this configuration outperformed exsiting baseline employing speechVGG pre-trained to classify 1000 words from 100 hours of speech, as reported in [19]. The lack of improvement for larger sizes of either dictionary or training dataset may be attributed to the fact that over half of the 960 hours of LibriSpeech data belonged to the 'other' category, which contains inaccurate annotations of words [25]. We used the best performing configuration of speechVGG (460 hours + 3000 words) for the remaining experiments.

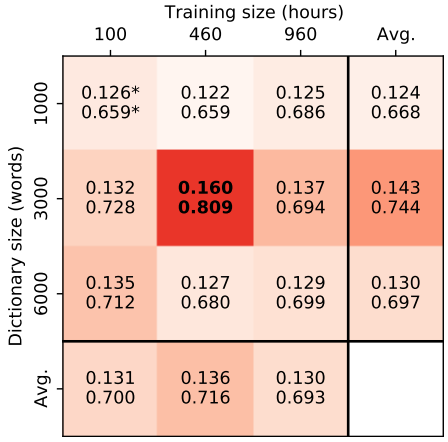

Fig. 3. Impact of the speechVGG on the training of the speech inpainting system via deep feature loss. Each cell in the array represents improvement of STOI (top) and PESQ (bottom) scores, with respect to the unprocessed case, averaged across the evaluation dataset. *-baseline performance from [19].

B. Language identification

The language identification experiment was performed using the Spoken Language Identification Kaggle dataset [20] that contains voice recordings in three languages: English, German, and Spanish. We followed the recommended train/test data split. Twenty randomly chosen 1024-ms-long segments were extracted from each recording and pre-processed to obtain their spectrograms, as specified in section II-B. Each segment was processed through the pre-trained speechVGG (460 hours + 3000 words), serving as feature extractor with fixed weights, to obtain its representation by flattening the output from the last block (Fig. 1, red). A set of features describing a particular recording was each time obtained by averaging representations of its 20 parts. Features obtained from training recordings were used to train a simple logistic regression classifier [31] to distinguish the three languages. The trained classifier was evaluated on a separate portion of the data, as stated above.

Results: Table I compares our transfer learning approach with the self-supervised audio representation learning [8] and a task-specific convolutional neural network [20]. With **97.6%** accuracy, our approach outperformed its counterparts using the pre-trained speechVGG as a fixed-weight feature extractor with no additional task-specific fine-tuning of the encoder itself. Although the speechVGG was pre-trained only on English, it was able to accurately distinguish all three

languages. This suggests that the representation of speech captured during the encoder pre-training is not language-specific and can generalize to other languages.

TABLE I
LANGUAGE IDENTIFICATION TASK.

Method	Accuracy (held-out data)
Tagliasacchi, et al. [8]	90.0%
Task-specific ConvNet [20]	97.0%
speechVGG	**97.6%**

C. Speech, music and noise classification

We used the MUSAN dataset [21] to classify three different categories of audio recordings: speech (recordings from the US government and librivox.org), music, and noise. We discarded all audio samples shorter than 1024 ms from the dataset. All such short samples were recordings of noise, and including them could lead to biased predictions based solely on the sample duration rather than its acoustic content. From the remaining data we set aside randomly selected 10% as a held-out evaluation set. Analogously to the previous task (section III-B), twenty randomly chosen 1024-ms-long segments were obtained from each recording in the dataset and pre-processed as specified in section II-B. The pre-trained speechVGG (460 hours + 3000 words), with fixed weights and no additional fine tuning, was used to obtain features from each segment. Same as before, a set of features for each recording was obtained by averaging representation of its segments. The features from the training portion of the data were used to train a simple logistic regression classifier [31] to distinguish speech, music, and noise. The classifier was evaluated using samples from the held-out portion of the data.

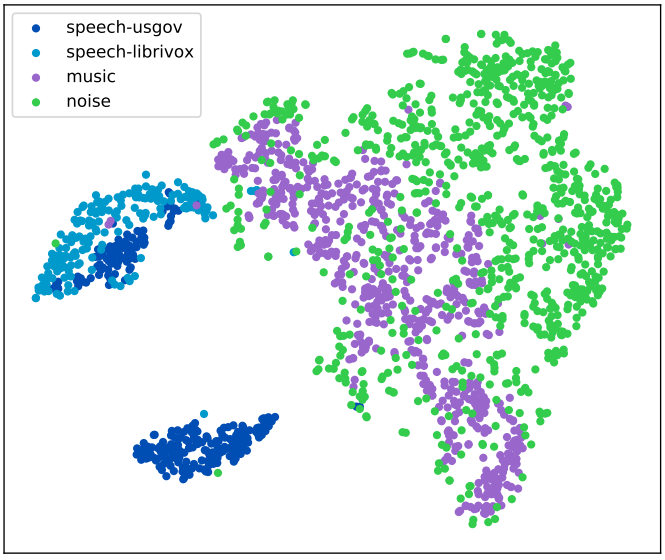

Fig. 4. t-SNE visualization of high-dimensional embeddings of speech, music and noise recordings from MUSAN, obtained via the pre-trained speechVGG.

Results: High-dimensional embeddings of the MUSAN dataset were visualized via t-SNE [32] (Fig. 4). Clusters representing speech recordings were clearly distinguishable from music and noise. Interestingly, embeddings of speech recordings were divided into two distinct clusters; one made up almost exclusively of US government recordings (Fig. 4, dark blue). Our approach yielded **96.5%** classification accuracy. This suggests that speechVGG, pre-trained on the LibriSpeech data, not only successfully transferred the generalized speech representation to this task, but also allowed to reliably distinguish samples of music and noise. Tagliasacchi, et al. [8] reported 99.0% accuracy on this task using representations from 0.975-seconds-long segments of recordings. Importantly, their approach was tailored specifically for this task, while our encoder was designed for versatile *cross-task* applications. Moreover, our approach used only twenty 1024-ms-long segments from each clip, i.e. up to 20.5 seconds of audio, instead of all available, as Tagliasacchi, et al. [8] did.

D. Speaker identification

In the speaker identification task, we used speech recordings from the TIMIT dataset, including 630 speakers [33]. We randomly selected one recording per speaker to form a set-aside evaluations set, while the rest of the data was used for training. All the data was chunked into 1024-ms-long segments, and the log-magnitude spectrogram of each chunk was obtained as specified in section II-B. The previously introduced approach, where the pre-trained speechVGG (460 hours + 3000 words) was used as a feature extractor with fixed weights, did not succeed in this task and led to poor performance. We thus replaced the output layer of the speechVGG to accommodate a different number of classes in the new task (630 speakers) and fine-tuned the model (Fig. 2b). For fine tuning, we used the same training routines as in the speechVGG pre-training, but fed the model with the task-specific data. In particular, the model was trained to classify speakers based on a single 1024-ms-long window. During evaluation on a set-aside portion of the data, the speaker identity was determined by averaging model predictions from a window sliding over the entire recording with a 50% overlap (512 ms).

Results: Results of speaker identification using the fine-tuned speechVGG, alongside the baseline approaches, are presented in Table II. The fine-tuned speechVGG achieved **99.7%** accuracy on the set-aside portion of the data and therefore outperformed existing methods evaluated on the entire TIMIT corpus [22], [23]. Ge et al. (2017) [24] reported 100% accuracy in this task employing a 1-second-long window using a subset of 100 male speakers. In the same setup our fine-tuned model also solved the task yielding **100%** accuracy (Table II - 100 male speakers).

TABLE II
SPEAKER IDENTIFICATION TASK.

Method	Accuracy (held-out data)	
	All speakers	**100 male speakers**
Ming, et al. [23]	608/630 (96.5%)	-
Wildermoth, et al. [22]	623/630 (99.0%)	-
Ge, et al. [24]	-	**100/100 (100%)**
speechVGG (fine-tuned)	**628/630 (99.7%)**	**100/100 (100%)**

IV. Discussion

Here, we proposed an approach for learning word-level embeddings, suitable for flexible knowledge transfer across different speech and audio processing tasks. In contrast to most existing *task-specific* transfer learning approaches our method is focused on versatility and *cross-task* compatibility. We evaluated the proposed spoken word encoder as a 'general-purpose' speech feature extractor and explored its performance in a range of distinct speech and audio processing tasks.

The generalized representation of speech captured during the encoder pre-trained on the LibriSpeech dataset [25] was transferable over four distinct tasks, employing different datasets, not limited to speech [20], [21], [33]. Interestingly, relatively simple applications of our pre-trained spoken word encoder were capable of achieving results comparable to the recent task-specific approaches with little to no additional fine-tuning (section III). Implementation of the speechVGG, pre-trained models and example applications are available at[1].

We would like to re-iterate that the proposed pre-training and transfer learning methodology is not restricted to the example encoder architecture introduced in Section II. Depending on the tasks of interest, the feature extractor can be of considerably higher or lower complexity than the example presented here. In particular, systematic exploration of different encoder configurations and fusion of our approach with existing (self-)supervised and unsupervised training setups may further improve efficacy of the proposed framework.

References

[1] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International Conference on Artificial Neural Networks*, pp. 270–279, Springer, 2018.

[2] 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.

[3] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, *et al.*, "Unsupervised and transfer learning challenge: a deep learning approach," in *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop*, pp. 97–111, 2011.

[4] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for asr using lf-mmi trained neural networks," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 279–286, IEEE, 2017.

[5] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in neural information processing systems*, pp. 658–666, 2016.

[6] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[7] F. Quitry, M. Tagliasacchi, and D. Roblek, "Learning audio representations via phase prediction," *arXiv preprint arXiv:1910.11910*, 2019.

[8] M. Tagliasacchi, B. Gfeller, F. Quitry, and D. Roblek, "Self-supervised audio representation learning for mobile devices," *arXiv preprint arXiv:1905.11796*, 2019.

[9] M. Tagliasacchi, B. Gfeller, F. Quitry, and D. Roblek, "Pre-training audio representations with self-supervision," *IEEE Signal Processing Letters*, vol. 27, pp. 600–604, 2020.

[10] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *Proc. Interspeech 2019*, pp. 3465–3469, 2019.

[11] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Learning robust and multilingual speech representations," *arXiv preprint arXiv:2001.11128*, 2020.

[12] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[14] A.-L. Giraud and D. Poeppel, "Cortical oscillations and speech processing: emerging computational principles and operations," *Nature neuroscience*, vol. 15, no. 4, p. 511, 2012.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[16] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 131–135, IEEE, 2017.

[17] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[18] Y.-A. Chung and J. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," *Proc. Interspeech 2018*, pp. 811–815, 2018.

[19] M. Kegler, P. Beckmann, and M. Cernak, "Deep Speech Inpainting of Time-Frequency Masks," in *Proc. Interspeech 2020*, pp. 3276–3280, 2020.

[20] T. Oponowicz, "Spoken language identification," 2018. Kaggle, https://www.kaggle.com/toponowicz/spoken-language-identification.

[21] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015. arXiv:1510.08484v1.

[22] B. R. Wildermoth and K. K. Paliwal, "Gmm based speaker recognition on readily available databases," in *Microelectronic Engineering Research Conference, Brisbane, Australia*, vol. 7, p. 55, 2003.

[23] J. Ming, T. J. Hazen, J. R. Glass, and D. A. Reynolds, "Robust speaker recognition in noisy conditions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1711–1723, 2007.

[24] Z. Ge, A. N. Iyer, S. Cheluvaraja, R. Sundaram, and A. Ganapathiraju, "Neural network based speaker classification and verification systems with enhanced features," in *2017 Intelligent Systems Conference (IntelliSys)*, pp. 1089–1094, IEEE, 2017.

[25] 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*, pp. 5206–5210, IEEE, 4 2015.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, IEEE Signal Processing Society, 2011.

[27] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[28] J. L. Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in *Proc. International Conference on Digital Audio Effects (DAFx)*, pp. 397–403, Sept. 2010.

[29] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4214–4217, IEEE, 2010.

[30] "Methods for subjective determination of transmission quality," *ITU-T Rec. P.800*, 1998.

[31] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[32] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[33] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 11 1992.

[1] https://github.com/bepierre/SpeechVGG