

# Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks

Santiago Pascual<sup>1</sup>, Mirco Ravanelli<sup>2</sup>, Joan Serra<sup>3</sup>, Antonio Bonafonte<sup>1\*</sup>, Yoshua Bengio<sup>2,4</sup>

<sup>1</sup>Universitat Politècnica de Catalunya, <sup>2</sup>Mila - Université de Montréal,

<sup>3</sup>Telefónica Research, <sup>4</sup>CIFAR Fellow

santi.pascual@upc.edu, mirco.ravanelli@gmail.com

## Abstract

Learning good representations without supervision is still an open issue in machine learning, and is particularly challenging for speech signals, which are often characterized by long sequences with a complex hierarchical structure. Some recent works, however, have shown that it is possible to derive useful speech representations by employing a self-supervised encoder-discriminator approach. This paper proposes an improved self-supervised method, where a single neural encoder is followed by multiple workers that jointly solve different self-supervised tasks. The needed consensus across different tasks naturally imposes meaningful constraints to the encoder, contributing to discover general representations and to minimize the risk of learning superficial ones. Experiments show that the proposed approach can learn transferable, robust, and problem-agnostic features that carry on relevant information from the speech signal, such as speaker identity, phonemes, and even higher-level features such as emotional cues. In addition, a number of design choices make the encoder easily exportable, facilitating its direct usage or adaptation to different problems.

**Index Terms:** speech representation, speech classification, transfer learning, self-supervised learning.

## 1. Introduction

The success of deep learning techniques strongly depends on the quality of the representations that are automatically discovered from data. These representations should capture intermediate concepts, features, or latent variables, and are commonly learned in a supervised way using large annotated corpora. Even though this is still the dominant paradigm, some crucial limitations arise. Collecting large amounts of annotated examples, for instance, is very costly and time-consuming. Moreover, if not learned with a large pool of tasks [1], supervised representations are likely to be biased towards the considered problem, limiting their exportability to other problems and applications [2].

A natural way to mitigate these issues is unsupervised learning [3]. Unsupervised learning attempts to extract knowledge from unlabeled data, and can potentially discover representations that capture the underlying structure of such data. Several approaches have been proposed for unsupervised learning in the last decade. Notable examples are deep autoencoders [4] and restricted Boltzmann machines [5], which can be employed as a pre-training step for a subsequent supervised task like speech recognition [6]. More recent techniques include variational autoencoders [7] and generative adversarial networks [8].

A related sub-field that is gaining popularity, especially within the computer vision community, is self-supervised learning, where targets are computed from the signal itself [9–11].

This is often performed by applying known transforms or sampling strategies to the input data and using the resulting outcomes as targets. Some attempts have also been done to extend self-supervised learning to different modalities [12, 13] or to audio representations only [14–17]. With this regard, a recent trend consists of learning speech representations using a neural network encoder followed by a binary discriminator [16–18].

Despite recent progress, applying self-supervised learning to speech remains challenging. Speech signals are not only high-dimensional, long, and variable-length sequences, but also entail a complex hierarchical structure that is difficult to infer without supervision (phonemes, syllables, words, etc.). It is thus hard to find a single self-supervised task that can learn general and meaningful representations able to capture this latent structure.

To mitigate this issue, we propose to jointly tackle multiple self-supervised tasks using an ensemble of neural networks that cooperate to discover good speech representations. The intuition is that each self-supervised task may bring a different view or soft constraint on the learned representation. Even though not all the self-supervised tasks may help for the supervised problem of interest, there is likely a subset of them that could be useful. Another important implication is that our approach requires consensus across tasks, imposing several constraints into the learned representations. This way, our approach is more likely to learn general, robust, and transferable features, and less likely to focus on superficial features of the signal which may be sufficient for the given training data but are insufficient when considering broader types of data. To highlight the latter property, we call our proposed architecture the problem-agnostic speech encoder (PASE). PASE encodes the raw speech waveform into a representation that is fed to multiple regressors and discriminators. Regressors deal with standard features computed from the input waveform, resembling a decomposition of the signal at many levels. Discriminators deal with either positive or negative samples and are trained to separate them by minimizing binary cross-entropy [17]. Both regressors and discriminators (hereinafter called *workers*) contribute to add prior knowledge into the encoder, which turns out to be crucial to derive meaningful and robust representations.

Our experiments suggest that PASE is able to discover robust representations from the raw speech waveform directly. We find that such representations outperform more traditional hand-crafted features in different speech classification tasks such as speaker identification, emotion classification, and automatic speech recognition. Interestingly, even though our representations are learned from a clean data set, the derived features turn out to work well also when processing speech that is corrupted by a considerable amount of noise and reverberation. PASE is designed to be efficient and fully parallelizable, and it can be seen as a first step towards a universal speech feature extractor. Moreover, PASE can be used as a

\*A. Bonafonte is currently at Amazon Research, Cambridge, UK.

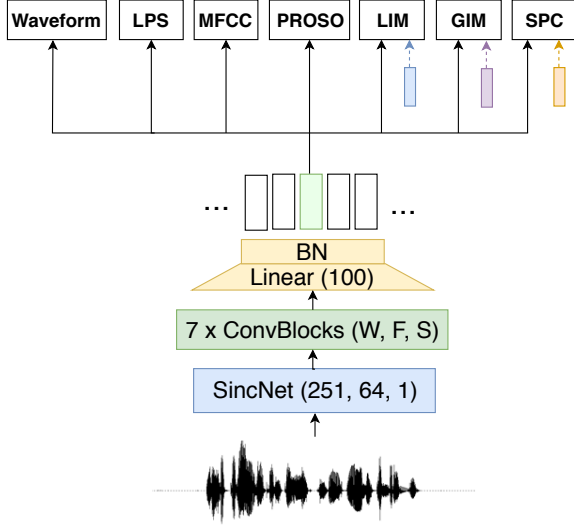


Figure 1: The PASE architecture, with the considered workers.

pre-trained network avoiding to train models from scratch for each new task, as commonly done for computer vision models [19, 20]. PASE code and pre-trained model are available from <https://github.com/santi-pdp/pase>.

## 2. Problem-agnostic Speech Encoder

The PASE architecture, depicted in Figure 1, is composed of a fully-convolutional speech encoder, followed by seven multilayer perceptron (MLP) workers, which cooperatively solve different self-supervised tasks. We now describe these modules.

### 2.1. Encoder

The first layer of the encoder is based on the recently-proposed SincNet model [21]. SincNet performs the convolution of the raw input waveform with a set of parameterized sinc functions that implement rectangular band-pass filters. An interesting property of SincNet is that the number of parameters does not increase with the kernel size. Similarly to [21, 22], we use a large kernel width  $W = 251$  to implement  $F = 64$  filters with a stride  $S = 1$ . The subsequent layers are composed of a stack of 7 convolutional blocks (Fig. 1). Each block employs a one-dimensional convolution, followed by batch normalization (BN) [23], and a multi-parametric rectified linear unit (PReLU) activation [24]. For the 7 blocks we use kernel widths  $W = \{20, 11, 11, 11, 11, 11, 11\}$ ,  $F = \{64, 128, 128, 256, 256, 512, 512\}$  filters, and strides  $S = \{10, 2, 1, 2, 1, 2, 2\}$ . An additional layer performs a convolution with  $W = 1$  that projects 512 features to embeddings of dimension 100. The final PASE representation is produced by a non-affine BN layer that normalizes by the mean and variance of each dimension.

Note that, similarly to common speech feature extractors based on the short-time Fourier transform, we emulate an overlapping sliding window using a set of convolutions. The convolution, in fact, employs a sliding kernel over the signal that extracts localized patterns at different time shifts. In our case, we use stride factors  $S > 1$  for most of the convolutional blocks, such that the input signal is decimated in time by a factor of 160. Therefore, given an input waveform of  $T$  samples, the amount

of output feature vectors (frames) is  $N = \frac{T}{160}$ . At 16 kHz, this is equivalent to a 10 ms stride, similar to common speech processing pipelines. The receptive field of the encoder is about 150 ms.

### 2.2. Workers

Workers are fed by the encoded representation and solve seven self-supervised tasks, defined as regression or binary discrimination tasks (Fig. 1). In all cases, workers are based on very small feed-forward networks, composed of a single hidden layer of 256 units with PReLU activation (the only exception is the waveform worker, see below). Notice that we here employ simple networks on purpose. This way, we encourage the encoder, and not the workers, to discover high-level features that can be successfully exploited even by classifiers with limited capacity.

We first consider the use of regression workers, which break down the signal components at many levels in an increasing order of abstraction. These workers are trained to minimize the mean squared error (MSE) between the target features and the network predictions (again the waveform worker is an exception, see below). Features are extracted with librosa [25] and pysptk [26] using default parameters, if not stated otherwise. As regression workers we consider:

- **Waveform:** we predict the input waveform in an auto-encoder fashion. The waveform decoder employs three deconvolutional blocks with strides 4, 4, and 10 that upsample the encoder representation by a factor of 160. After that, an MLP of 256 PReLU units is used with a single output unit per time-step. This worker learns to reconstruct waveforms by means of mean absolute error (L1) minimization. The choice of L1 is driven by robustness, as the speech distribution is very peaky and zero-centered with prominent outliers [27].
- **Log power spectrum (LPS):** as with the next features, we compute it using a Hamming window of 25 ms and a step size of 10 ms, with 1025 frequency bins per time step.
- **Mel-frequency cepstral coefficients (MFCC):** we extract 20 coefficients from 40 mel filter banks (FBANKS).
- **Prosody:** we also predict four basic features per frame, namely the interpolated logarithm of the fundamental frequency, voiced/unvoiced probability, zero-crossing rate, and energy. These features are called “Prosody”, inheriting a terminology often used in emotion recognition [28, 29].

Next, we also consider three binary discrimination tasks, learning a higher level of abstraction than that of signal features. These tasks rely on a pre-defined sampling strategy that draws an anchor  $x_a$ , a positive  $x_p$ , and a negative  $x_n$  sample from the pool of PASE-encoded representations available in the training set. The reference anchor  $x_a$  is an encoded feature extracted from a random sentence, while  $x_n$  and  $x_p$  are encodings drawn using the different sampling strategies described below. An MLP then minimizes the following formulation of the binary cross-entropy:

$$L = \mathbb{E}_{X_p} [\log(g(x_a, x_p))] + \mathbb{E}_{X_n} [\log(1 - g(x_a, x_n))],$$

where  $g$  is the discriminator function, and  $\mathbb{E}_{X_p}$  and  $\mathbb{E}_{X_n}$  denote the expectation over positive and negative samples, respectively. Intuitively, by minimizing  $L$ , the model learns a speech embedding such that positive examples end up closer to their anchors than the corresponding negatives. Notice that the encoder and the discriminators are not adversarial here, but must cooperate to derive good representations. In this work, we explore the following approaches to sample positive and negative examples:



- **Local info max (LIM):** as proposed in [17], we draw the positive sample from the same sentence of the anchor and a negative sample from another random sentence that likely belongs to a different speaker. Since the speaker identity is a reliable constant factor within random features of the same sentence, this worker can learn a representation that embeds this kind of information.



- **Global info max (GIM):** in this and the subsequent worker, we compare global representations rather than local ones. The anchor representation is obtained by averaging all the PASE-encoded frames of a random utterance within a long random chunk of 1 s. The positive sample is similarly derived from another random chunk within the same sentence, while the negative one is obtained from another sentence. This way, we encourage the encoder to learn representations containing high-level information on the input sequence, that are hopefully complementary to those learned by LIM. GIM is also related to Deep InfoMax [18], which recently proposed to exploit local and global samples to learn image representations.



- **Sequence predicting coding (SPC):** in this case, the anchor is a single frame, while positive and negative samples are randomly extracted from its future and past elements. In particular,  $x_p$  contains 5 consecutive future frames, while  $x_n$  gathers 5 consecutive past ones. To make the task less trivial, we avoid sampling inside the current-frame receptive field (150 ms). On the other hand, to avoid making this task too complex or even unfeasible, we sample up to 500 ms away from the anchor. We expect this worker to capture information about the sequential order of the frames and the signal causality, encouraging PASE to embed a longer time contextual information. This approach is similar to the sampling strategy used in the contrastive predicting coding work [30]. The main difference is that our negative sample is extracted from the past of the same sentence, rather than coming from a different one.

### 2.3. Self-supervised Training

Encoder and workers are jointly trained with backpropagation by optimizing a total loss that is computed as the average of each worker cost. Within the encoder, the gradients coming from the workers are thus averaged as well, and the optimization step will update its parameters pointing to a direction that is a compromise among all the worker losses [1]. To balance the contribution of each regression loss, we standardize all worker outputs using their mean and variance train set statistics, before computing the MSE. The encoder and the workers are optimized with Adam [31], using an initial learning rate of  $5 \cdot 10^{-4}$  which is halved every 30 epochs. We use mini-batches of 32 waveform chunks, each with 16 k samples corresponding to 1 s at a 16 kHz sampling rate. The system is trained for 150 epochs (i.e., until the validation losses reach a plateau for all the workers).

### 2.4. Usage in Supervised Classification Problems

The representations discovered by the encoder can be later used for supervised classification in different ways. One possibility is to keep the encoder frozen while training the classifier (PASE-Frozen). The encoder is thus used as a standard feature extractor and the features do not dynamically change during training. A better way consists of fine-tuning both the encoder and classifier during supervised training (PASE-FineTuned). This way, the extracted features are further optimized to better adapt themselves to the application of interest. For comparison, our results also

include the case where PASE is trained on the supervised task from scratch, with random initialization (PASE-Supervised).

## 3. Corpora and Tasks

The self-supervised training of PASE is performed with the portion of the LibriSpeech dataset [32] used in [17]. Speech sentences have been randomly selected to exploit about 15 s of training material for each of the 2484 speakers.

To assess the quality of the learned representations, we consider three supervised problems: (1) speaker identification (Speaker-ID), (2) speech emotion classification (Emotion), and (3) automatic speech recognition (ASR). For speaker identification, we use the VCTK dataset [33], which contains 109 speakers with different English accents. To make this task more challenging and realistic, we consider a subset of it that only contains 11 s of training for each speaker. For emotion recognition, we use the English utterances of the INTERFACE dataset [34]. This corresponds to approximately 3 h for training, 40 min for validation, and 30 min for test. For speaker and emotion recognition, the neural posterior probabilities are averaged over all the time frames and we take the class with the highest score. To evaluate the capability of PASE to learn phoneme representations, a first set of ASR experiments is performed with the standard TIMIT dataset [35]. Next, to assess our approach in more challenging noisy and reverberant conditions, in Section 4.3 we use the DIRHA dataset [36]. Training and validation sets are based on the original WSJ-5k corpus (consisting of 7138 sentences uttered by 83 speakers) that is contaminated with a set of impulse responses measured in a real apartment. The test set is composed of 409 WSJ sentences uttered by six American speakers and is based on real recordings in a domestic environment with a reverberation time of 0.7 s and an average signal-to-noise ratio of about 10 dB. ASR experiments are performed with the PyTorch-Kaldi toolkit [37] and are based on the DNN-HMM framework. The DNN is trained to predict context-dependent phones and an HMM decoder is later employed to retrieve the sequence of phonemes for TIMIT or words for DIRHA (using the language models of the Kaldi recipes [38]).

## 4. Results

### 4.1. Worker Ablation

First of all, we study whether all considered workers contribute to the final accuracy of PASE, and assess their impact on different target problems. To do so, we retrain the encoder discarding one of the workers at a time. We then extract PASE features (using the frozen encoder described in section 2.4), and we use them to feed MLP classifiers that solve the considered supervised problems. The experiments in this section are conducted with simple MLP classifiers based on a single layer, except for ASR, where we use three layers.

The classification accuracies of Table 1 show that no worker is dispensable. The best results are achieved with all workers, and we never observe performance improvements when discarding any of them. Nevertheless, while some workers are helpful for all the speech tasks, the benefits of some others turn out to be more application-dependent. For instance, Waveform, LPS, and MFCC regressors are generally helpful for all the applications, since they force the encoded representation to retain low-level information of the speech signal itself. The MFCC worker, in particular, is the most crucial one since it injects valuable prior knowledge on the most important frequency bands of the speech

Table 1: Accuracies using PASE and an MLP as classifier. Rows below the “all workers” model report absolute accuracy loss when discarding each worker for self-supervised training.

| Model              | Classification accuracy [%] |                        |                |
|--------------------|-----------------------------|------------------------|----------------|
|                    | Speaker-ID<br>(VCTK)        | Emotion<br>(INTERFACE) | ASR<br>(TIMIT) |
| PASE (All workers) | 97.5                        | 88.3                   | 81.1           |
| – Waveform         | –1.3                        | –3.9                   | –0.3           |
| – LPS              | –1.5                        | –5.3                   | –0.5           |
| – MFCC             | –2.4                        | –3.2                   | –0.7           |
| – Prosody          | –0.5                        | –5.3                   | –0.1           |
| – LIM              | –0.8                        | –1.3                   | –0.0           |
| – GIM              | –0.6                        | –0.5                   | –0.3           |
| – SPC              | –0.4                        | –1.6                   | –0.0           |

sequence. The prosody worker, instead, has a remarkable and expectable impact on emotion recognition only (+131% in relative error). This is due to the fact that our prosody features are correlated with intonation, expressiveness, and voicing, which are crucial clues for detecting emotion. LIM and GIM seem to be more helpful for Speaker-ID and Emotion rather than for ASR. These workers are designed to extract high-level information of speech that can be better exploited by higher-level classification tasks. A similar trend is observed for the SPC worker. This tends to extract longer contextual information, which turns out to be helpful for speaker and emotion recognition (+16% and in +13% relative error, respectively). The adopted receptive field of 150 ms, instead, embeds a context large enough for a DNN-HMM ASR system, as observed in [39].

#### 4.2. Comparison with Standard Features

We now compare our PASE representations with more standard features such as MFCCs and FBANK [25]. Despite being proposed more than 40 years ago [40], these coefficients are still the most common speech features, and it is not easy to find alternatives that consistently outperform them. To provide a more fair comparison, MFCCs and FBANK are gathered in context windows that embed contextual information of about 150 ms (similar to the receptive field of the encoder). MFCCs are also augmented with their first and second derivatives. As mentioned, we also compare with the purely supervised version of PASE, trained from scratch on the target task.

Table 2 shows the classification accuracies obtained with both MLP and recurrent neural network (RNN) classifiers based on gated recurrent units (GRU) [41]. The hyperparameters of all classifiers (number of hidden layers and neurons, learning rate, batch sizes, dropout rates, etc.) are independently tuned on the validation set and for each problem. PASE features provide most of the times a performance better than MFCCs and FBANKs, even when freezing the encoder (PASE-Frozen). The performance improvement becomes more evident when pre-training the encoder and fine-tuning it with the supervised task of interest (PASE-FineTuned). This approach consistently provides the best performance over all the tasks and classifiers considered here. Our best Speaker-ID result compares favorably with some recent works on the same dataset, such as [42] and [43]. Interestingly, our best emotion recognition system achieves an accuracy of 97.7%, which outperforms the human-level performance (80%) measured in [34] for the whole INTERFACE corpus. The phoneme accuracy of 85.3% on the TIMIT dataset (an error rate of 14.7%) is a competitive performance as well, especially

Table 2: Accuracy comparison on the considered classification tasks using MLPs and RNNs as classifiers.

| Model           | Classification accuracy [%] |             |                        |             |                |             |
|-----------------|-----------------------------|-------------|------------------------|-------------|----------------|-------------|
|                 | Speaker-ID<br>(VCTK)        |             | Emotion<br>(INTERFACE) |             | ASR<br>(TIMIT) |             |
|                 | MLP                         | RNN         | MLP                    | RNN         | MLP            | RNN         |
| MFCC            | 96.9                        | 72.3        | 90.8                   | 91.1        | 81.1           | 84.8        |
| FBANK           | 98.4                        | 75.1        | 94.1                   | 92.8        | 80.9           | 85.1        |
| PASE-Supervised | 97.0                        | 80.5        | 93.8                   | 92.8        | 82.1           | 84.7        |
| PASE-Frozen     | 97.3                        | 82.5        | 91.5                   | 92.8        | 81.4           | 84.7        |
| PASE-FineTuned  | <b>99.3</b>                 | <b>97.2</b> | <b>97.7</b>            | <b>97.0</b> | <b>82.9</b>    | <b>85.3</b> |

Table 3: Word error rate (WER) obtained on the DIRHA corpus.

|                 | WER [%]     |
|-----------------|-------------|
| MFCC            | 35.8        |
| FBANK           | 34.0        |
| PASE-Supervised | 33.5        |
| PASE-Frozen     | 32.5        |
| PASE-FineTuned  | <b>29.8</b> |

when compared to state-of-the-art results that do not use complex techniques as system combination, speaker adaptation, or multiple steps of lattice rescoring and decoding [37, 38, 44, 45].

#### 4.3. Transferability

Finally, we study the exportability of PASE to acoustic conditions that are very different from the clean one used to train it. Table 3 reports the results obtained with the DIRHA dataset, which contains speech signals characterized by considerable noise and reverberation. We here employ the same version of PASE encoder used so far (trained on clean LibriSpeech data) coupled with a GRU classifier. Interestingly, PASE clearly outperforms the other systems. Even the frozen version of PASE overtakes FBANKs, MFCCs, and the supervised training baseline. PASE-FineTuned also outperforms our previous results obtained with the standard SincNet model [22]. This result suggests the ability of PASE to effectively transfer its representation abstractions to different acoustic scenarios.

## 5. Conclusion

The proposal of this work was twofold. On the one hand, we proposed a multi-task self-supervised approach to learn speech representations. On the other hand, we provided an effective and exportable speech encoder that conveys waveforms into a sequence of latent embeddings. As evidenced by the considered problems, the discovered embeddings turn out to carry important information of the speech signal, related to, at least, speaker-identity, phonemes, and emotional cues. Learnt embeddings also showed their potential for transferability to different datasets, tasks, and acoustic conditions. Moreover, PASE is easily extendable as a semi-supervised framework and can embed in the future many other self-supervised tasks.

## 6. Acknowledgements

This research was partially supported by the project TEC2015-69266-P (MINECO/FEDER, UE), Calcul Québec, and Compute Canada. We also thank Loren Lugosch, Titouan Parcollet, and Maurizio Omologo for helpful comments.

## 7. References

- [1] J. Serrà, S. Pascual, and A. Karatzoglou, “Towards a universal neural network encoder for time series,” in *Artificial Intelligence Research and Development*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, 2018, vol. 308, pp. 120–129.
- [2] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, “To transfer or not to transfer,” *NIPS Workshop in Inductive Transfer: 10 Years Later*, 2005.
- [3] Y. Bengio, “Deep learning of representations for unsupervised and transfer learning,” in *Proc. of the ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 17–36.
- [4] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Proc. of NIPS*, 2006.
- [5] G. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, pp. 1527–1554, 2006.
- [6] G. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large vocabulary speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
- [7] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *Proc. of ICLR*, 2014.
- [8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proc. of NIPS*, 2014.
- [9] C. Doersch and A. Zisserman, “Multi-task self-supervised visual learning,” in *Proc. of ICCV*, 2017, pp. 2070–2079.
- [10] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” *Proc. of ICLR*, 2018.
- [11] I. Misra, C. L. Zitnick, and M. Hebert, “Shuffle and learn: Unsupervised learning using temporal order verification,” in *Proc. of ECCV*, 2016, pp. 527–544.
- [12] R. Arandjelovic and A. Zisserman, “Objects that sound,” in *Proc. of ECCV*, 2018, pp. 451–466.
- [13] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, “Learning sight from sound: Ambient sound provides supervision for visual learning,” *International Journal of Computer Vision*, vol. 126, no. 10, pp. 1120–1137, 2018.
- [14] A. Jansen, M. Plakal, R. Pandya, D. P. W. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, “Unsupervised learning of semantic audio representations,” in *Proc of ICASP*, 2018, pp. 126–130.
- [15] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, “Unsupervised speech representation learning using wavenet autoencoders,” *CoRR*, vol. abs/1901.08810, 2019.
- [16] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *Arxiv: 1807.03748*, 2018.
- [17] M. Ravanelli and Y. Bengio, “Learning speaker representations with mutual information,” *Arxiv: 1812.00271*, 2018.
- [18] D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” *Proc. of ICLR*, 2018.
- [19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Proc. of ICLR*, 2015.
- [20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. of CVPR*, 2016, pp. 2818–2826.
- [21] M. Ravanelli and Y. Bengio, “Speaker recognition from raw waveform with SincNet,” *Proc. of SLT*, 2018.
- [22] M. Ravanelli and Y. Bengio, “Interpretable convolutional filters with SincNet,” *NIPS Workshop on Interpretability and Robustness for Audio, Speech and Language (IRASL)*, 2018.
- [23] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. of ICML*, 2015, pp. 448–456.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. of ICCV*, 2015, pp. 1026–1034.
- [25] B. McFee *et al.*, “librosa/librosa: 0.6.3,” Feb. 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.2564164>
- [26] R. Yamamoto, J. Felipe, and M. Blaauw, “r9y9/pysptk: 0.1.14,” Jan. 2019. [Online]. Available: <https://github.com/r9y9/pysptk>
- [27] S. Pascual, A. Bonafonte, and J. Serrà, “Segan: Speech enhancement generative adversarial network,” in *Proc. of Interspeech*, 2017, pp. 3642–3646.
- [28] M. Neumann and N. T. Vu, “Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech,” in *Proc. of Interspeech*, 2017, pp. 1263–1267.
- [29] W. S. A. Paeschke, M. Kienast, “F0-contours in emotional speech,” in *Proc. of ICPh*, 1999, pp. 929–932.
- [30] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in *Proc. of NIPS*, 2017.
- [31] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. of ICLR*, 2015.
- [32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. of ICASSP*, 2015, pp. 5206–5210.
- [33] C. Veaux, J. Yamagishi, and K. MacDonald, “CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit,” 2016.
- [34] V. Hozjan, Z. Kacic, A. Moreno, A. Bonafonte, and A. Nogueiras, “Interface databases: Design and collection of a multilingual emotional speech database,” in *Proc. of the Int. Conf. on Language Resources and Evaluation (LREC)*, 2002, p. 174.
- [35] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, “DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM,” 1993.
- [36] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo, “The DIRHA-ENGLISH corpus and related tasks for distant-speech recognition in domestic environments,” in *Proc. of ASRU 2015*, pp. 275–282.
- [37] M. Ravanelli, T. Parcollet, and Y. Bengio, “The PyTorch-Kaldi Speech Recognition Toolkit,” in *Proc. of ICASSP*, 2019.
- [38] D. Povey *et al.*, “The Kaldi Speech Recognition Toolkit,” in *Proc. of ASRU*, 2011.
- [39] M. Ravanelli and M. Omologo, “Automatic context window composition for distant speech recognition,” *Speech Communication*, vol. 101, pp. 34–44, 2018.
- [40] S. B. Davis and P. Mermelstein, “Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [41] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *Proc. of NIPS*, 2014.
- [42] J. Wang, K. Wang, M. Law, F. Rudzicz, and M. Brudno, “Centroid-based deep metric learning for speaker recognition,” in *Proc. of ICASSP*, 2019.
- [43] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. J. Witbrock, M. Hasegawa-Johnson, and T. S. Huang, “Dilated recurrent neural networks,” in *Proc. of NIPS*, 2017.
- [44] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Light gated recurrent units for speech recognition,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, April 2018.
- [45] J. Michálek and J. Vanek, “A survey of recent DNN architectures on the TIMIT phone recognition task,” in *TSD*, ser. Lecture Notes in Computer Science, vol. 11107. Springer, 2018, pp. 436–444.