

# SpeechVGG: A Deep Feature Extractor for Speech Processing

Pierre Beckmann<sup>\*,1,2</sup>

Mikolaj Kegler<sup>\*,1,3</sup>

Hugues Saltini<sup>1,2</sup>

Milos Cernak<sup>1</sup>

<sup>1</sup>Logitech Europe S.A., 1015, Lausanne, Switzerland

<sup>2</sup>Swiss Federal Institute of Technology Lausanne, 1015, Lausanne, Switzerland

<sup>3</sup>Imperial College London, SW7 2AZ, London, UK

pierre.beckmann@epfl.ch, mak616@ic.ac.uk, hugues.slt@protonmail.com,  
milos.cernak@ieee.org

## Abstract

Recent breakthroughs in deep learning often rely on representation learning and knowledge transfer. In particular, readily available models pre-trained on large datasets are key for the efficient transfer of knowledge. They can be applied as feature extractors for data preprocessing, fine-tuned to perform a variety of tasks, or used for computing feature losses in the training of deep learning systems. While applications of transfer learning are common in the fields of computer vision and natural language processing, audio- and speech processing are surprisingly lacking readily available and transferable models. Here, we introduce speechVGG, a flexible, transferable feature extractor tailored for integration with deep learning frameworks for speech processing. Our transferable model adopts the classic VGG-16 architecture and is trained on a spoken word classification task. We demonstrate the application of the pre-trained model in four speech processing tasks, including speech enhancement, language identification, speech, noise and music classification, and speaker identification. Each time, we compare the performance of our approach to existing baselines. Our results confirm that the representation of natural speech captured using speechVGG is transferable and generalizable across various speech processing problems and datasets. Notably, relatively simple applications of our pre-trained model are capable of achieving competitive results.

**Index Terms:** Speech processing, deep learning, transfer learning, feature extractor, deep feature losses

## 1. Introduction

Deep learning frameworks for image and natural language processing often make use of representation learning and transfer of knowledge [1, 2, 3]. The goal of the process is to build up domain-specific knowledge on one task and transfer it to another downstream task [4, 5]. Currently, three main transfer learning approaches can be distinguished: (i) feature extraction [6], whereby the pre-trained model provides rich, compact representations of domain-specific data, (ii) fine-tuning [7], whereby the knowledge captured by a pre-trained model can be adjusted to a particular task or dataset, and (iii) deep feature losses [8], whereby high-dimensional representations, obtained through the pre-trained feature extractor, are used to compute losses for training deep learning systems.

Following numerous successful applications of deep learning in speech processing, learning representations of speech became the next focus in the field [9, 10]. Indeed, some speech production and perception theories [11] suggest that access to the invariant representation of speech could make

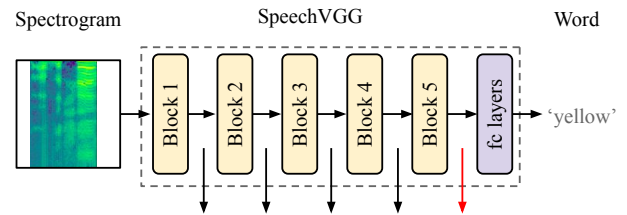


Figure 1: *speechVGG* architecture and knowledge acquisition via the word classification task. Vertical arrows represent activation of pooling layers, which reflect representation of speech features captured by the model. Output from the last block provides the most compact representation of speech features (red).

speech processing systems more robust to noise, as well as, more speaker- and language-independent. In particular, learning unsupervised audio representations and evaluating them on downstream classification tasks has recently shown promising results [12, 13]. There is also a growing number of studies that apply feature losses for training audio and speech processing system [14, 15, 16, 17]. However, applications of transfer learning in speech processing still remain scarce and surprisingly underexplored, possibly due to the lack of readily-available frameworks and pre-trained models. Representation learning and transfer of knowledge have remarkable potential for facilitating developments in the field of speech processing for a number of reasons. They could not only make the training faster, by using pre-trained models (i.e. hot start), but also increase efficacy and overall performance of the trained systems, especially when access to the problem-specific data is limited.

In this paper, we introduce speechVGG, a deep feature extractor tailored for speech processing. The proposed extractor adopted the architecture of the seminal deep convolutional neural network VGG-16 [18] and was trained on the spoken word classification task. We hypothesize that successive layers of the pre-trained model can capture hierarchically organized generalized representations of speech-specific features. We evaluated the pre-trained speechVGG on four potential use cases of transfer learning in speech processing to investigate its capability to transfer knowledge across different tasks and datasets. The selected tasks were: (i) speech inpainting [17], (ii) language identification [19], (iii) speech, noise and music classification [20] and (iv) speaker identification [21, 22, 23, 24].

The paper is organized as follows: section 2 introduces architecture and training of the speechVGG feature extractor, and its applications in transfer learning. Section 3 introduces four experiments employing speechVGG, presents their results and compares them to existing baselines. Finally, section 4 concludes the paper and outlines plans for future developments.

\*-These authors contributed equally to this work.

## 2. speechVGG

### 2.1. System architecture

Diagram illustrating the system architecture and the pre-training task is presented in Fig. 1. The model adopts the VGG-16 architecture [18]. 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 convolutional layer 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).

### 2.2. Model training

The LibriSpeech dataset [25], an open read speech corpus sampled at 16 kHz, was used 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 model. Note that the latter set contains 500 hours of ‘other’ speech with possible mistakes in word annotations. 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 trained model.

We trained speechVGG on the word classification task using different training dictionaries extracted from the transcription of the training recordings. We considered three dictionaries containing 1000, 3000 & 6000 most frequent, at least 4-letters-long, words from the data. Together, all considered dictionary and training data sizes made up nine training configurations.

For each configuration, 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 resulting log-magnitude STFT was normalized using mean and standard deviation obtained from the training data. To address the issue of varying duration of spoken words each time-frequency representation of a word was randomly padded with zeros to a size of 128 x 128, corresponding to 1024-ms-long segment.

Each sample used for training the speechVGG was augmented using SpecAugment [27] to improve the generalization capacity of the feature extractor. 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. Such 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 input containing speech, while augmentation makes the learned representations generalized.

Each configuration of speechVGG was trained using cross-entropy loss for 30 epochs via ADAM optimizer [28] with a learning rate set to  $5 \times 10^{-5}$ . All of the considered configurations of speechVGG yielded classification accuracy of over 92% suggesting successful training and knowledge acquisition.

### 2.3. Analysis of activation patterns

Insights from computer vision literature suggest that subsequent blocks of convolutional neural networks are sensitive to differ-

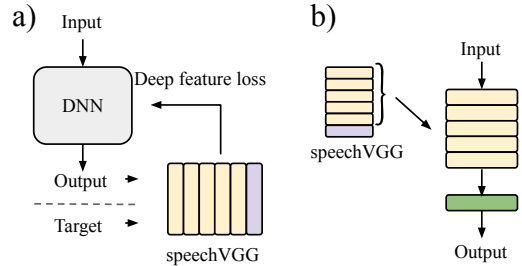


Figure 2: Applications of speechVGG in transfer learning. The pre-trained model can be used to: (a) compute deep feature losses or (b) to transfer knowledge to a new task as a feature extractor (i.e. fixed weights) or as a fully fine-tuneable module.

ent features of the input [29]. In particular, analysing patterns of activation at pooling layers, specific sub-classes of edge or texture detectors can be distinguished.

To determine if the pre-trained speechVGG acts as a set of hierarchically-organized extractors of speech features, we performed a similar analysis to visualize activation patterns across the network. Specifically, we employed a minimal implementation of the so-called ‘deep dream’ technique [30] to our speechVGG trained using 3000 words obtained from 460 hours of speech recordings. Starting from a Gaussian noise input, we optimized it to maximize the mean activation of different layers’ outputs in the network. The images producing the largest mean activation at pooling layers in the subsequent blocks of the pre-trained speechVGG are presented in Fig. 3.

Activation patterns develop a higher level structure with increasing depth of the network. In particular, the output from the first block represents a fine pattern strongly emphasizing edges. The readout from the second block represents a set of regular horizontal bars, which might reflect harmonic structure of speech. In the final fourth and fifth blocks, one can notice the formation of distinct large-scale time-frequency patterns. We speculate that these can represent phonemes or formants. Interestingly, these results may resemble the stages of auditory neural processing [31, 32]. In particular, early subcortical processing emphasizes the fine structure of sound and later cortical stages become tuned to more complex time-frequency features.

### 2.4. Application to transfer learning

We designed speechVGG to extract features from up to 1024-ms-long samples of audio. Features from longer samples can be obtained by averaging the representations of several windows. Due to the modular block architecture, activation of max-pooling layers across the model provide representations of the input, each emphasizing distinct speech-specific features. The highest-level features, obtained by flattening the output of the last max-pooling in the model, provide the smallest, most compact, representation of the input (Fig. 1, red).

The pre-trained speechVGG models 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 [14, 15, 16, 17] (Fig. 2a). In the latter case, the activation of the extractor pooling layers provide rich representations of both training output and the target. The direct loss computed between these two representations (for example  $L_1$ ) is then used to train the main system.

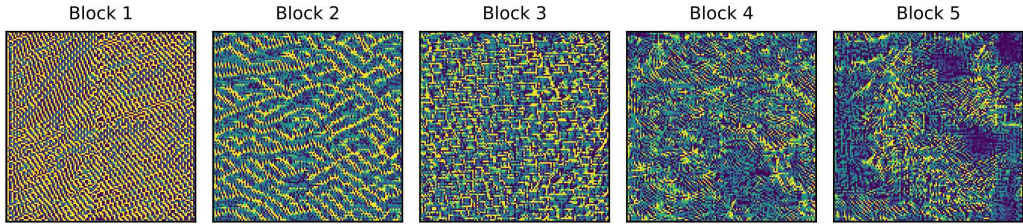


Figure 3: Sample inputs that maximize activation of pooling layers, at the end of each block, of the pre-trained speechVGG.

The pre-trained speechVGG can be applied as a preprocessing stage or a base of a system designed to perform a brand new task. This can be accomplished, by swapping the final output layers of the extractor or attaching other system to the end of the final block of the feature extractor (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 weights of the extractor 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 facilitate the (re-)training process and the overall performance of the trained system.

### 3. Transfer learning experiments

#### 3.1. Speech inpainting

In our previous work we employed the pre-trained speechVGG to train a deep speech inpainting system for recovering missing parts of time-frequency representation of speech [17]. Thereby, the speechVGG was pre-trained on the same task using 1000 words extracted from 100 hours of LibriSpeech data (*train-clean-100*, baseline). Here, we explored how different configurations of speechVGG, outlined in section 2.2, applied in the training, influence the speech inpainting performance.

We used *train-clean-100* from the LibriSpeech dataset to train the inpainting framework and *dev-clean* to evaluate its performance in speech inpainting task. All the speech material was chunked into 1024-ms-long segments and preprocessed in the same way as for the speechVGG training. Each log-magnitude spectrogram was distorted using random time & frequency masks. The masks removed from 10% up to 40% time and frequency bins from the input STFT. Such samples, along with their mask (i.e. the position of time-frequency intrusion was known) were processed through the network to recover the original time-frequency representations of speech. Waveforms were obtained directly from the recovered STFT magnitudes by applying the local weighted sums algorithm [33, 34].

The system for speech inpainting was trained using nine different configurations of speechVGG (section 2.2). The feature extractor was each time applied to obtain feature losses for training the main framework (Fig. 2a). For each training sample, the recovered and the actual samples of speech were processed through the pre-trained speechVGG. The deep feature loss was obtained by computing  $L_1$  between activations of the speechVGG’s pooling layers and used to train the main model. The inpainting performance was quantified using the short term objective intelligibility (STOI) [35] and perceptual evaluation of speech quality (PESQ) [36]. Both scores were computed between the original (i.e. non-distorted) and the recovered speech samples from the held-out portion of the data (*dev-clean*).

**Results:** Evaluation results are reported in table 1. All of the considered configuration of speechVGG succeeded as deep feature extractors for training the system for speech inpainting,

yet their efficacy varied. STOI & PESQ scores indicated that speechVGG pre-trained to classify 3000 words extracted from 460 hours of speech was the optimal configuration for this task. The lack of improvement for larger sizes of either dictionary or training set 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. For the remaining transfer learning experiments, we used the best performing pre-trained speechVGG (3000 words & 460 hours).

Table 1: Impact of the speechVGG configuration on the training of the speech inpainting system [17]. Each cell in the array represents STOI (top) and PESQ (bottom), measured between the recovered and the original speech samples, averaged across all the evaluation data. Baseline scores for unprocessed, corrupted speech samples were 0.664 STOI & 1.675 PESQ.

		Training size (#hours)			Avg.
		100	460	960	
Dictionary size (#words)	1000	0.790 2.334	0.786 2.334	0.789 2.361	0.788 2.343
	3000	0.796 2.403	0.824 2.484	0.801 2.369	0.807 2.419
	6000	0.799 2.387	0.791 2.355	0.793 2.374	0.794 2.372
Avg.		0.795 2.375	0.800 2.391	0.794 2.368	

#### 3.2. Language identification

The language identification experiments were performed using the Spoken Language Identification (SLI) dataset [19] that contains voice recordings in three languages: English, German and Spanish. We used the recommended train/test data split. 20 randomly chosen 1024-ms-long segments were obtained from each recording and preprocessed to obtain their spectrograms, as specified in section 2.2. Each segment was processed through the pre-trained speechVGG to obtain its representation by flattening the output from the last block (Fig. 1, red). A set of features describing the entire recording was each time obtained by averaging representations of its parts. Having processed all the recordings, a logistic regression classifier [37] was trained on the obtained features to distinguish the three classes.

**Results:** In the language identification task we obtained a classification accuracy of **97.6%**. Our model outperformed the similar representation-based approach proposed in Tagliasacchi et al. [13] reporting an accuracy of 90%, as well as, the task-specific ConvNet [19], which achieved 97% accuracy.

### 3.3. Speech, music and noise classification

We used the MUSAN dataset [20] to classify the following three different categories of audio recordings: speech (recordings from the US government and librivox.org), music and noise. We kept randomly selected 10% of the data as the held-out evaluation set. We employed the same setup as in the previous experiment except, here, we discarded all samples shorter than 1024 ms. All such short samples were recordings of noise and including them could lead to the biased classification based solely on their duration, rather than the acoustic content.

**Results:** High-dimensional embeddings of the MUSAN dataset were visualized via t-SNE [38] (Fig. 4). Clusters representing speech recordings were clearly distinguishable from music and noise. This suggests that speechVGG, pre-trained on the LibriSpeech data, successfully transferred the generalized representation of speech to this task. Interestingly, embeddings of speech recordings were divided into two distinct clusters, one made up exclusively of US government recordings (Fig. 4, dark blue). Our approach yielded **96.5%** classification accuracy. Remarkably, the high-dimensional embeddings, obtained from the speechVGG pre-trained on voice recordings, allowed to reliably distinguish samples of music and noise. Tagliasacchi, et al. [13] reported 99.0% accuracy on the MUSAN classification task using representations of 0.975-seconds-long segments of recordings. However, their approach was tailored specifically for this task and each time used the entire audio clip, while here we used only 20 random segments (i.e. up to 20 seconds of audio).

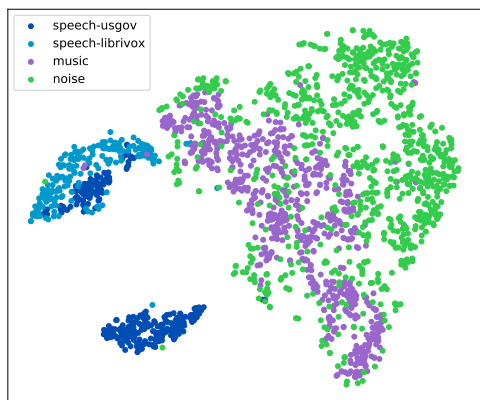


Figure 4: *t*-SNE visualization of high-dimensional embeddings of speech, music and noise recordings from MUSAN, obtained via the pre-trained speechVGG.

### 3.4. Speaker identification

In the speaker identification task, we used speech recordings from the TIMIT dataset including 630 speakers [39]. We randomly selected one recording per speaker to form the 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 2.2. The previously introduced approach did not succeed in this task. Therefore, here, we replaced the final fully-connected layers of the speechVGG to accommodate different number of classes (630) in the new task, and fine-tuned such model in three different ways: (i) we discarded the knowledge obtained in the pre-training and fully re-trained the network on the speaker identification task *speechVGG-fresh*, (ii) we re-trained only the final layers, keeping the previous, pre-

trained blocks unchanged (i.e. frozen) *speechVGG-frozen* and (iii) we fine-tuned the entire pre-trained model using the new data *speechVGG-finetune*. For fine tuning the model, we used the same training routines as in the speechVGG pre-training, but using the task-specific data. In the model evaluation, the speaker identity was determined by averaging the predictions from a window (50% overlap) sliding over the entire recording.

**Results:** Results of speaker identification using different variants of fine-tuned the speechVGG, alongside the baseline approaches are presented in Table 2. *speechVGG-fresh* failed to converge; *speechVGG-frozen* converged but overfitted to the data, failing to generalize; *speechVGG-finetune* converged and achieved **99.7%** accuracy on the set-aside portion of the data. The latter approach outperforms existing methods evaluated on the TIMIT corpus [21, 22, 23]. Notably, Ge et al. (2017) [24] reported 100% accuracy, also employing one second long window, but only using a subset of 100 male speakers what makes a direct comparison difficult.

Table 2: *TIMIT speaker identification. Single window - label assigned for each window separately. Sliding window - label assigned by averaging predictions from a window sliding over the entire recording. \* - the largest observed overfitting.*

Method	Single window		Sliding window
	train	valid	valid
speechVGG-fresh	-	-	-
speechVGG-frozen	99.3%*	0.7%	1/630 (0.2%)
speechVGG-finetune	97.6%	95.1%	<b>628/630 (99.7%)</b>
Ward, et al. (1998) [21]			607/630 (96.3%)
Ming, et al. (2007) [22]			608/630 (96.5%)
Wildermoth, et al. (2003) [23]			623/630 (99.0%)

## 4. Conclusion

Here, we introduced speechVGG, a trainable deep speech features extractor tailored for transfer learning in speech processing problems. We showed that the speechVGG can capture speech-specific features in a hierarchical fashion (Fig. 3). Importantly, the generalized representation of speech captured by the pre-trained model was transferable over four distinct speech processing tasks, each employing a different dataset. Notably, relatively simple applications of the pre-trained speechVGG were capable of achieving, to the best of authors' knowledge, results comparable to the state-of-the-art (section 3).

We hope that this work will facilitate further applications of transfer learning in the field audio- and speech processing and the development of new approaches. In particular, the exploration of different architectures, as well as, supervised and unsupervised training setups should allow to extend the method. Notably, multiple speech feature extractors sensitive to distinct features could be gathered together to form ensembles and maximize their efficacy. For example, training the deep learning frameworks via deep feature losses doesn't have to be limited to only one feature extractor. There can be many, each tailored to capture representations of different aspects of speech, such as generalized representations of language, speakers or different linguistic units across timescales.

Python implementation of the speechVGG and pre-trained models are openly available at<sup>1</sup>.

## 5. Acknowledgements

This work was funded by Logitech & Imperial College London.

<sup>1</sup><https://github.com/bepierre/SpeechVGG>

## 6. References

- [1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [2] 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*. Springer, 2018, pp. 270–279.
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [4] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp. 17–36.
- [5] 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.
- [6] 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-Volume 27*, 2011, pp. 97–111.
- [7] 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 (ASRU)*. IEEE, 2017, pp. 279–286.
- [8] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in neural information processing systems*, 2016, pp. 658–666.
- [9] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7304–7308.
- [10] 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.
- [11] A. M. Liberman and I. G. Mattingly, "The motor theory of speech perception revised," *Cognition*, vol. 21, no. 1, pp. 1–36, 1985.
- [12] F. d. C. Quitry, M. Tagliasacchi, and D. Roblek, "Learning audio representations via phase prediction," *arXiv preprint arXiv:1910.11910*, 2019.
- [13] M. Tagliasacchi, B. Gfeller, F. d. C. Quitry, and D. Roblek, "Self-supervised audio representation learning for mobile devices," *arXiv preprint arXiv:1905.11796*, 2019.
- [14] F. Germain, Q. Chen, and V. Koltun, "Speech Denoising with Deep Feature Losses," in *Proc. Interspeech 2019*, 2019, pp. 2723–2727.
- [15] A. Sahai, R. Weber, and B. McWilliams, "Spectrogram feature losses for music source separation," *arXiv preprint arXiv:1901.05061*, 2019.
- [16] C.-F. Liao, Y. Tsao, X. Lu, and H. Kawai, "Incorporating Symbolic Sequential Modeling for Speech Enhancement," in *Proc. Interspeech 2019*, 2019, pp. 2733–2737.
- [17] M. Kegler, P. Beckmann, and M. Cernak, "Deep speech inpainting of time-frequency masks," *arXiv preprint arXiv:1910.09058*, 2019.
- [18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.
- [19] T. Oponowicz, "Spoken language identification," 2018. [Online]. Available: <https://www.kaggle.com/toponowicz/spoken-language-identification>
- [20] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.
- [21] N. C. Ward and D. R. Dersch, "Text-independent speaker identification and verification using the timit database," in *Fifth International Conference on Spoken Language Processing*, 1998.
- [22] 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.
- [23] B. R. Wildermoth and K. K. Paliwal, "Gmm based speaker recognition on readily available databases," in *Microelectronic Engineering Research Conference, Brisbane, Australia*, vol. 7, 2003, p. 55.
- [24] Z. Ge, A. N. Iyer, S. Cheluvareja, R. Sundaram, and A. Ganapathiraju, "Neural network based speaker classification and verification systems with enhanced features," in *2017 Intelligent Systems Conference (IntelliSys)*. IEEE, 2017, pp. 1089–1094.
- [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 (ICASSP)*. IEEE, 4 2015, pp. 5206–5210.
- [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] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [29] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.
- [30] A. Mordvintsev, C. Olah, and M. Tyka, "Deepdream - a code example for visualizing neural networks," *Google Research*, vol. 2, no. 5, 2015. [Online]. Available: <https://ai.googleblog.com/2015/07/deepdream-code-example-for-visualizing.html>
- [31] T. Chi, P. Ru, and S. A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *The Journal of the Acoustical Society of America*, vol. 118, no. 2, pp. 887–906, 2005.
- [32] D. L. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature neuroscience*, vol. 19, no. 3, p. 356, 2016.
- [33] 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)*, Sep. 2010, pp. 397–403.
- [34] —, "Phase initialization schemes for faster spectrogram-consistency-based signal reconstruction," in *Proc. Acoustical Society of Japan Autumn Meeting (ASJ)*, no. 3-10-3, Mar. 2010.
- [35] C. Taal, R. 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*. IEEE, 2010, pp. 4214–4217.
- [36] "Methods for subjective determination of transmission quality," *ITU-T Rec. P.800*, 1998.
- [37] 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.
- [38] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [39] 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.