SINCNET BASED SPEAKER RECOGNITION ON VOXCELEB

Akshay Pakhle, UNI - avp2131

Columbia University, NY

ABSTRACT

Deep learning is progressively gaining popularity as a viable alternative to i-vectors for speaker recognition. Promising results have been recently obtained with Convolutional Neural Networks (CNNs) when fed by raw speech samples directly. Rather than employing standard hand-crafted features, the latter CNNs learn low-level speech representations from waveforms, potentially allowing the network to better capture important narrow-band speaker characteristics such as pitch and formants. Proper design of the neural network is crucial to achieve this goal. A considerable amount of research has been carried out on this front, with the use of SincNets that encourages the first convolutional layer to discover more meaningful filters. SincNets are based on parameterized sinc functions which implement band-pass filters. In contrast to standard CNNs, that learn all elements of each filter, only low and high cutoff frequencies are directly learned from data with the proposed method. This offers a very compact and efficient way to derive a customized filter bank specifically tuned for the desired application. [1]

Index Terms— SincNets, Xvectors, TDNN, VoxCeleb2

1. PROBLEM DESCRIPTION

This technique has previously been tested on corpora such as TIMIT and LibriSpeech, which are considerably clean, generated under controlled conditions, and hence have the absence of external noisy factors that are present in most audio samples. Their experiments, conducted on both speaker identification and speaker verification tasks, show that the proposed architecture converges faster and performs better than a standard CNN on raw wave-forms.

The objective of this paper is to analyse the performance of Speaker Recognition from raw waveforms using SincNets, under noisy and unconstrained conditions. The main aim is to obtain results using SincNets, while essentially following the same proposed architecture as in [1], in order to allow for comparisons to the results without the use of Sincnets on the Voxceleb Corpus and the TIMIT and LIBRISPEECH corpus experiments.

2. INTRODUCTION

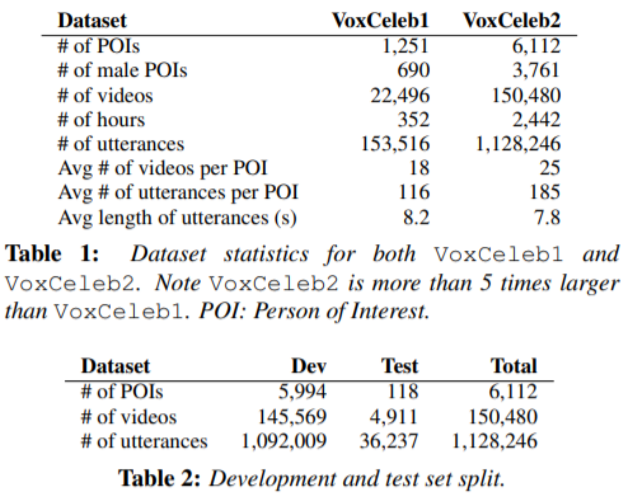
Speaker recognition is a very active research area with notable applications in various fields such as biometric authentication, forensics, security, speech recognition, and speaker diarization, which has contributed to steady interest towards this discipline [2] .

Traditional methods: For a long time, i-vectors have been employed in the field of speaker recognition[3], classified using techniques such as heavy-tailed PLDA [4] and GaussPLDA [5]. While defining the state-of-the-art for a long time, such methods are disadvantaged by their reliance on handcrafted feature engineering. An in-depth review of these traditional methods is given in [6]. Deep learning methods. The success of deep learning in computer vision and speech recognition has motivated the use of deep neural networks (DNN) as feature extractors combined with classifiers, though not trained end-to-end . While such fusion methods are highly effective, they still require hand-crafted engineering. In contrast, CNN architectures can be applied directly to raw spectrograms and trained in an end-to-end manner. For example,[7] uses a Siamese feedforward DNN to discriminatively compare two voices, however this relies on pre-computed MFCC features, whilst [8] also learns the features instead of using MFCCs. The most relevant to our work is [9], who train a neural embedding system using the triplet loss. However, they use private internal datasets for both training and evaluation, and hence a direct comparison with their work is not possible.

Most of past attempts, however, employed hand-crafted features such as FBANK and MFCC coefficients [10] [11] [12]. These engineered features are originally designed from perceptual evidence and there are no guarantees that such representations are optimal for all speech-related tasks. CNNs have recently become the most popular architecture for processing raw speech samples, since weight sharing, local filters, and pooling help discover robust and invariant representations. The authors of [1] believe that one of the most critical part of current waveform-based CNNs is the first convolutional layer. This layer not only deals with high-dimensional inputs, but is also more affected by vanishing gradient problems, especially when employing very deep architectures. The filters learned by the CNN often take noisy and incongruous multi-band shapes, especially when few training samples are available. These filters certainly make some sense for the neural network, but do not appeal to human intuition, nor appear to lead to an efficient representation of the speech signal.

To help the CNNs discover more meaningful filters in the input layer, they have proposed to add some constraints on their shape. Compared to standard CNNs, where the filterbank characteristics depend on several parameters (each element of the filter vector is directly learned), the SincNet convolves the waveform with a set of parametrized sinc functions that implement band-pass filters. The low and high cutoff frequencies are the only parameters of the filter learned from data. This solution still offers considerable flexibility, but forces the network to focus on high-level tunable parameters with broad impact on the shape and bandwidth of the resulting filter. Their tests and results show promising signs and they've proposed to carry out future work on this technique by testing it out on other corpora, including Voxceleb. This paper is aimed to being a small contribution to their research on this aspect.

3. DATASET DESCRIPTION



Dataset	VoxCeleb1	VoxCeleb2
# of POIs	1,251	6,112
# of male POIs	690	3,761
# of videos	22,496	150,480
# of hours	352	2,442
# of utterances	153,516	1,128,246
Avg # of videos per POI	18	25
Avg # of utterances per POI	116	185
Avg length of utterances (s)	8.2	7.8

Table 1: *Dataset statistics for both* VoxCeleb1 *and* VoxCeleb2*. Note* VoxCeleb2 *is more than 5 times larger than* VoxCeleb1*. POI: Person of Interest.*

Dataset	Dev	Test	Total
# of POIs	5,994	118	6,112
# of videos	145,569	4,911	150,480
# of utterances	1,092,009	36,237	1,128,246

Table 2: *Development and test set split.*

Fig. 1. Dataset Specifications[13]

The VoxCeleb2, is a large scale speaker recognition dataset obtained automatically fromopen-source media. VoxCeleb2 consists of over a million utterances from about 7363 speakers in total including the Voxceleb1 test and dev sets as well as Voxceleb 2 test and dev sets. Since the dataset is collected'in the wild', the speech segments are corrupted with real world noise including laughter, cross-talk, channel effects, music and other sounds. The dataset is also multilingual, with speech from speakers of 145 different nationalities, covering a wide range of accents, ages, ethnicities and languages. Audio segments present in the dataset are degraded with background chatter, laughter, overlapping speech and varying room acoustics [14]. Hence, it is essential for any speaker recognition technique designed for practical real-world applications to perform well under such conditions. The dataset was collected by a specifically designed computer vision pipeline, which included Listing candidate POI's, downloading POI-specific videos, Face Tracking, Face Verification, Active Speaker Verification, Duplicate Removal and Obtaining Nationality Labels.

4. ARCHITECTURE

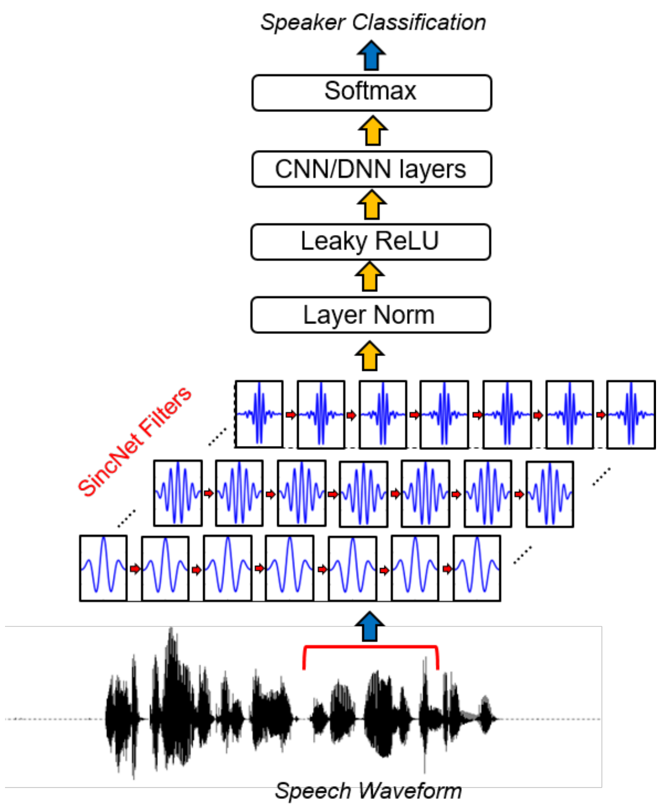


Fig. 2. SincNet Architecture[1]

The first layer of a standard CNN performs a set of time domain convolutions between the input waveform and some Finite Impulse Response (FIR) filters [15]. Each Convolution is defined as follows:

$$y[n] = x[n]h[n] = \sum_{l=0}^{L-1} x[l].h[n-l]$$

Here, x[n] is the signal chunk of speech, h[n] is the filter of length L, and y[n] is the filtered output. In standard CNNs, all the L elements (taps) of each filter are learned from data.Conversely, the proposed SincNet performs the convolution with a predefined function g that depends on few learnable parameters only, as highlighted in the following equation:

$$y[n] = x[n]g[n,]$$

This predefined function is a band pass filter in the frequency domain, which turns into parameterized sinc functions in the time domain.

$$g[n, f1, f2] = 2f2sinc(2f2n)2f1sinc(2f1n)$$

where, f1 and f2 are the learned low and high cutoff frequencies. The cut-off frequencies may be initialized randomly in the range of [0, fs/2], where fs represents the sampling frequency of the input signal.

This architecture also utilizes Hamming Windows which are particularly uselful to achieve high frequency selectivity. [16] . Also, the filters g are symmetric and thus do not introduce any phase distortions. Due to the symmetry, the filters can be computed efficiently by considering one side of the filter and inheriting the results for the other half, which are implemented in the SinCconvFast function in the repository. All operations involved in SincNet are fully differentiable and the cutoff frequencies of the filters can be jointly optimized with other CNN parameters using Stochastic Gradient Descent (SGD) or other gradient-based optimization routines. As shown in Fig. 2, a standard CNN pipeline (pooling,normalization, activations, dropout) can be employed after the first sinc-based convolution. Multiple standard convolutional, fully-connected or recurrent layers [37–40] can then be stacked together to finally perform a speaker classification with a softmax classifier.

5. IMPLEMENTATION

The architecture used is in exact coherence with what was proposed as the SincNet architecture, which has been evaluated on different corpora and compared to numerous speaker recognition baselines. The authors perform experiments on publicly availaible datasets such as TIMIT and LibriSpeech, to foster reproducible research. We extend this research by testing it on a much larger and noisier dataset called VoxCeleb2, specifications of which are mentioned in the previous sections.

5.1. Data Preprocessing

As stated, the Train dataset consists of the entire Voxceleb2 corpora in addition to the Voxceleb1 dev set. The Test dataset is the Voxceleb1 test set. The first step involved in the preprocessing of the data, was to convert the .m4a speech files of the Voxceleb2 dataset into .wav format, to be compatible with the libraries and architecture of SincNet, in which the TIMIT and LibriSpeech experiments were carried out. This step was particularly necessary because in order to make direct comparisons with these results, it's important to keep the core architecture the same, while making as little changes to the other dependencies in order to run these experiments. Further, some of the speakers were repeated in Voxceleb1 and Voxceleb2, which were first found out and then the duplicates were removed, in order to maintain the integrity of the training procedure. As for the .scp files that are required to be fed and the labels dictionary that must be generated, all code was written according to the specifications mentioned by the authors and the respective files were generated. Also, the .wav files of each speech sentence from each speaker was split into chunks of 200 ms (with 10 ms overlap), which were then fed into the SincNet architecture, which was as follows.

5.2. SincNet Setup

The primary layer performs sinc-based convolutions as portrayed in the past segments, utilizing 80 channels of length L = 251 samples. The architecture at that point utilizes two standard convolutional layers, both utilizing 60 channels of length 5. Layer normalization [17] was done for both the input samples and for all convolutional layers (including the SincNet input layer). Next, three fully-connected layers made out of 2048 neurons and standardized with batch standardization [18] were applied. Every hidden layer utilized Leaky ReLU [19] non-linearities. The parameters of the sinc-layer were introduced utilizing mel-scale cutoff frequencies, while the remainder of the system was instated with the notable "Glorot" introduction conspire [20]. Frame level speaker calssification was obtained by applying a softmax classifier, giving a lot of back probabilities over the focused on speakers. A sentence-level classification was just derived by averaging the frame predictions and voting for the speaker which augments the average posterior. Training utilized the RMSprop enhancer, with a learning rate lr = 0.001, = 0.95, = 107, and minibatches of size 128. All the hyper-parameters of the engineering were tuned on TIMIT, at that point inherited for Librispeech too.The speaker verification system was derived from the speaker-id neural network considering two possible setups. First, we consider the d-vector framework, which relies on the output of the last hidden layer and computes the cosine distance between test and the claimed speaker dvectors. As an alternative solution (denoted in the following as DNN-class), the speaker verification system can directly take the softmax posterior score corresponding to the claimed identity. The two approaches will be compared in Sec. 5. Ten utterances from impostors were randomly selected for each sentence coming from a genuine speaker. Note that to assess our approach on a standard open-set speaker-id task, all the impostors were taken from a speaker pool different from that used for training the speaker-id DNN.

5.3. Configuration

Sampling Frequency = 16000,
Frame shift = 10 ms,
window frame length = 200 ms,
softmax classification layer = 7363,
dropout = 0,
batch-size = 128,
No. of Epochs = 400,
No. of Batches = 1500,
CNN filter length = 251, 5, 5.

6. RESULTS OBTAINED

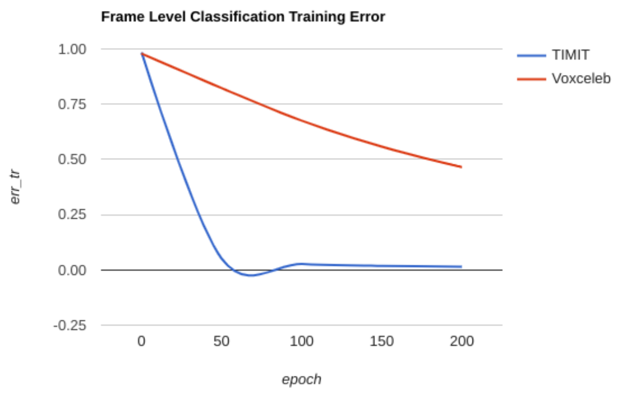


Fig. 3. Training error Comparison

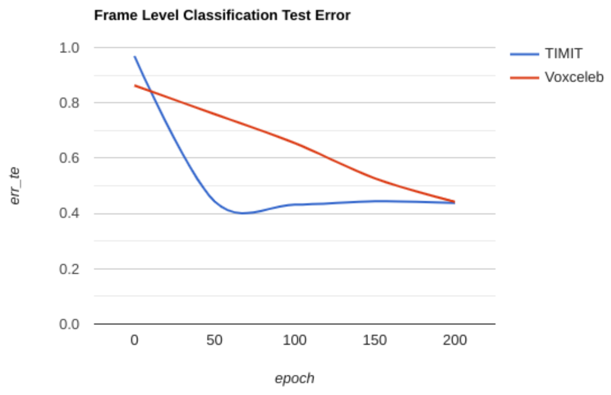


Fig. 4. Test error Comparison

Epoch	TIMIT	Voxceleb
0	0.91	0.85
40	0.023	0.70
100	0.012	0.46
180	0.009	0.12

Table 1. Sentence-level Classification error of Test data.

6.1. Analysis

As seen by the experiment on the Voxceleb corpora, we have results to about 183 epochs after over two weeks of training. The results that were obtained yet, are mentioned in the table above, and a head to head comparison to the TIMIT results can be made. Firstly, as in the case of the TIMIT dataset, which converges pretty fast after about 30 epochs, the voxceleb data doesn't converge even after over 180 epochs. This indicates the inefficiency of the architecture, wherein the TDNN-based voxceleb recipe converges after about 6 epochs and 1 week of training, obtaining close to optimum results. The train and test errors do seem to be decreasing at a steady rate, which eventually might lead to the expected results, but it surely requires much more compute power and time, to achieve those results. On the other hand, the sentence-level classification error rate shows a significantly faster decrease, thus indicating that the batchwise error rate for each sentence in the batch, of which the Sinc features were extracted, do work quite well to an extent, and might even reduce more with further epochs. Finally, we can't really comment yet, about the effectiveness of Sinc-based features over MFCC or F-Bank features, but one thing can be said that this architecture is not particularly suitable for larger datasets, given limited time and computing ability.

7. CONCLUSION

To achieve comparable results, we followed the exact same steps as followed by [1] in their recipe for training Voxceleb. Although we couldn't train it in its entirety until the sincnet converges, and then make a head to head comarisons, our training of the dataset upto 183 eposchs reveals some key insights. First, although the authors claim that the SincNets usually have a fast convergence, and better performance as the training continues, the Voxceleb dataset didn't converge even after about two weeks of training, which is pretty slower as compared to the x-vector TDNN based recipe on VoxCeleb2, which converges after about a week's training and 6 epochs. On the other hand, although we cant directly comment on the effectiveness of the Sinc-based features on the Voxceleb data as of yet, we do tend to notice that the error rate has a steady decrease in the error rates, indicating that it might require quite a bit more time to converge and ofcourse more computing power. We can do comment that this is probably not the ideal architecture to implement these sinc-based features with the voxceleb dataset, especially because of the sheer size of the dataset. Another thing quite interesting is that since the voxceleb is quite a bit noisy as compared to TIMIT, this might be one of the reasons why the Voxceleb dataset hasn't converged yet, and further tests on this aspect might help discover this aspect of the experiment a bit better.

8. FUTURE WORK

To achieve comparable results and ensure the effictiveness of the Sinc-ased features, we can follow the exact same steps as followed by [21] in their recipe for training voxceleb2. Hence, the only thing that we could possibly modify is the input waveforms using sincnet based filters, in place of the MFCC features, which are fed to the xvector TDNN-based model for training. All other steps must remain same, to ensure that whatever change in results is obtained, is because of this one single change in the recipe that was made.

In the recipe for VoxCeleb v2, the .wav files from Voxceleb2 are used for taining the xvector TDNN-based model and the .wav files from Voxceleb1 are used as test files for speaker recognition purposes. The approach would be to follow the recipe as it creates the train and test datasets, and instead of making the mfcc's, we would pass the wav.files through the Sincnet scripts to extract band-pass filtered features. Then, we follow the recipe as it augments the data adding reverberations and combining it with noisy files such as the musan corpora. Then, we move on to do the x-vector training, which is essentially a DNN model as decribed in this paper [21].LDA and PLDA based transformations are also employed to reduce the dimensionality of the feature set.[22] Finally, the obtained results in the last stage, essentially the EER remains to be checked with the current results obtained without using SincNet filters.

9. REFERENCES

[1] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[2] Homayoon Beigi, *Speaker recognition*, Springer, 2011.

[3] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[4] Pavel Matějka, Ondřej Glembek, Fabio Castaldo, Md Jahangir Alam, Oldřich Plchot, Patrick Kenny, Lukáš Burget, and Jan Černocky, "Full-covariance ubm and heavy-tailed plda in i-vector speaker verification," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 4828–4831.

[5] Sandro Cumani, Oldřich Plchot, and Pietro Laface, "Probabilistic linear discriminant analysis of i-vector posterior distributions," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7644–7648.

[6] John HL Hansen and Taufiq Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[7] David Chen, Sam Tsai, Vijay Chandrasekhar, Gabriel Takacs, Huizhong Chen, Ramakrishna Vedantham, Radek Grzeszczuk, and Bernd Girod, "Residual enhanced visual vectors for on-device image matching," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, 2011, pp. 850–854.

[8] Sree Harsha Yella, Andreas Stolcke, and Malcolm Slaney, "Artificial neural network features for speaker diarization," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 402–406.

[9] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[10] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[11] Fred Richardson, Douglas Reynolds, and Najim Dehak, "A unified deep neural network for speaker and language recognition," *arXiv preprint arXiv:1504.00923*, 2015.

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

[13] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[14] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[15] Lawrence R Rabiner and Ronald W Schafer, *Theory and applications of digital speech processing*, vol. 64, Pearson Upper Saddle River, NJ, 2011.

[16] Gordana Jovanovic-Dolecek and Sanjit K Mitra, "A new two-stage sharpened comb decimator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 7, pp. 1414–1420, 2005.

[17] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E Hinton, "Layer normalization. corr abs/1607.06450," *Bulletin of Mathematical Linguistics (Proc. EAMT)(108)*, pp. 49–60, 2016.

[18] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[19] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, 2013, vol. 30, p. 3.

[20] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[21] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[22] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.