Learning Filterbanks from Raw Waveform for Accent Classification

Rashmi Kethireddy
Speech Processing Laboratory
IIIT-Hyderabad, India
rashmi.kethireddy@research.iiit.ac.in

Sudarsana Reddy Kadiri
Department of Signal Processing and Acoustics
Aalto University, Finland
sudarsana.kadiri@aalto.fi

Suryakanth V. Gangashetty
Speech Processing Laboratory
IIIT-Hyderabad, India
svg@iiit.ac.in

***Abstract*—Most of the applications in speech use mel-frequency spectral coefficients (MFSC) as features as they match the human perceptual mechanism, where the emphasis is given to vocal tract characteristics. But in accent classification, mel-scale distribution of filters may not always be the best representations, e.g., pitch accented languages where the emphasis should be on vocal source information too. Motivated by this, we use end-to-end classification of accents directly from waveforms which will reduce the effort of designing features specific to each corpus. The convolution neural network (CNN) model architecture is designed in such a way that the initial layers exhibit similar operation as in MFSC by initializing the weights using time approximate of MFSC. The entire network along with initial layers is trained to learn accent classification. We observed that learning directly from waveform improved the performance of accent classification when compared to CNN trained on hand-engineered features by 10.94% UAR on the test dataset of common voice corpus. Analyzing the filters after learning, we observed changes in distribution and bandwidths of center frequencies. We further observed the importance of appropriately initializing CNN filters.**

***Index Terms*—Accent classification, convolution neural network, raw waveform, first order scattering transform.**

I. Introduction

The speech of speakers belonging to a particular region exhibit some similar patterns in pronunciations that distinguish from other areas. These dissimilarities in the patterns of pronunciations due to geographic spread or socio-regional or influence of first language of the speaker are termed as accents/dialects of speech. Dialect is the super-class of accents which further includes vocabulary and grammatical variations. This paper focuses on accent, i.e., pronunciation variations. Accent specific automatic speech recognition (ASR) [1] or deep accent embedding derived from a network trained to classify accent can improve the performance of the ASR system [2].

Previous studies on automatic classification of accent/dialect is divided into three areas: the first one is to find the appropriate frame-level feature extraction which contains accent components, second representing variable-length features in compact and fixed representations, and third is to design better classifiers. Accent can be varied in speech either in acoustics (distribution of sounds, stress, rhythm, and intonation patterns) [3]–[6] or phonotactics (sequence of sounds) [7]–[10] of the speech.

Representing the spectral features in unsupervised and compact form is the most popular area of research, where interesting approaches such as i-vectors [3], [4], [11], [12], unsupervised bottleneck features (uBNF) [13], [14], autoencoders with recurrent neural networks [15] and factorized hierarchical variational autoencoder (FHVAE) are explored. The most widely used classifiers are support vector machine (SVM), linear discriminant analysis (LDA) and its variants such as quadratic discriminative analysis (QDA), probabilistic linear discriminant analysis (PLDA), and heteroscedastic linear discriminant analysis (HLDA) [12], [16]–[18].

With the invent of convolution neural networks (CNN) in dialect classification [19], [20] which can handle variable length utterances along with classification, three stages reduced to two. In [19], CNNs are evaluated over the Arabic database (MGB-3) with various acoustic features such as - mel-frequency cepstral coefficients (MFCC), log mel-scale filterbank energies (FBANK), and spectrogram. This system outperformed all the other baselines till then. Motivated by this we considered this system as our baseline system.

There has been a tremendous improvement in the field of vision [21], [22] by directly learning from pixels. Currently, some applications of speech explored learning directly from raw waveform such as speech recognition [23]–[26], speaker verification [27], emotion recognition [28], and environment sound recognition [29]. In [30], raw waveform modeling approaches are used in Styrian dialect identification which performed better than the baseline methods. Inspired by this, we focus on analyzing the CNN filters trained on raw waveform for accent classification. Using manually designed filterbanks arranged non-linearly (mel-scale) might perform well for most accents of English, but a few of them are pitch accented (Hong Kong English, South African English, and Welsh), where the emphasis should be given to voice (glottal) sources than vocal tract components of speech. Therefore, learning directly from waveform compensates manual feature engineering based on the characteristics of the corpus.

In this study, we used an end-to-end convolution neural network (CNN) with the first three layers acting as a replacement to log mel filterbanks (FBANK). First CNN filter weights are initialized to the time domain approximated spectral filters derived using first-order deep scattering spectrum [31]. The approach used to initialize the weights is similar to that

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

described in [25], [32]. Then the complete network is trained along with first layers to classify accents. We conducted experiments with the common voice database [33]. Major contributions of this study are as follows:

- To the best of our knowledge, this is the first study to look into CNN filters while learning directly from raw waveform for accent classification.
- Compared the rate of convergence when filters initialized to time-approximate MFSC and random initialization.
- Analyzed the importance of dynamic filterbanks when compared to fixed filter banks.
- Performance evaluations for CNN filters when initialized to approximates of linearly placed filterbanks.

The organization of this paper is as follows: Section II presents the architecture of the raw waveform CNN network. Section III describes the corpus and the baseline system. The configuration and the tools used in neural network architecture is described in Section IV. Section V presents the results with analysis, followed by a summary in Section VI.

II. Raw waveform CNN Architecture

This section describes the complete architecture of the neural network used in this study. The left side of the network in Figure 1 represents the network which is approximate of computing log mel filterbank energies which we term as trainable filterbank network (TFN), while the right side represents the other layers of the network for classifying the accents. For classifying the network designed using two 1-dimensional CNN layers followed by three fully connected (FC) layers. Each layer is defined by aF-bK-cS, where a: number of filters, b: size of the kernel, and c: stride along the time axis.

A. MFSC approximated CNN layers

This section gives the overview of initial layers of CNN which acts as learnable replacement to mel-filter spectral coefficients(MFSC). We refereed them as the trainable filterbank network (TFN) in this study.

1) Trainable filterbank network (TFN) definition: The implementation of the initial layers in TFN is based on studies reported in [32]. This architecture of this network consists of three 1-dimensional CNN layers, one L2 pooling layer, one instance normalization layer and two operations (absolute and log operations). These layers are structured such that after an appropriate weight initialization, these acts as a replacement to MFSC. The structural details of each layer are given below:

(a) **First layer**: CNN filter operation in this layer is equivalent to pre-emphasis operation.

$$y^t = \mathbf{W}\mathbf{X}^t,$$

where $\mathbf{W}$ is the weight row vector of convolution filter (size= 1x2), $\mathbf{X}_i^t$ represents the input column vector $[x_{t-1}\ , x_t]$ (which represents the previous and present input along the sequence), and y^t represents the output at time t along time sequence.

(b) **Second layer**: This is a complex convolution layer with a filter size 200, and 80 such filters (40 representing real and other 40 representing imaginary part) which are initialized to complex Gabor wavelets approximating to MFSC.

(c) L2 pooling is performed which is the approximation of modulus operation, which computes the magnitude of the output from real and imaginary filters, and reduces 80 filters to 40 filters.

(d) **Third layer**: This CNN filter acts as the square of the Hanning window with a width of 25 ms (which is equivalent to 200 samples along the time axis for 8 kHz as sampling frequency) and a stride variant of 80. There are 40 such filters in this layer.

(e) Since the weights are not constrained to have positive values, a log compression over the one added to the absolute of the output of the previous layer is computed.

(f) Then finally an instance normalization is applied over the log compressed output to stabilize training, which is similar to mean-variance normalization.

Biases in all these three convolution layers are set to zero, to have a similar structure as in the first order scattering transform.

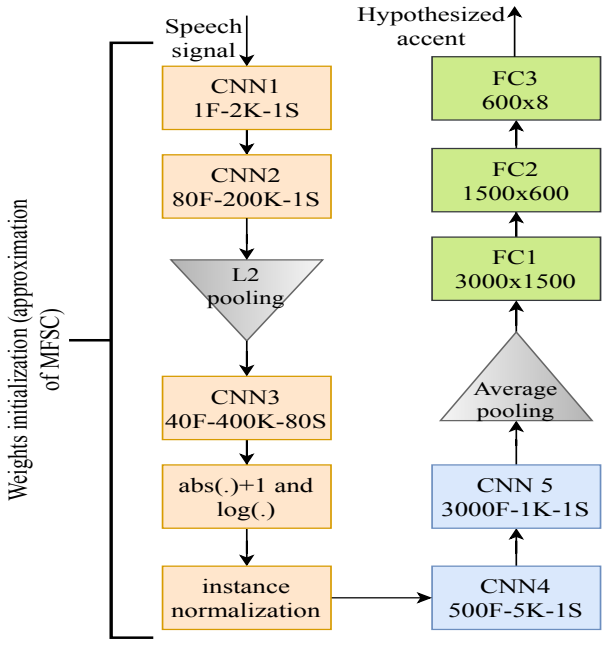

Fig. 1. Raw waveform CNN network architecture for accent classification.

2) Weight initialization: The details of the weight initialization of CNN filters in three layers of TFN are given below:

(i) **First layer**: The weights of CNN filters 'W' in the first layer are initialized to $[-0.97, 1]$ which makes this layer computation similar to the pre-emphasize operation of the speech signal.

(ii) **Second layer**: We evaluated three variants of weight initializations in the second layer of TFN. They are weight initialization approximating mel-scaled filterbank

spectral coefficients, approximating linear-scaled filterbank spectral coefficients, and random initialization.
For the first two variants, time domain approximates of filters are computed based on the first order scattering transform [31] which is give by:

$$Mx(t,n) \approx |x * \psi_n|^2 * |\phi|^2(t) \quad (1)$$

where $Mx(t,n)$ represents triangular filterbanks at center frequency η_n with full width at half maximum (FWHM) w_n. n takes values from 1 to N, where N defines the number of filters, and $\phi(t)$ defines the Hanning window. In equation (1), ψ_n defines the wavelet which is an approximation of n^{th} triangular filter place at center frequency η_n. Similar to [25], we use Gabor wavelets for ψ_n whose equation is given by:

$$\psi_n(t) \; \alpha \; e^{-2\pi i \eta_n t} \frac{1}{\sqrt{2\pi}\sigma_n} e^{\frac{-t^2}{2\sigma_n^2}}, \quad (2)$$

where σ_n is computed from w_n, $\sigma_n = \frac{2\sqrt{2\log 2}}{w_n}$. Each wavelet is normalized to have same energy as triangular filter and the time support of ψ_n is constrained to be less than the window size of ϕ.
In the first variant, the filters are initialized to the approximate of MFSC. So, the center frequencies 'η' of triangular filters should be spaced based on mel-scale and the variance σ_n of each filter is computed from FWHM of triangular filter. Second variant of initialization is approximating time domain filters to linearly spaced filters in frequency domain. Therefore, weights are computed using right part of the equation (2) by setting the center frequencies 'η' of Gabor wavelets to a linear scale. Third variant is random initialization of weights in CNN filters.

(iii) **Third layer**: Weight initialization of CNN filters in third layer is computed by $|\phi|^2(t)$ from equation (1) which is the square of Hanning window.

B. Other layers of the network

The architecture of CNN model used in this study has two variants: first one is the TFN model which is the replacement of learnable MFSC's and the second variant of the network is learned to distinguish accents. For ease of understanding, we termed this second variant as classifier network. The details of network architecture of second variant are provided below.

The classifier network has two 1-dimensional CNN layers (500F-5K-1S and 3000F-1K-1S) which covers a span of 5 frames with a stride of 1. Global averaging is done to get a fixed-length output of size 3000 passed through three fully connected (FC) layers with output 1500, 600, and 8. Finally, the log softmax layer gives a normalized estimated probability. Rectified linear unit (ReLU) activation [34] with a dropout [35] of 0.51 is used across the network. The network is trained with negative log-likelihood (NLL) loss function and stochastic gradient descent (SGD) optimizer. Since the corpus which we considered is highly imbalanced, the loss function is class balanced which is computed based on proposition 1 in [36].

III. Corpus and Baseline System

This section describes the corpus used in this study for evaluations and an overview of the baseline system.

A. Common voice

Common voice corpus (version 1) is collected using crowdsourcing from the people across the world [33]. Along with speech clips of the user, some metadata of the speaker is collected which includes accent. It is the read speech with a sampling frequency of 48 kHz. The data collected has 16 accents in English, namely: United States English (US), Australian English (AU), England English (EN), Canadian English (CA), Filipino English (FP), Hong kong English (HK), India and South Asia (IN), Irish English (IR), Malaysian English (ML), New Zealand English (NZ), Scottish English (SE), Singaporean English (SG), South Atlantic (Falkland Islands, Saint Helena) English (SA), Southern African (South Africa, Zimbabwe, Namibia) English (SAF), Welsh English (WE), and West Indies and Bermuda (Bahamas, Bermuda, Jamaica, Trinidad) English (WI). Considering only the speech utterances for which accent is provided and the utterances with no downvotes, resulted in a subset of the dataset with imbalanced distribution of samples across accents. So, we considered only top frequently occurred 8 accents (SA, AU, CA, EN, IN, NZ, SE, and US) which resulted in 57356 utterances in train set, 1200 utterances in validation (val.) set, and 1175 utterances in test set. The average length of utterance is 4.12 seconds. The distribution of data (in %) is shown in Table I.

TABLE I
COMMON VOICE DATA DISTRIBUTION (IN %) WITH A TOTAL OF 57356 UTTERANCES IN TRAIN, 1200 IN VALIDATION (VAL.), AND 1175 IN TEST SETS.

	SA	**AU**	**CA**	**EN**	**IN**	**NZ**	**SE**	**US**
train	1.7	6.7	6.3	24.2	6.9	1.8	2.5	49.4
val.	1.6	7.9	5.7	25.1	6.8	1.6	2.6	48.2
test	1.9	7.3	8.2	23.6	6.8	1.2	2.0	48.5

B. Baseline system

Convolutional neural networks (CNN) are widely used deep network architecture [21], [37], due to their automatic detection of important features and it's architecture is mainly motivated by the observations in [38], on visual cortex of a cat.

From the literature for accent classification in [19], it can be observed that CNN model trained with log mel filterbank energies (FBANK) outperformed all the state-of-the-art till then. So, our baseline system is an end-to-end CNN model which is based on [19] except for few layers excluded from the network architecture. The architecture of the network is as in the right side of Figure 1. Instead of TFN as in the left side of Figure 1, hand-engineered features, such as FBANK, spectrogram, and MFCC features are given as input to CNN.

In both the baseline and proposed systems, the architecture of this network is the same with the input of size 40 except for spectrogram whose input size is 200. The input of the baseline is hand-engineered features while the input of the proposed system is from the TFN (initialized to approximate of MFSC).

IV. Experimental setup

All the speech samples are down sampled to 8 kHz and the utmost 4 seconds of each utterance is considered to compensate the time and memory complexities while working with waveform directly. From our initial experiments on the baseline systems, we found that after 170 epochs the weight and the accuracy on validation data are toggling. So, we set number of epochs to 170 on all the variants of models and with a learning rate of 0.05 to have a fair comparison. For evaluations, unweighted average recall (UAR) is considered as our primary metric as it is unbiased to imbalanced classes while accuracy is considered as a secondary metric.

In this study, we experimented with five variants of TFN configurations. Out of them, first four are similar to the configurations in [25], [39] while the last one is introduced in this study. (1) LearnFBANK: learns only the second convolution layer of TFN (2) Fixed: Fixes the TFN during training (3) Learnall: learns all the layers in TFN architecture (4) RandInt: Randomly initialized the second convolution layer of TFN and learns only that layer of TFN, and (5) LinearInt: Initialized convolution layer of TFN to time approximate of linearly scaled filter coefficients and learns only that layer of TFN. There is no restriction of on classifier network, all the layers of classifier network are learned during training.[1]

V. Results and Discussion

This section gives an analysis of CNN filters after learning and it also presents and analyzes the evaluation results of baseline and proposed systems for accent classification.

A. Analysis of CNN filters

Figure 2 shows the heat-map of the magnitude of frequency responses for the filters at variant stages and configurations. Figure 2(a) represents the non-linearly initialized filters, and Figure 2(b) shows the frequency responses of the learned filters which are initialized to time domain filters that are approximate of mel-scaled filterbanks. Figure 2(c) and 2(d) shows the responses of learned filters initialized to time domain filterbanks which are approximates of linear-scaled filters in frequency domain and random weights, respectively. For phone recognition, there is a lot of variability in bandwidth but there is no distortion in center frequencies observed in [25] . However, for accent classification the distribution of filters after learning is distorted where filters 1 to 25 are arranged below 1 kHz giving much more emphasis to the low frequency components and other filters (25 to 40) are arranged with a very steep linear scale (in Figure 2 (b)),

[1]The trained models along with the split of exact data set used is provided at : https://github.com/r39ashmi/cvaccent_wav.

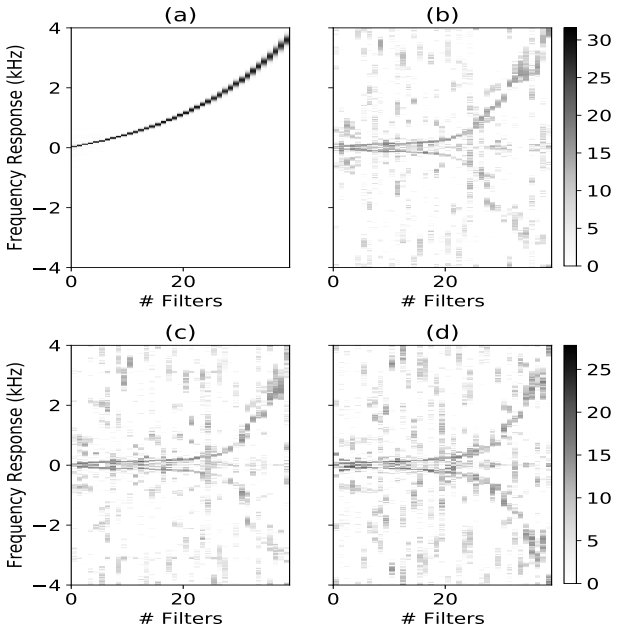

Fig. 2. Magnitude of frequency response of 40 filters (modulus of complex filters) ordered by their peak amplitudes. (a) initialized to time approximated MFSC, (b) learned filters which were initialized to time approximated MFSC initialization, (c) learned filters which were initialized to time approximate of linearly places filterbanks, and (d) learned filters which were initialized randomly.

which supported our initial hypothesis. Even though filters are analytically initialized, a moderate amount of energy is leaked into negative frequencies after learning to result in symmetry at low frequencies below 1 kHz. The heat-maps of filters in Figure 2 (c) and (d) represents filters learnt from randomly initialized and time approximates of linearly placed filters. We observed that all the variants of initialization showed a similar distribution after learning, however training took more epochs for randomly initialized weights and time approximates of linear scaled filters.

B. Analysis of results

Table II shows the performance evaluation of the baseline system (FBANK, spectrogram, and MFCC with classifier network) and the proposed system (raw waveform+TFN+classifier network). First three rows of Table II show the results of the baseline system, while others show the results of the proposed system with five different configurations. For discussion of results, here we considered only UAR which is our primary metric. It can be observed that among the three baselines features, classifier network trained on MFCC's performed better. The performance of all the proposed systems (training directly from waveform) are better than baseline systems on both validation and test datasets. Now analyzing the results of proposed systems among them, "LearnFBANK" configuration outperformed both on validation and test datasets. By comparing best systems from both baseline (MFCC) and proposed (LearnFBANK), we found an improvement of 10.94% UAR on test dataset.

TABLE II
PERFORMANCE EVALUATION (IN UAR [%] AND ACCURACY [%]) OF BASELINE SYSTEM WITH INPUT AS ACOUSTIC FEATURES AND PROPOSED SYSTEM WITH INPUT AS RAW WAVEFORM FOR ACCENT CLASSIFICATION.

system configuration	val.		Test	
	UAR	ACC.	UAR	ACC.
Input features	**Baseline systems results**			
FBANK	66.46	79.31	67.08	76.23
spectrogram	62.14	74.64	58.98	71.55
MFCC	70.17	76.91	68.97	77.31
Input raw waveform	**Proposed systems results**			
LearnFBANK	**72.83**	**81.15**	**76.52**	**81.26**
Fixed	72.46	80.31	71.50	79.30
Learnall	71.36	80.48	**76.83**	81.09
RandInt	70.79	80.06	70.09	77.34
LinearInt	72.07	79.48	74.62	78.87

Further, we also introduced "LinearInt" configuration in this study and found comparable performance with the highest performer of the proposed system, and better than all the baseline systems considered. This improvement in performances, when learnt directly from waveform supports our hypothesis that dynamic learning of filterbanks improves the performance of accent classification.

During the experiments, we observed that the initial 15 iterations of training with "Fixed" configuration exhibited better performance, later on, the configurations "LearnFBANK" and "Learnall" outperformed all the other configs which showed the importance of dynamic filterbanks and appropriate initialization.

From figure 3, it can be observed that the network with random initialized weights (green color) converged slowly when compared to MFSC approximated time domain (TD) filters (red color), which highlights the importance of initialization of CNN filters to MFSC approximated filters. Other than the configurations from [25], we also introduced "LinearInt" configuration which also gave a comparable performance and better than "RandInt" and "Fixed" configurations.

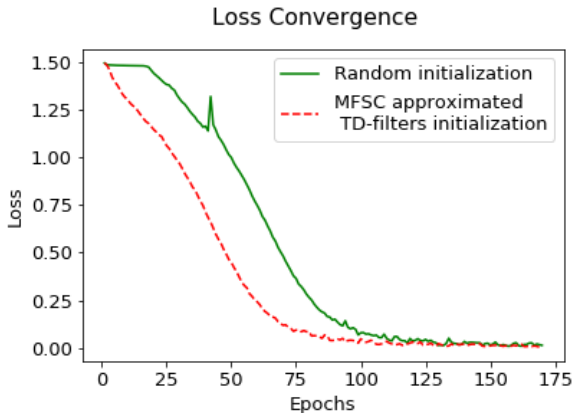

Fig. 3. Loss convergence of random initialization vs initialization to MFSC approximated filters.

VI. SUMMARY AND CONCLUSION

Our initial hypothesis was that hand engineering features for accent classification may need domain knowledge and may not perform up to the mark. So, this study mainly explored learning filterbanks which are initialized based on hand-designed features which are embedded as part of CNN network for accent classification. The learned filters changed in their distribution and bandwidths. We observed that these learned filters improved the performance when compared to fixed filters and hand-engineered features which supported our hypothesis. We also found that the efficient initialization of filterbanks will converge the network faster.

VII. ACKNOWLEDGEMENTS

The first author would like to thank the University Grants Commission, India (award No. 3582/(NET-NOV2017)) for supporting her as a Ph.D. scholar. The second author would like to thank the Academy of Finland (project no. 312490) for supporting his stay in Finland as a Postdoctoral Researcher.

REFERENCES

[1] F. Biadsy, "Automatic dialect and accent recognition and its application to speech recognition," Ph.D. dissertation, Columbia University, 2011.
[2] A. Jain, M. Upreti, and P. Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning." in *Proc. Interspeech*, 2018, pp. 2454–2458.
[3] H. Behravan, V. Hautamäki, S. M. Siniscalchi, T. Kinnunen, and C. Lee, "i-vector modeling of speech attributes for automatic foreign accent recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 29–41, 2016.
[4] A. Hanani, M. J. Russell, and M. J. Carey, "Human and computer recognition of regional accents and ethnic groups from British English speech," *Computer Speech & Language*, vol. 27, no. 1, pp. 59–74, 2013.
[5] J. H. L. Hansen, U. H. Yapanel, R. Huang, and A. Ikeno, "Dialect analysis and modeling for automatic classification," in *Proc. Interspeech*, 2004, pp. 1569–1572.
[6] F. Biadsy and J. Hirschberg, "Using prosody and phonotactics in Arabic dialect identification," in *Proc. Interspeech*, 2009, pp. 208–211.
[7] M. Najafian, S. Safavi, P. Weber, and M. J. Russell, "Identification of British English regional accents using fusion of i-vector and multi-accent phonotactic systems," in *Proc. ODYSSEY*, 2016, pp. 132–139.
[8] F. Biadsy, J. Hirschberg, and N. Habash, "Spoken arabic dialect identification using phonotactic modeling," in *Proc. Workshop on Computational Approaches to Semitic Languages*, 2009, pp. 53–61.
[9] M. A. Zissman, T. P. Gleason, D. Rekart, and B. L. Losiewicz, "Automatic dialect identification of extemporaneous conversational, Latin American Spanish speech," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 1996, pp. 777–786.
[10] F. S. Richardson, W. M. Campbell, and P. A. Torres-Carrasquillo, "Discriminative n-gram selection for dialect recognition," in *Proc. Interspeech*, 2009, pp. 192–195.
[11] H. Behravan, V. Hautamäki, and T. Kinnunen, "Foreign accent detection from spoken Finnish using i-vectors," in *Proc. Interspeech*, 2013, pp. 79–83.
[12] A. DeMarco and S. J. Cox, "Iterative classification of regional British accents in i-vector space," in *Proc. Symposium on Machine Learning in Speech and Language Processing*, 2012, pp. 1–4.
[13] Q. Zhang and J. H. Hansen, "Dialect recognition based on unsupervised bottleneck features." in *Proc. Interspeech*, 2017, pp. 2576–2580.
[14] Q. Zhang and J. H. L. Hansen, "Language/dialect recognition based on unsupervised deep learning," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 873–882, May 2018.
[15] B. W. Schuller, A. Batliner, C. Bergler, F. B. Pokorny, J. Krajewski, M. Cychosz, R. Vollmann, S.-D. Roelen, S. Schnieder, E. Bergelson, A. Cristià, A. Seidl, A. Warlaumont, L. Yankowitz, E. Nöth, S. Amiriparian, S. Hantke, and M. Schmitt, "The interspeech 2019 computational paralinguistics challenge: Styrian dialects, continuous sleepiness, baby sounds and orca activity," in *Proc. Interspeech*, 2019, pp. 2378–2382.

[16] K. Kumpf and R. W. King, "Foreign speaker accent classification using phoneme-dependent accent discrimination models and comparisons with human perception benchmarks," in *Proc. Eurospeech*, 1997, pp. 2323–2326.

[17] P. Angkititrakul and J. H. L. Hansen, "Advances in phone-based modeling for automatic accent classification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 634–646, 2006.

[18] H. Behravan, V. Hautamäki, and T. Kinnunen, "Factors affecting i-vector based foreign accent recognition: A case study in spoken Finnish," *Speech Communication*, vol. 66, no. C, pp. 118–129, 2015.

[19] S. Shon, A. Ali, and J. Glass, "Convolutional neural network and language embeddings for end-to-end dialect recognition," in *Proc. ODYSSEY*, 2018, pp. 98–104.

[20] R. Ubale, Y. Qian, and K. Evanini, "Exploring end-to-end attention-based neural networks for native language identification," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2018, pp. 84–91.

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

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Proc. Interspeech*, 2014, pp. 890–894.

[24] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2015, pp. 4624–4628.

[25] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve, and E. Dupoux, "Learning filterbanks from raw speech for phone recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2018, pp. 5509–5513.

[26] T. Sainath, R. J. Weiss, K. Wilson, A. W. Senior, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Proc. Interspeech*, 2015, pp. 1–5.

[27] H. Muckenhirn, M. M. Doss, and S. Marcell, "Towards directly modeling raw speech signal for speaker verification using CNNs," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2018, pp. 4884–4888.

[28] S. Latif, R. Rana, S. Khalifa, R. Jurdak, and J. Epps, "Direct modelling of speech emotion from raw speech," *CoRR*, vol. abs/1904.03833, 2019.

[29] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2017, pp. 421–425.

[30] S. P. Dubagunta and M. Magimai-Doss, "Using speech production knowledge for raw waveform modelling based styrian dialect identification," in *Proc. Interspeech*, 2019, pp. 2383–2387.

[31] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Trans. Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[32] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, "End-to-end speech recognition from the raw waveform," in *Proc. Interspeech*, 2018, pp. 781–785.

[33] Mozilla, "Project Common Voice,[online]," Available: https://voice.mozilla.org/en/data, 2017.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[36] Y. Cui, M. Jia, T. Lin, Y. Song, and S. J. Belongie, "Class-balanced loss based on effective number of samples," *CoRR*, vol. abs/1901.05555, 2019. [Online]. Available: http://arxiv.org/abs/1901.05555

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

[38] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[39] https://github.com/facebookresearch/tdfbanks.git, 2018.