Kethireddy, Rashmi; Kadiri, Sudarsana Reddy; Gangashetty, Suryakanth V.

Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations

Published in:
The Journal of the Acoustical Society of America

DOI:
10.1121/10.0009405

Published: 01/02/2022

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Kethireddy, R., Kadiri, S. R., & Gangashetty, S. V. (2022). Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations. *The Journal of the Acoustical Society of America*, *151*(2), 1077-1092. https://doi.org/10.1121/10.0009405

Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations

Rashmi Kethireddy, Sudarsana Reddy Kadiri and Suryakanth V. Gangashetty

Citation: The Journal of the Acoustical Society of America **151**, 1077 (2022); doi: 10.1121/10.0009405
View online: https://doi.org/10.1121/10.0009405
View Table of Contents: https://asa.scitation.org/toc/jas/151/2
Published by the Acoustical Society of America

JASA ARTICLE

Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations

Rashmi Kethireddy,[1,a)] Sudarsana Reddy Kadiri,[2,b)] and Suryakanth V. Gangashetty[3,c)]
[1]*Speech Processing Laboratory, International Institute of Information Technology-Hyderabad (IIIT-H), 500032, India*
[2]*Department of Signal Processing and Acoustics, Aalto University, Otakaari 3, FI-00076 Espoo, Finland*
[3]*Koneru Lakshmaiah Education Foundation, Vaddeswaram, 522502, Andhra Pradesh, India*

ABSTRACT:
The goal of this study is to investigate advanced signal processing approaches [single frequency filtering (SFF) and zero-time windowing (ZTW)] with modern deep neural networks (DNNs) [convolution neural networks (CNNs), temporal convolution neural networks (TCN), time-delay neural network (TDNN), and emphasized channel attention, propagation and aggregation in TDNN (ECAPA–TDNN)] for dialect classification of major dialects of English. Previous studies indicated that SFF and ZTW methods provide higher spectro-temporal resolution. To capture the intrinsic variations in articulations among dialects, four feature representations [spectrogram (SPEC), cepstral coefficients, mel filter-bank energies, and mel–frequency cepstral coefficients (MFCCs)] are derived from SFF and ZTW methods. Experiments with and without data augmentation using CNN classifiers revealed that the proposed features performed better than baseline short-time Fourier transform (STFT)–based features on the UT-Podcast database [Hansen, J. H., and Liu, G. (**2016**). "Unsupervised accent classification for deep data fusion of accent and language information," Speech Commun. **78**, 19–33]. Even without data augmentation, all the proposed features showed an approximate improvement of 15%–20% (relative) over best baseline (SPEC–STFT) feature. TCN, TDNN, and ECAPA-TDNN classifiers that capture wider temporal context further improved the performance for many of the proposed and baseline features. Among all the baseline and proposed features, the best performance is achieved with single frequency filtered cepstral coefficients for TCN (81.30%), TDNN (81.53%), and ECAPA-TDNN (85.48%). An investigation of data-driven filters, instead of fixed mel-scale, improved the performance by 2.8% and 1.4% (relatively) for SPEC–STFT and SPEC–SFF, and nearly equal for SPEC–ZTW. To assist related work, we have made the code available ([Kethireddy, R., and Kadiri, S. R. (**2022**). "Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations," https://github.com/r39ashmi/e2e_dialect (Last viewed 21 December 2021)].). © *2022 Acoustical Society of America*. https://doi.org/10.1121/10.0009405

(Received 7 July 2021; revised 16 January 2022; accepted 16 January 2022; published online 16 February 2022)
[Editor: John H. L. Hansen] Pages: 1077–1092

I. INTRODUCTION

Identifying the regional origin of a speaker from the acoustic characteristics of speech is known as dialect identification. The task of dialect identification is usually considered to be a sub-class of language identification; however, dialect discrimination is a bit more challenging than language identification due to low variability among dialects.

Dialect information in speech is reflected in both acoustic and linguistic variations. Studies by Hansen and Liu (2016) have shown that acoustic variations are more prominent than the linguistic variations [acoustic models performed better than linguistic models by 15.8% absolute unweighted average recall (UAR)] for major dialects of English. The acoustic variations among dialects include segmental and supra-segmental features, and they can be extracted directly from the speech signal (Behravan *et al.*, 2016; Bougrine *et al.*, 2018; DeMarco and Cox, 2012; Rajpal *et al.*, 2016; Rouas, 2007) or they can be modelled indirectly from the phonetic information derived from the speech signal (Chen *et al.*, 2011; Chen *et al.*, 2014; Najafian *et al.*, 2018; Shon *et al.*, 2018a).

Hand-engineered segmental feature representations obtained from short-time Fourier transform (STFT) spectrum [such as spectrogram, mel filter-bank energies (MFBE)/mel spectrogram and mel–frequency cepstral coefficients (MFCCs)] are widely investigated to represent acoustic variations between dialects (DeMarco and Cox, 2012; Shon *et al.*, 2018a). These features represent the speech signal at frame-level. To obtain low-dimensional and uncorrelated utterance-level representations, machine learning approaches, such as the Gaussian mixture model (GMM)–based i-vector model (Behravan *et al.*, 2016), Siamese network model (Siddhant *et al.*, 2017), and factorized hierarchical variational auto-encoder (FHVAE) model (Shon *et al.*, 2018b) were investigated.

Further, for better classification, linear classifiers [such as the support vector machine (SVM) and linear discriminant analysis (LDA)] and non-linear classifiers [such as feed-forward neural networks (FFNNs)] (DeMarco and Cox,

[a)]Electronic mail: rashmi.kethireddy@research.iiit.ac.in, ORCID: 0000-0002-3047-8158.
[b)]ORCID: 0000-0001-5806-3053.
[c)]ORCID: 0000-0001-6745-4363.

2012; Siddhant *et al.*, 2017) were investigated. In DeMarco and Cox (2012), i-vectors derived from MFCC features were investigated with different classifiers (SVM, LDA, iterative LDA, quadratic discriminant analysis (QDA), and iterative QDA) for classification of British English dialects. Out of them, iterative LDA classifiers performed better (accuracy of 68%).

Modern end-to-end deep neural classifiers can handle both compression and classification (Cai *et al.*, 2019; Qi *et al.*, 2018; Shon *et al.*, 2018a). The compressed latent representations learnt from these networks retain the temporal dependencies across the frames. However, neural network classifiers require larger amounts of data for training. To overcome this, different data-augmentation approaches are investigated in this study. Different weight initialization of the neural network can lead to unstable performances. To mitigate this, in this study, networks are trained multiple times and tested against each trained model, and then the performance is averaged across all models.

Deep neural classifiers were mainly investigated with convolution neural networks (CNNs) and recurrent neural networks (RNNs) for dialect classification (Cai *et al.*, 2019; Najafian *et al.*, 2018; Qi *et al.*, 2018; Shon *et al.*, 2018a; Wu *et al.*, 2018). From studies by Shon *et al.* (2018a,b), it was found that compared to traditional statistical methods (i-vectors+SVM), the end-to-end CNN architectures (with mel spectrogram as input) performed better by 10% absolute accuracy for Arabic English dialects. Further, it was shown that data augmentation improved the performance by 5.5% absolute accuracy. Even though RNNs were used for classification tasks in speech as they capture long temporal context, they also require $O(n)$ sequential operations for each unit while CNNs require $O(1)$ sequential operations. Lower order sequential operations for CNN lead to parallelization of computations in CNNs. In contrast, higher order sequential processing will lead to higher computation time for RNNs. Networks that provide similar temporal context, such as temporal convolution neural networks (TCNs) (Bai *et al.*, 2018) and time-delay neural networks (TDNNs) (Snyder *et al.*, 2018) with computation time similar to CNNs are investigated in this study. Significant architectural changes were made to TDNN to obtain emphasized channel attention, propagation and aggregation in TDNN (ECAPA-TDNN), which was shown to improve the performance of speaker verification system (Desplanques *et al.*, 2020) and language identification (Ravanelli *et al.*, 2021) is investigated in this study.

From the early studies on accent classification (Kat and Fung, 1999; Arslan and Hansen, 1997), it was found that the favourable spectral scale depends on the language of dialects and sub-dialects contained in it. Furthermore, from the accent classification studies with neural networks (Kethireddy *et al.*, 2020b), it was found that the distribution of learnt frequency bands are different from standard mel-scale distribution. It was observed that learnt scale showed an improvement of 10.94% UAR (relative) over mel-scale. Motivated by this, the current study introduces learnable spectral scale filters as a convolution layer and learnt along with the other network layers to discriminate dialects.

This study considers three major dialects of English, namely, Australian (AU), American (US), and British (UK) from UT-Podcast corpus (Hansen and Liu, 2016). The main challenges involved in the usage of this corpus for deep architectures is insufficient data for training and imbalanced classes. To overcome this, speed and volume perturbations are proposed in order to improve the training space and class balanced training to tackle the imbalanced classes. The initial study was conducted with UT-Podcast corpus by Hansen and Liu (2016) using traditional i-vector model and reported 74.5% UAR. Later, Wu *et al.* (2018) investigated deep neural classifier models, time distributed CNN with one attention layer, and frequency distributed CNN with two attention layers, which improved the performance of dialect classification system by 1.38% and 4.82% (in absolute UAR) over traditional i-vector model.

In this study, the features derived from two recently proposed signal processing methods, namely single frequency filtering (SFF) (Aneeja and Yegnanarayana, 2015) and zero-time windowing (ZTW) (Yegnanarayana and Dhananjaya, 2013), are explored for dialect classification. These methods were shown to provide higher spectro-temporal resolution compared to STFT (Aneeja and Yegnanarayana, 2015; Yegnanarayana and Dhananjaya, 2013). The SFF method was shown to provide better spectral features, such as harmonics, resonances (Chennupati *et al.*, 2019; Pannala *et al.*, 2016), and time-domain features, such as glottal closure instances and voice-onset time (VOT) (Kadiri and Yegnanarayana, 2017; Nellore *et al.*, 2017). Inspired by the advantages of SFF, mel filter-bank energies derived from SFF (MFBE–SFF) were investigated with an SVM classifier in our previous studies (Kethireddy *et al.*, 2020a), which showed promising results in identifying dialects compared to conventional STFT representations, such as the mel-spectrogram and MFCCs. In extension to the preliminary studies (Kethireddy *et al.*, 2020a), this study proposes to derive four different feature representations: namely (1) SFF spectrogram (SPEC–SFF), (2) single frequency filtered cepstral coefficients (SFFCCs), (3) mel filter-bank energies derived from SFF spectrum (MFBE–SFF), and (4) mel–frequency cepstral coefficients derived from SFF spectrum (MFCC–SFF).

In studies (Dhananjaya, 2011; Dhananjaya *et al.*, 2012; Yegnanarayana and Dhananjaya, 2013), ZTW spectrum was shown to differentiate different speech sound characteristics effectively compared to the STFT spectrum. In order to capture acoustic variations in the articulation of different dialects, the high spectral resolution of the ZTW spectrum could be helpful. Motivated by this, zero-time windowed cepstral coefficients (ZTWCCs) are investigated with SVM as a classifier in our preliminary studies (Kethireddy *et al.*, 2020c) and have shown promising results in identifying dialects compared to conventional STFT representations. In continuation to the preliminary work, this study proposes to derive four different feature representations: namely (1) ZTW spectrogram (SPEC–ZTW), (2) zero-time windowed cepstral coefficients (ZTWCCs), (3) mel filter-bank energies derived from ZTW spectrum (MFBE–ZTW), and (4) mel–frequency cepstral coefficients derived from ZTW spectrum (MFCC–ZTW). These four feature representations derived from

https://doi.org/10.1121/10.0009405

each method are used as input to advanced deep neural classifiers for dialect classification. To assist related work, we have made the code available (Kethireddy and Kadiri, 2022).

The major contributions of this study are as follows:

- Exploration of two recent signal processing methods (SFF and ZTW) that provides high spectro-temporal resolutions, and to derive four feature representations from SFF spectrum and ZTW spectrum for dialect classification.
- Exploration of recent deep neural architectures (TCN, TDNN, and ECAPA-TDNN) that provide long temporal context, along with traditional CNN for dialect classification.
- Introduced data-driven learnt spectral scale filters (as a convolution layer) instead of fixed mel-scale filters as used in traditional feature representations.
- Investigated the effectiveness of data-augmentation techniques (speed and volume perturbation) to handle an insufficient amount of data for training deep neural classifiers, and class balanced loss function to handle imbalanced classes in the corpus.

The organization of the article is as follows: Sec. II describes the SFF and ZTW methods along with the proposed feature representations derived from the SFF/ZTW spectrum. Section III gives the details of deep neural architectures investigated in this study. Details of the experimental setup, such as baseline feature configurations, proposed feature configurations, training configurations, and the corpus used are provided in Sec. IV. Results of the experiments with analysis are provided in Sec. V. Finally, Sec. VI gives a summary of the study.

II. SINGLE FREQUENCY FILTERING (SFF) AND ZERO-TIME WINDOWING (ZTW) METHODS AND EXTRACTION OF FEATURES

This section first describes two recently proposed signal processing methods, namely, SFF (Aneeja and Yegnanarayana, 2015; Kadiri and Yegnanarayana, 2017) and ZTW (Yegnanarayana and Dhananjaya, 2013) for deriving high-resolution spectrum, and then, gives a procedure to extract the proposed features from spectra of SFF and ZTW.

A. SFF method

SFF (Aneeja and Yegnanarayana, 2015) is a time-frequency analysis method that is used to compute an amplitude envelope of speech signal as a function of time at each selected frequency. In this method, the amplitude envelope at a particular frequency is obtained by first frequency-shifting (i.e., modulating) the speech signal ($s[n]$) (i.e., multiplying the $s[n]$ with an exponential function): $\hat{s}[n,k] = s[n]e^{j\hat{\omega}_k n}$, where $\hat{\omega}_k = \pi - 2\pi f_k/f_s$, f_k is the desired frequency and f_s is the sampling frequency. The frequency-shifted signal is filtered using a single pole filter, whose transfer function is given by: $H(z) = 1/1 + rz^{-1}$. The pole of the filter is located on the negative real axis (at $z = -r$). In this study, $r = 0.99$ is used, which is closer to the unit circle. The output of the filter is given by

$$y[n,k] = -ry[n-1,k] + \hat{s}[n,k]. \tag{1}$$

The amplitude envelope ($S_{SFF}[n,k]$) of $y[n,k]$ at frequency f_k is given by

$$S_{SFF}[n,k] = \sqrt{(y_r[n,k])^2 + (y_i[n,k])^2}, \tag{2}$$

where $y_r[n,k]$ is the real part and $y_i[n,k]$ is the imaginary part of $y[n,k]$. The amplitude envelopes can be computed for several frequencies at intervals of Δf by defining f_k as follows:

$$f_k = k\Delta f, \quad k = 1,2,\ldots,K, \tag{3}$$

where $K = (f_s/2)/\Delta f$. In this study, the value of Δf is chosen such that 1024 frequency samples exist in between 0 to f_s. From $S_{SFF}[n,k]$, the SFF magnitude spectrum (or SFF spectrum) can be obtained at each instant of time ("n") by considering all the amplitude envelope values at a particular time instant. However, in this study, an averaged SFF spectrum ($S_{SFF}[n,k]$) at regular intervals of 12.5 msec is considered. A schematic block diagram describing the steps involved in the computation of SFF spectrum is shown in Fig. 1.

B. ZTW method

ZTW method was proposed by Yegnanarayana and Dhananjaya (2013) to derive the instantaneous spectral characteristics, so that the time-varying characteristics of the speech production mechanism can be captured. In this method, the speech signal is windowed with a heavily decaying window (unlike the conventional Hamming window, etc.) that provides higher emphasis at the samples near the starting/zeroth time instant, and hence, the name zero-time windowing (ZTW). This heavily decaying window is shifted for every time instant and hence, the method provides higher temporal resolution. Spectrum is estimated using group delay that was shown to provide good spectral resolution. Hence, the method provides higher temporal resolution while simultaneously maintaining good spectral resolution. The steps involved in extracting the instantaneous spectral characteristics using the ZTW method are as follows:

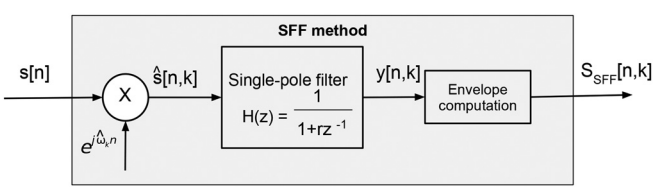

FIG. 1. Schematic block diagram describing the steps involved in the computation of SFF spectrum.

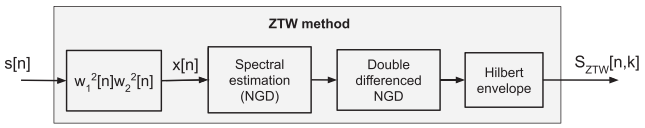

FIG. 2. Schematic block diagram describing the steps involved in the computation of ZTW spectrum.

- A segment of L *msec* speech signal $s[n]$ (number of samples: $M = Lf_s/1000$) is considered at each instant (i.e., $s[n]$ is defined for $n = 0, 1, \ldots, M-1$). The segment is multiplied with a heavily decaying window function $w_1^2[n]$, where

$$w_1[n] = 0, \qquad n = 0,$$
$$= \frac{1}{4\sin^2(\pi n/2N)}, \quad n = 1, 2, \ldots, N-1. \tag{4}$$

  N is the number of points used in the computation of discrete Fourier transform (DFT) ($N \gg M$). Multiplying the signal with $w_1^2[n]$ is approximately equivalent to integration in the frequency domain (Yegnanarayana and Dhananjaya, 2013). In this study, L = 25 msec and N = 1024 are chosen.
- Truncation of the signal at the instant $n = M-1$ may result in a ripple effect in the frequency domain. This effect can be reduced by using another window, $w_2[n]$, for $n = 0, 1, \ldots, M-1$, defined as:

$$w_2[n] = 2(1 + \cos(\pi n/M)) = 4\cos^2(\pi n/2M). \tag{5}$$

- The spectrum of the windowed signal (i.e., $x[n] = w_1^2[n]w_2[n]s[n]$) is computed using the numerator of the group delay (NGD) function ($g_n[k]$) given by

$$g_n[k] = X_R[k]Y_R[k] + X_I[k]Y_I[k], \quad k = 0, 1, 2, \ldots, N-1, \tag{6}$$

  where $X_R[k]$ is the real and $X_I[k]$ is the imaginary parts of the $X[k]$ (DFT of $x[n]$). Likewise, $Y_R[k]$ is the real and $Y_I[k]$ is the imaginary part of the $Y[k]$ (N-point DFT of $y[n] = nx[n]$).
- To highlight the hidden spectral characteristics due to heavily decaying window, the NGD function is differentiated twice. Then, the Hilbert envelope of the double-differentiated NGD is computed. This is referred to as the ZTW spectrum, denoted by $S_{ZTW}[n,k]$.

ZTW spectrum ($S_{ZTW}[n,k]$) can be obtained at every instant of time "n". However, in this study, the sub-sampled ZTW spectrum at regular intervals of 12.5 msec is considered. A schematic block diagram describing the steps involved in the computation of ZTW spectrum is shown in Fig. 2.

C. Extraction of feature representations from SFF/ZTW methods

This study proposes to derive four types of features from both SFF and ZTW spectra. They are: (1) SFF/ZTW spectrogram (SPEC–SFF/SPEC–ZTW), (2) cepstral coefficients derived from the SFF/ZTW spectrum (SFFCC/ZTWCC), (3) mel filter-bank energies derived from the SFF/ZTW spectrum (MFBE–SFF/MFBE–ZTW), and (4) mel–frequency cepstral coefficients derived from the SFF/ZTW spectrum (MFCC–SFF/MFCC–ZTW). Out of four features derived from SFF spectrum, only MFBE–SFF was investigated for dialect identification in Kethireddy *et al.* (2020a) and out of four features derived from ZTW spectrum, only ZTWCC was investigated for dialect classification in Kethireddy *et al.* (2020c). As per our knowledge, this is the first attempt to propose to use these feature representations for dialect classification. Illustrations of spectrograms obtained with STFT, SFF, and ZTW methods are shown in Figs. 3(a)–3(c), respectively. From the figures, it can be clearly seen that SFF spectrogram [Fig. 3(b)] highlights the harmonic structure (with sharper harmonics) compared to STFT spectrogram [Fig. 3(a)], even though both of them show similar formant structure. On the other hand, ZTW spectrogram clearly highlights the formant structure compared to STFT spectrogram.

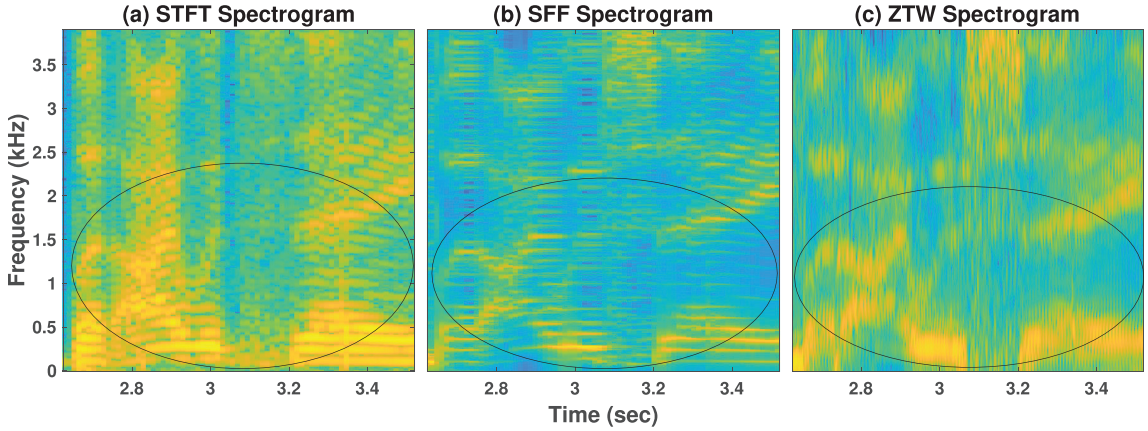

FIG. 3. (Color online) Illustration of spectrograms obtained with (a) STFT, (b) SFF, and (c) ZTW methods.

https://doi.org/10.1121/10.0009405

1. Extraction of SFF/ZTW spectrogram

The combination of SFF/ZTW spectrum at all the time instants gives the SFF/ZTW spectrogram. The logarithm of the SFF/ZTW spectrogram is used in this study which is referred to as SPEC–SFF/SPEC–ZTW.

2. Extraction of SFFCC/ZTWCC

SFFCC/ZTWCC are computed from the cepstrum of SFF/ZTW spectrum ($S_{SFF/ZTW}[n,k]$), as follows (Kadiri and Yegnanarayana, 2018a, 2018b):

$$C_{SFF/ZTW}[n,k] = \text{IFFT}(\log_{10}(S_{SFF/ZTW}[n,k])). \quad (7)$$

From cepstrum $C_{SFF/ZTW}[n,k]$, the first 80 coefficients are considered in this study. A schematic block diagram describing the steps involved in the extraction of SFFCC/ZTWCC is shown in Fig. 4(a).

3. Extraction of MFBE from SFF/ZTW spectrum (MFBE–SFF/MFBE–ZTW)

A schematic block diagram describing the steps involved in the extraction of MFBE from the SFF/ZTW spectrum is shown in Fig. 4(b). The MFBE extraction involves the computation of energies from the mel filter-banks placed on SFF/ZTW spectrum ($S_{SFF/ZTW}[n,k]$) followed by logarithm, and which can be expressed as follows:

$$MFBE_{SFF/ZTW}[n,k] = \log(Mel(S_{SFF/ZTW}[n,k]^2)). \quad (8)$$

These features are denoted as MFBE–SFF/MFBE–ZTW in this study. Here, 80 mel filters are integrated with the SFF/ZTW spectrum to obtain MFBE–SFF/MFBE–ZTW.

D. Extraction of MFCCs from SFF/ZTW spectrum (MFCC–SFF/MFCC–ZTW)

A schematic block diagram describing the steps involved in the extraction of MFCC from the SFF/ZTW spectrum is shown in Fig. 4(c). The MFCC extraction consists of the mel filter-bank analysis on the SFF/ZTW spectrum, followed by logarithm and discrete cosine transform (DCT) operations, and which can be expressed as follows (Kadiri and Alku, 2019):

$$MFCC_{SFF/ZTW}[n,k] = DCT(\log(Mel(S_{SFF/ZTW}[n,k]^2))), \quad (9)$$

where $MFCC_{SFF/ZTW}[n,k]$ denotes the mel-cepstrum. The resulting cepstral coefficients are referred to as MFCC–SFF/MFCC–ZTW, and they represent compactly the spectral characteristics. From the mel-cepstrum, all 80 cepstral coefficients (including the zeroth coefficient) are considered.

III. DEEP NEURAL ARCHITECTURES FOR DIALECT CLASSIFICATION

Figure 5 shows the schematic block diagram of the proposed dialect classification system. The proposed system consists of mainly two stages: (1) feature extraction, where feature representations from SFF- and ZTW-based methods are derived for dialect classification, and (2) classifier, where the deep neural classifiers, such as convolution neural network (CNN), temporal convolution neural network (TCN), time-delay neural network (TDNN), and emphasized channel attention, propagation and aggregation in TDNN (ECAPA-TDNN), are explored. Deep neural classifiers are trained with frame-level features from an entire utterance. The sub-sections in this section give the details of network architectures of CNN, TCN, TDNN, and ECAPA-TDNN.

A. Convolution neural network (CNN)

CNNs are most widely used deep neural architectures in speech (Abdel-Hamid *et al.*, 2012), text (Johnson and Zhang, 2017), and image processing (Lo *et al.*, 1995). CNNs were investigated previously for dialect classification with one-dimensional (1D) convolutions (Shon *et al.*, 2018a) and two-dimensional (2D) convolutions (Wu *et al.*,

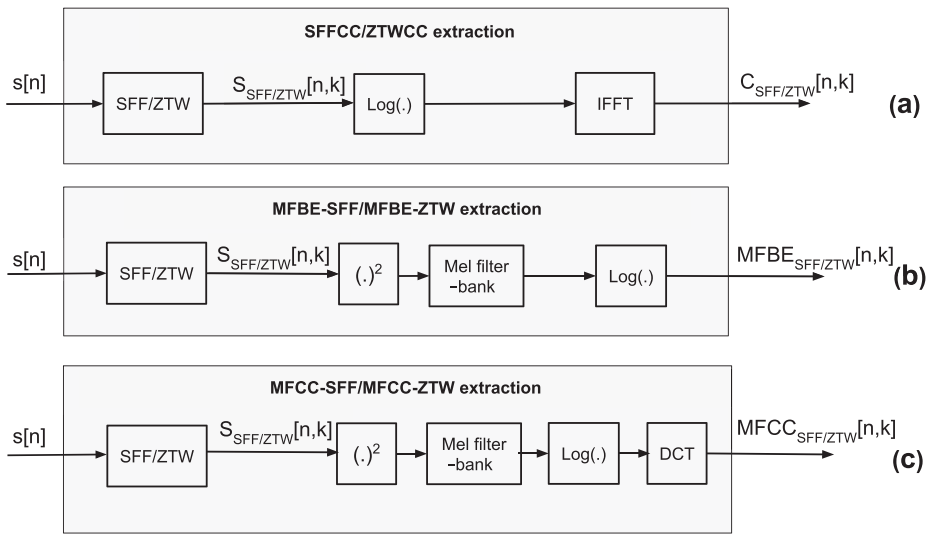

FIG. 4. Schematic block diagrams describing the steps involved in the extraction of features from SFF/ZTW method. (a) Steps involved in the extraction of SFFCC/ZTWCC. (b) Steps involved in the extraction of MFBE–SFF/MFBE–ZTW. (c) Steps involved in the extraction of MFCC–SFF/MFCC–ZTW.

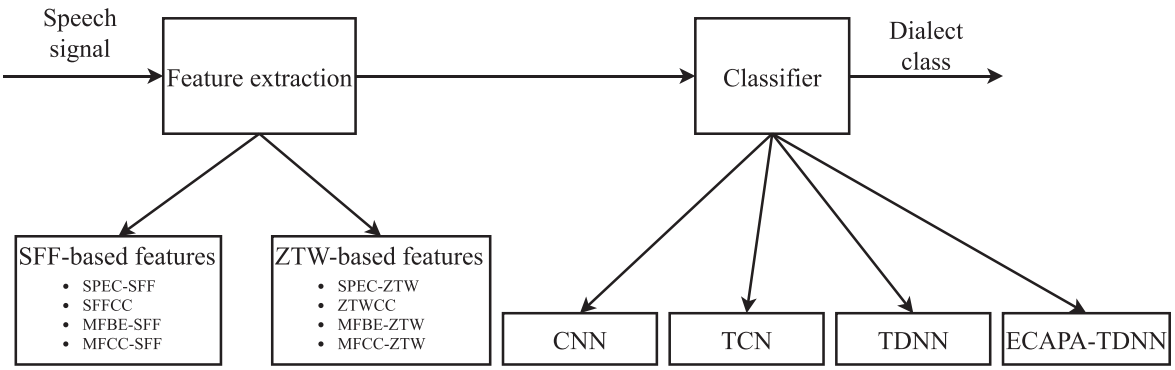

FIG. 5. A schematic block diagram of the proposed dialect classification system with proposed feature extraction methods and deep neural classifiers.

2018). Convolution neural network is usually formed by convolution layers (Conv), max-pooling, and fully connected (FC) feed-forward layers. The Conv layers of CNN extract the translation invariant and localized temporal features by striding over windows. The pooling layer compresses the segmental level information derived from the convolution layer to utterance-level information. FC layers are trained to classify the dialects. CNN with 1D convolution layers is investigated for dialect classification in this study.

Table I shows the architecture of the CNN classifier investigated in this study. The hyper-parameters that define the Conv layer are the number of filters (# filters), filter size, and stride, while the max-pool layer is defined only by kernel size and stride. FC layers are defined by input and output dimension. Columns of the table represent the layers of the CNN with configurations defined along rows. Convolution layers and max-pooling layers are segmental layers, and the layers after global average pool processes on utterance-level representations. Rectified linear unit (ReLU) activation is commonly applied in all the layers.

1. Spectral filters as convolution layer in CNN

Instead of using fixed mel-scale spectral filters in feature representations for input to CNN, data-driven learnt spectral scale filters (as convolution layer) for dialect classification are investigated. Note that learnt spectral scale filters are well known and previously used for speech recognition (Seki *et al.*, 2017), spoofing detection (Yu *et al.*, 2017), and accent classification (Kethireddy *et al.*, 2020b). As per our knowledge, this is the first attempt to propose to use learnt spectral scale filters for dialect classification. Figure 6 shows the schematic block diagram of a convolution layer of CNN that acts as learnable spectral filters. Given spectrogram as input, the spectrum at each time instant is integrated with a set of convolution filters (or learnable filters) to obtain data-driven learnt filter-bank energies which are further passed to other layers of CNN (as given in Table I). The learnable spectral filters are trained, along with other layers of the network, to classify dialects. The data-driven learnt scale is used to compress higher dimension spectrograms for dialect classification. For the Conv layer to match mel-scale spectral filters, 80 Conv filters (each initialized to triangular-shaped mel-scale spectral bands) and a stride of one frame (to obtain filter-bank energies for each frame) by Conv filter along the temporal axis. Further, the weights of convolution layer are constrained to have non-negative values during training.

B. Temporal convolution network (TCN)

TCN (Bai *et al.*, 2018) belongs to the family of CNNs with few constraints. The temporal convolution layers (Tconv) of TCN differ from CNNs by four architectural changes as given below:

(1) Each node of the temporal convolution (TConv) layer of the network is constrained only to the past information. This prevents leakage from future to past which is achieved by convolving with k frames in the past (k is the kernel size).
(2) TConv layers model sequentially resulting in same output length from each hidden layer. This is achieved by introducing zero-padding of length $(k-1)$ in each hidden layer.
(3) The convolutions in each layer are dilated to widen the temporal context without deepening the network. The receptive field at each layer is defined by $(k-1)*d$.
(4) Residual block that adds input to output before activation function.

TABLE I. End-to-end CNN architecture for dialect classification. Conv: convolution layer; FC: fully connected layer.

Layers	Conv1	Conv2	Max-pooling	Conv3	Conv4	Global average pool	FC1	FC2	FC3
No. filters/output dimensions	500	500	—	3000	3000	3000	1500	600	3
Kernel size	5	3	10	5	3	—	—	—	—
Stride	1	1	10	1	1	—	—	—	—

https://doi.org/10.1121/10.0009405

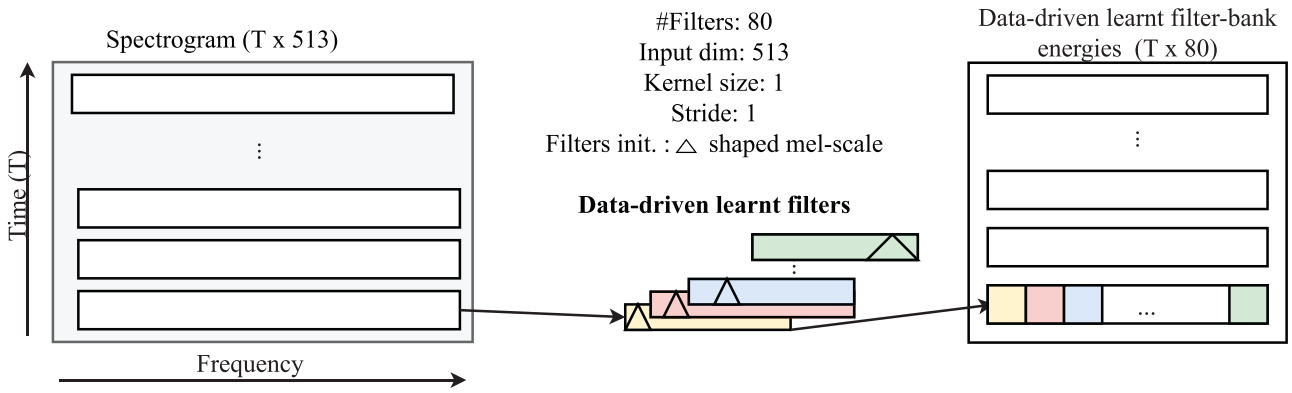

FIG. 6. (Color online) A schematic block diagram showing learnable spectral filters as a convolution layer initialized with mel-scaled triangular-shaped filters.

TCNs were previously explored in speech enhancement for sequential output processing that could replace RNNs with few network parameters and wider context (Pandey and Wang, 2019). Motivated by this, TCNs are investigated in a classification framework by adding pooling layers and fully connected layers as in CNNs.

Table II shows the architecture of the TCN classifier investigated in this study. The hyper-parameters that define the TConv layer are number of filters (#filters), kernel size, stride, and dilation. The layers after global average pool processes the dependencies across entire utterance.

C. Time-delay neural network (TDNN)

TDNNs also belong to the family of CNNs. TDNN differ from CNNs by introducing sub-sampling in higher layers that led to wider temporal context and does not lose much information due to correlated neighbourhood activations. They were first introduced for speech recognition (Waibel, 1989) and widely used in extraction of speaker embeddings (x-vectors) (Snyder *et al.*, 2018) and speech recognition (Peddinti *et al.*, 2015b). Apart from introducing the wider temporal context, the TDNNs also optimize the time and space complexity during training by reducing the operations (during forward pass and backward propagation) and the parameters of the network.

Table III shows the architecture of the TDNN classifier investigated in this study. The time-delay (TD) layers of TDNN are combined with pooling layers and FC layers as in CNNs. The hyper-parameters that define TD layer are input dimension, output dimension, and context. Along with them, the cumulative context of the layer is also defined in the table as total context. The first five TD layers process acoustic dependencies at the segmental level, while the layers after global average pooling process the utterance-level dependencies. The TD layers of TDNN used in this study are similar to the architecture defined in Snyder *et al.* (2018) for speaker embeddings.

D. Emphasized channel attention, propagation and aggregation in TDNN (ECAPA-TDNN)

Multiple enhancements were made to TDNN, which resulted in ECAPA-TDNN (Desplanques *et al.*, 2020). They are:

- **Squeeze-Excitation Res2Block** (SE Res2Block) combines the benefits of Squeeze-Excitation (SE) block (scales each channel according to global properties of the utterance) with the Res2Net module (computes multi-scale features with hierarchical residual connections within and reduces the model parameters) (Gao *et al.*, 2019). Skip connection is applied across an entire unit. The entire unit includes a dilated convolution [that is preceded (to reduce feature dimension) and succeeded (to restore feature dimension) by a dense layer] and a SE block.
- Multilayer feature aggregation and summation of feature maps from all three SE Res2Blocks to capture the relevant information from both shallow and deeper feature maps.
- Channel and context-dependent statistics pooling is used to convert variable length frame-level features to fixed length utterance-level features. This computation allows temporal attention scores for every channel.

The network architecture of ECAPA-TDNN in this study is similar to that in Ravanelli *et al.* (2021).

TABLE II. End-to-end TCN architecture for dialect classification. TConv: temporal convolution layer; FC: fully connected layer.

Layers	TConv1	TConv2	Max1	TConv3	TConv4	Global average pool	FC1	FC2	FC3
No. filters/Output dimensions	500	80	—	500	500	500	1500	600	3
Kernel size	5	3	10	5	3	—	—	—	—
Stride	1	1	10	1	1	—	—	—	—
Dilation	1	2	—	1	2	—	—	—	—

TABLE III. End-to-end TDNN architecture for dialect classification. t: current frame; T: the entire utterance; TD: time-delay layer; FC: fully connected layer, feat. dim.: dimension of input features at each frame.

Layers	TD1	TD2	TD3	TD4	TD5	Global average pool	FC1	FC2	FC3
Input dimensions	(feat. dim.)∗5	1536	1536	512	512	1500 T	1500	1500	600
Output dimensions	512	512	512	512	1500	1500	1500	600	3
Context	(t − 2, t + 2)	(t − 2, t, t + 2)	(t − 3, t, t + 3)	(t)	(t)	T	0	0	0
Total context	5	9	15	15	15	T	T	T	T

IV. EXPERIMENTAL PROTOCOL

This section describes the baseline feature configurations, proposed feature configurations, training configurations for deep neural classifiers, and the details of corpus used for dialect classification.

A. Baseline feature representations

Feature representations derived from STFT spectrum are considered as baseline due to their wider use in deep neural architectures for dialect classification (Shon *et al.*, 2018a). For computing STFT spectrum, speech signal is segmented into sliding windows and then each segment is transformed into frequency domain using Fourier transform. In this study, three feature representations derived from STFT spectrum are considered as baseline. They are: (1) STFT spectrogram (SPEC–STFT), (2) mel filter-bank energies derived from STFT spectrum (MFBE–STFT), and (3) mel–frequency cepstral coefficients derived from STFT (MFCC–STFT). STFT spectrum integrated with mel-scaled spectral filters and logarithm of the resultant gives MFBE–STFT. The cepstral coefficients derived from MFBE–STFT are referred to as MFCC–STFT.

In this study, the speech signal is segmented with a Hamming window of 25 msec length with shift equal to half of the window size (i.e.,12.5 msec). The number of DFT points considered in STFT spectrum computation are 1024. For MFBE–STFT extraction, spectrum is integrated with 80 mel-scaled filters. For each frame, the dimension is 80 for MFBE–STFT and MFCC–STFT, and 513 for SPEC–STFT.

B. Proposed feature configurations

For computing SFF spectrum, the root of the resonator r is set to 0.99 and the value of Δf is chosen such that 1024 frequency samples exist between $0 - f_s$. Instead of considering SFF spectrum at every instant, averaged spectrum for every 12.5 msec is considered, similar to baseline features. SFFCCs are derived from cepstrum of SFF spectrum. MFBE–SFF are extracted from SFF spectrum by integrating the spectrum with 80 mel filters and then applying the logarithm. MFCC–SFFs are the cepstral coefficients extracted from MFBE–SFF. For each frame, the dimension is 80 for SFFCC, MFBE–SFF and MFCC–SFF, and 513 for SPEC–SFF.

For computing ZTW spectrum, speech signal is segmented by a heavily decaying window of 25 msec length with a single sample shift. Instead of considering ZTW spectrum at every instant, sub-sampled spectrum for every 12.5 msec is considered, similar to baseline and SFF features. The number of DFT points used to compute ZTW spectrum is 1024. ZTWCCs are derived from cepstrum of ZTW spectrum. MFBE–ZTW are extracted from ZTW spectrum by integrating the spectrum with 80 mel filters and then applying logarithm. MFCC–ZTWs are the cepstral coefficients extracted from MFBE–ZTW. For each frame, the dimension is 80 for ZTWCC, MFBE–ZTW, and MFCC–ZTW, and 513 for SPEC–ZTW.

C. Training configuration

The deep neural classifiers are trained with the baseline and proposed features. The number of training epochs are decided approximately based on the loss convergence and over fitting. CNN and TCN models are trained for 50 epochs, TDNN is trained for 70 epochs, and ECAPA-TDNN is trained for 30 epochs. All the classifier models are trained to reduce cross-entropy loss with gradient descent optimizer. The learning rate of CNN, TCN, and TDNN is set to 0.001, and 0.0001 for ECAPA-TDNN. To mitigate the side-effect of the neural network weights initialization, networks are trained multiple times (six times for all the experiments) and tested against each trained model. The performance is averaged across all models, and mean and standard deviation of UAR (%) are reported for all the experiments.

To handle the imbalanced classes in the corpus, models are trained with class balanced loss function, which is expressed as (Cui *et al.*, 2019):

$$CB(\mathbf{p}, y) = \frac{1-\beta}{1-\beta^{n_y}} L(\mathbf{p}, y), \tag{10}$$

where $\mathbf{p}$ is a vector of class probabilities, computed by the classifier given as $[p_1, p_2, \ldots p_C]^T$, y is the class label that takes values between 0 to C, n_y is the class strength for class y, $\beta = N - 1/N$, and N is the total strength of the corpus.

D. Corpora: UT-Podcast

This study uses the UT-Podcast speech corpus which was collected from major dialects of English (Australian: AU, Britain: UK, and American: US) from the podcasts (Hansen and Liu, 2016). Among the three dialects, US is the majority class and UK is the minority class. Data were collected from adults with 127 male and 104 female speakers. Variations in pronunciation, vocabulary, and grammar that are common to groups of people are considered as dialect.

https://doi.org/10.1121/10.0009405

TABLE IV. Distribution of #utterances in each dialect class of the UT-Podcast (AU, UK, and US) before data augmentation and after data augmentation for train and test data.

UT-Podcast	Before data augmentation			After data augmentation		
Data type	AU	UK	US	AU	UK	US
Train	449	246	406	1347	738	1218
Test	332	89	240	332	89	240

These variations might be due to regional, social, or language differences. Within a region (either US, UK, or AU), sub-variants can exist but as per this corpus, only the major dialect of the speaker is considered. As the size of the corpus is small to train deep neural classifiers, data-augmentation strategy is used to generate more data for training. Table IV shows the distribution of UT-Podcast corpus before and after data augmentation. The number of utterances available for training in each of the dialects before data augmentation are: AU:449, UK:246, and US:406. Data are augmented using speed and volume perturbation approaches to increase the training space which resulted in: AU:1347, UK:738, and US:1218 utterances. Speed perturbation involves time warping of speech signal $s(t)$ by a factor of α to get $s(\alpha t)$ (Ko *et al.*, 2015; Shon *et al.*, 2018a). Volume perturbation involves simulation of different recording volumes (Peddinti *et al.*, 2015a; Shon *et al.*, 2018a). Speed perturbation with 0.9 and 1.1 factors, and volume perturbation with 1.5 factors resulted in thrice the size of the corpus. Perturbations are implemented using SoX audio manipulation tool (SoX, 2021). The sampling frequency of the corpus is 8 kHz.

V. RESULTS AND DISCUSSION

This section reports the dialect classification experimental results and analysis. First, the effect of data augmentation (speed and volume perturbations) to increase the training space for CNN classifier is investigated in Sec. V A. Second, the baseline feature representations (derived from STFT spectrum) and proposed feature representations (derived from SFF and ZTW spectra) are investigated for dialect classification with four deep neural classifiers (CNN, TCN, TDNN, and ECAPA-TDNN) in Sec. V B. Further, to better understand the performance of dialect classification systems with respect to each class, class-wise accuracies are also discussed in Sec. V B. Third, the effectiveness of data-driven learnt spectral filters (as convolution layer) are investigated, instead of fixed mel-scale spectral filters with CNN classifier for dialect classification, in Sec. V C. In Sec. V D, the proposed feature representations with deep neural classifiers are compared to the previous approaches in the literature that use the deep neural classifiers. Finally, in Sec. V E, to determine the effect of proposed features, the baseline and proposed features are investigated with VoxCeleb1 corpus for dialect classification (Nagrani *et al.*, 2017). UAR is used as a primary metric to evaluate the imbalanced classes better, as present in both UT-Podcast and VoxCeleb1 corpora. UAR is defined as unweighted average of class-wise accuracies (i.e., $UAR = \sum_{i=1}^{C} ACC_i/C$, where ACC_i is the accuracy of class i and C is the total number of classes). For all the experiments, networks are trained six times (to mitigate the side-effect of neural network weights initialization and validate the inconsistency due to the smaller size of the UT-Podcast corpus) and tested against each trained model. The performance is averaged across all models, and the mean and standard deviation of UAR (%) are reported for all the experiments with UT-Podcast.

A. Effect of data-augmentation

DNN architectures are constrained to have sufficiently large amounts of data for training. The UT-Podcast dialect corpus used in this study is relatively smaller, and hence, different levels of data augmentations (speed, volume, and both) are investigated with CNN classifier. The results without and with data augmentation are reported in Table V. In Table V, the third column (NP: no perturbation) reports the results without any data augmentation; the fourth column (SP) and fifth column (VP) report the results with speed perturbation and volume perturbation, respectively; and the final column (SVP) reports the results with a combination of speed and volume perturbations. Experiments were

TABLE V. Performance [mean and standard deviation of UAR (%) from six trails] of CNN classifier without data augmentation (NP), with speed perturbation (SP), with volume perturbation (VP), and with a combination of speed and volume perturbations (SVP).

Features	Feature type	NP	SP	VP	SVP
STFT-based features (baseline)	SPEC–STFT	63.62 ± 0.22	70.53 ± 0.28	66.55 ± 0.30	76.36 ± 0.36
	MFBE–STFT	60.69 ± 1.10	72.31 ± 0.56	67.39 ± 0.62	74.52 ± 0.68
	MFCC–STFT	58.74 ± 1.02	73.20 ± 0.09	61.91 ± 0.69	76.70 ± 0.56
SFF-based features (proposed)	SPEC–SFF	71.72 ± 1.09	79.14 ± 0.34	78.00 ± 0.49	77.91 ± 0.17
	SFFCC	69.84 ± 1.10	74.42 ± 0.19	73.39 ± 0.34	77.11 ± 0.50
	MFBE–SFF	73.74 ± 0.23	78.71 ± 0.37	74.09 ± 0.52	80.10 ± 0.57
	MFCC–SFF	73.99 ± 0.08	78.69 ± 0.36	76.61 ± 0.98	76.33 ± 0.68
ZTW-based features (proposed)	SPEC–ZTW	74.31 ± 1.65	73.50 ± 0.80	78.60 ± 1.56	75.87 ± 0.24
	ZTWCC	72.72 ± 0.58	73.06 ± 0.12	71.81 ± 0.19	74.69 ± 0.14
	MFBE–ZTW	73.82 ± 0.81	76.66 ± 0.54	75.28 ± 0.27	77.95 ± 0.41
	MFCC–ZTW	75.77 ± 0.26	73.92 ± 0.24	75.23 ± 0.46	76.22 ± 1.82

conducted with baseline feature representations (SPEC–STFT, MFBE–STFT, and MFCC–STFT) and proposed feature representations (SPEC–SFF/SPEC–ZTW, SFFCC/ZTWCC, MFBE–SFF/MFBE–ZTW, and MFCC–SFF/MFCC–ZTW) to choose the best data-augmentation approach for further experiments.

The mean and standard deviation of UAR (%) from six trails are reported in the table. From the standard deviation values, it can be observed that the accuracy is stable across multiple trails. From the experiments without data augmentation (NP, as in column 3), it can be observed that all the proposed SFF- (rows 5–8) and ZTW-based features (rows 9–12) performed better than baseline features (rows 2–4). With the individual data augmentation (SP and VP) and combination of data augmentations (SVP), it can be seen that the performance is improved for all the baseline and proposed features.

Among the baseline features, it can be observed that without data augmentation, SPEC–STFT performed better than MFBE–STFT and MFCC–STFT with a mean UAR of 63.62%. Speed and volume perturbations improved the performance, and applying both the perturbations together (SVP), improved the performance of SPEC–STFT, MFBE–STFT, and MFCC–STFT by 20.0%, 22.8%, and 30.6%, relatively, compared to without data augmentation (NP).

From the results of SFF-based features with NP (i.e., without data augmentation), it can be observed that SPEC–SFF, SFFCC, MFBE–SFF, and MFCC–SFF outperformed the best baseline feature (SPEC–STFT) by 12.7%, 9.8%, 15.9%, and 16.3% (relative UAR), respectively. Among the SFF-based features, MFBE–SFF and MFCC–SFF performed reasonably well with UAR of 73.74% and 73.99%. Independently, SP and VP improved the performances of all the SFF-based features. Applying both the perturbations together (SVP) improved the performances of SPEC–SFF, SFFCC, MFBE–SFF, and MFCC–SFF by 8.6%, 10.4%, 8.6%, and 3.2% (relative UAR), respectively. From the results of ZTW-based features with NP, it can be observed that SPEC–ZTW, ZTWCC, MFBE–ZTW, and MFCC–ZTW outperformed the best baseline SPEC–STFT by 16.8%, 14.3%, 16.0%, and 19.1% (relative UAR), respectively. Applying both the perturbations together (SVP) improved the performance of SPEC–ZTW, ZTWCC, MFBE–ZTW, and MFCC–ZTW by 2.1%, 2.7%, 5.6%, and 0.6% (relative UAR), respectively.

Overall, it can be observed that a combination of both speed and volume perturbations (SVP) gave better performance for all the feature representations (baseline and proposed). Hence, throughout this study (unless mentioned), the combination of speed and volume perturbations data is used to train the neural models for dialect classification.

B. Results of deep neural classifiers with the proposed feature representations

This section presents the dialect classification results with four deep neural classifiers (CNN, TCN, TDNN, and

TABLE VI. Performance [in mean and standard deviation of UAR (%) from six trails] of four deep neural classifiers (CNN, TCN, TDNN, and ECAPA-TDNN) for baseline (STFT-based) and proposed (SFF- and ZTW-based) features.

	STFT-based features			SFF-based Features				ZTW-based Features			
	(Baseline)			(Proposed)				(Proposed)			
Models	SPEC–STFT	MFBE–STFT	MFCC–STFT	SPEC–SFF	SFFCC	MFBE–SFF	MFCC–SFF	SPEC–ZTW	ZTWCC	MFBE–ZTW	MFCC–ZTW
CNN	76.36 ± 0.36	74.52 ± 0.68	**76.70 ± 0.56**	77.91 ± 0.17	77.11 ± 0.50	**80.10 ± 0.57**	76.33 ± 0.68	75.87 ± 0.24	74.69 ± 0.14	**77.95 ± 0.41**	76.22 ± 1.82
TCN	78.12 ± 0.46	**80.79 ± 0.74**	78.34 ± 0.77	80.84 ± 0.72	**81.30 ± 0.44**	78.58 ± 0.23	79.16 ± 0.47	**78.90 ± 0.59**	76.84 ± 2.07	77.98 ± 1.28	77.33 ± 1.08
TDNN	76.07 ± 0.72	**76.78 ± 0.37**	76.61 ± 0.35	77.65 ± 1.25	**81.53 ± 1.15**	77.76 ± 0.23	80.01 ± 0.22	**78.78 ± 0.58**	**78.42 ± 0.80**	75.95 ± 0.57	76.16 ± 0.12
ECAPA-TDNN	82.08 ± 0.41	81.70 ± 0.36	**84.87** ± 2.29	79.99 ± 0.19	**85.48 ± 0.51**	81.76 ± 0.69	83.06 ± 2.99	78.62 ± 0.51	**84.31 ± 0.69**	80.92 ± 1.01	76.50 ± 0.27

https://doi.org/10.1121/10.0009405

TABLE VII. Class-wise accuracies of dialect classification (three classes: AU, UK, and US) for baseline and proposed features with respect to four deep neural classifiers (CNN, TCN, TDNN, and ECAPA-TDNN).

		STFT-based features (Baseline)			SFF-based features (Proposed)				ZTW-based features (Proposed)			
Models	Class	SPEC–STFT	MFBE–STFT	MFCC–STFT	SPEC–SFF	SFFCC	MFBE–SFF	MFCC–SFF	SPEC–ZTW	ZTWCC	MFBE–ZTW	MFCC–ZTW
CNN	AU	78.46	91.01	81.93	87.1	85.54	85.49	80.22	89.61	88.25	68.62	65.51
	UK	62.36	53.93	63.11	60.11	61.42	75.28	61.61	57.68	50.75	83.89	76.91
	US	88.26	78.61	85.01	86.54	84.38	79.51	87.15	87.99	85.07	79.03	86.18
TCN	AU	86.90	84.69	81.48	91.77	76.60	81.73	81.73	87.80	91.87	84.59	88.51
	UK	53.37	63.86	66.11	64.80	77.72	62.55	63.67	56.18	53.37	60.68	54.12
	US	94.10	93.82	87.43	85.97	89.58	91.46	92.09	92.71	85.28	88.68	89.36
TDNN	AU	76.10	83.13	80.32	91.62	78.06	80.02	77.16	91.17	89.61	81.38	84.29
	UK	57.12	58.05	63.11	53.56	77.15	61.05	69.47	57.12	57.68	62.73	59.77
	US	95.0	89.17	86.39	87.78	89.37	92.22	93.40	88.06	87.99	83.75	84.45
ECAPA-TDNN	AU	77.71	80.12	81.93	80.42	85.24	81.63	83.73	78.61	73.19	74.70	67.47
	UK	79.78	74.16	79.78	65.17	82.02	67.42	77.53	65.17	87.64	76.40	70.79
	US	88.75	90.83	92.92	94.17	89.17	96.25	87.92	92.08	92.08	91.67	91.25

ECAPA-TDNN) for all the baseline features (STFT-based) and proposed (SFF- and ZTW-based) features. Table VI gives the performances in UAR (%) and Table VII gives the class-wise accuracies for baseline and proposed features. To see the discriminability among the dialect classes, non-linear t-distributed stochastic neighbor embedding (t-SNE) projections of latent features, derived from CNN classifier for best performing baseline feature (MFCC–STFT), proposed SFF-based feature (MFBE–SFF), and proposed ZTW-based feature (MFBE–ZTW) are shown in Fig. 7. Also, t-SNE projections of latent features derived from four deep neural classifiers with best performing feature (SFFCC) are shown Fig. 8. The t-SNE projections for all the features with all classifiers are provided (see Kethireddy and Kadiri, 2022).

The columns of Table VI report the results [in mean and standard deviation of UAR (%)] of features with respect to deep neural classifiers specified along the rows. Here, also, it can be observed from the standard deviation values that the accuracy is stable across multiple trails for all the classifiers. For discussion, first let us consider the results of CNN classifier (row 3 of Table VI) for baseline and proposed features. It can be observed that all the proposed SFF-based features performed better than baseline STFT-based features. On the other hand, among the proposed ZTW-based features, MFBE–ZTW feature performed better than baseline features and the remaining ZTW-based features performed similar to the baseline. Between SFF- and ZTW-based features, SFF-based features performed better than ZTW-based features. Among the baseline features, it can be observed that performance of SPEC–STFT and MFCC–STFT is better than MFBE–STFT. Among the SFF-based features, MFBE–SFF performed better than SPEC–SFF, SFFCC, and MFCC–SFF. Among the ZTW-based features, MFBE–ZTW performed better than the

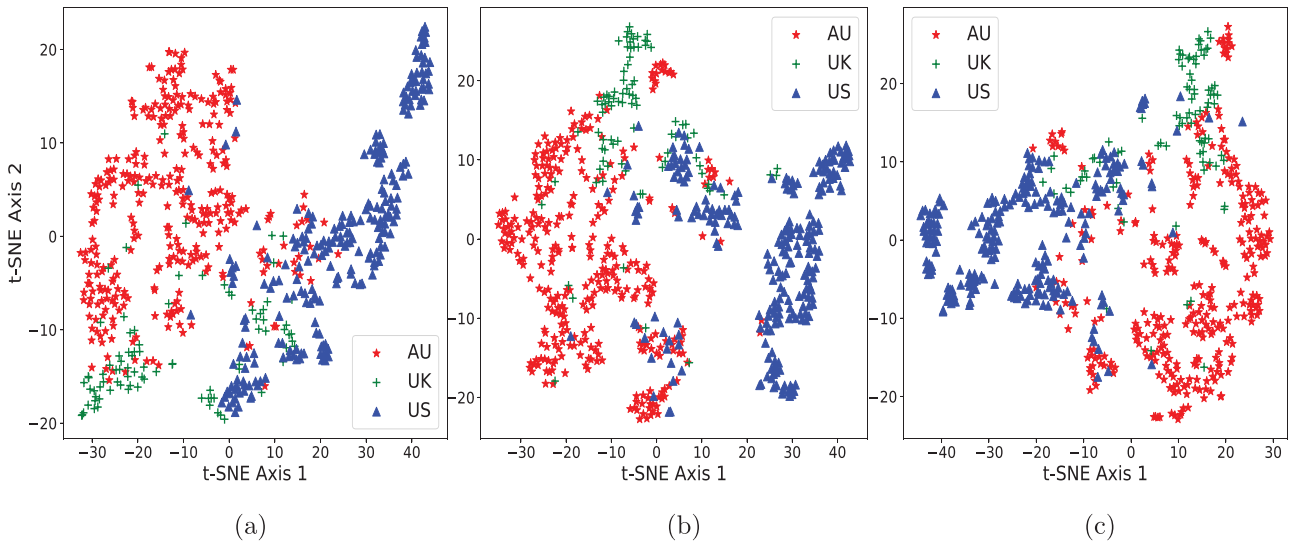

FIG. 7. (Color online) Plots showing t-SNE projections of the latent representations from the second fully connected layer (FC2) (see Sec. III) of CNN for (a) MFCC–STFT, (b) MFBE–SFF, and (c) MFBE–ZTW. Projections are color coded by their dialect class [AU:Red(∗), UK:Green(+), and US:Blue(Δ)].

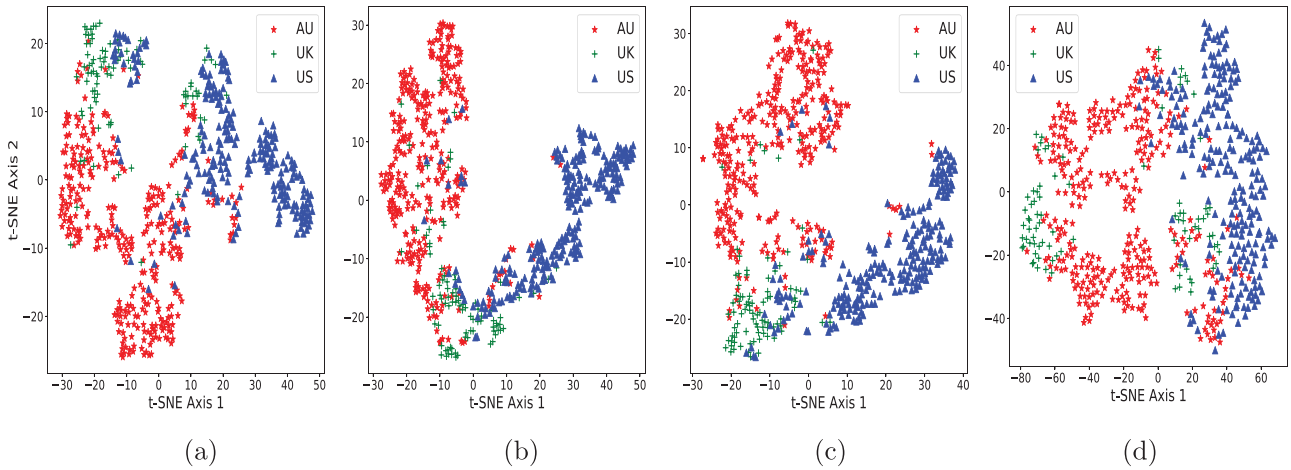

FIG. 8. (Color online) Plots showing t-SNE projections of the latent representations from the second fully connected layer (FC2) (see Sec. III) of (a) CNN, (b) TCN, (c) TDNN, and (d) ECAPA-TDNN for SFFCC features. Projections are color coded by their dialect class [AU:Red(∗), UK:Green(+), and US:Blue(Δ)].

remaining three (SPEC–ZTW, ZTWCC, and MFCC–ZTW). Overall, with the CNN classifier, it can be concluded that proposed features MFBE–SFF (80.10% UAR), SPEC–SFF (77.91% UAR), SFFCC (77.11% UAR), and MFBE–ZTW (77.95% UAR) performed better than the best baseline feature MFCC–STFT (76.70% UAR).

In comparison to CNN classifier, the results for TCN classifier (row 4 of Table VI) are better for all the baseline and proposed features. Again, it can be observed that the proposed SFF-based features (especially SFFCCs and SPEC–SFF) performed better than all the baseline features. ZTW-based features performed equally well or slightly less than baseline features. Between SFF- and ZTW-based features, SFF-based features performed better than ZTW-based features. Among the SFF-based features, SFFCCs gave the best performance (81.30 UAR %). Among the ZTW-based features, SPEC–ZTW gave the best performance (78.90 UAR %). The results of TDNN classifier (row 5 of Table VI) are better for some of the proposed features (SFFCC, MFCC–SFF, and ZTWCC) compared to CNN and TCN classifiers. Again, it can be seen that all the proposed SFF- and ZTW-based features performed better than all the baseline features (except MFBE–ZTW and MFCC–ZTW). Among the SFF-based features, SFFCCs gave best performance (with 81.53 UAR %). Among the ZTW-based features, SPEC–ZTW gave best performance (with 78.78 UAR %). The results for ECAPA-TDNN classifier (row 6 of Table VI) are better than all the other classifiers for all the baseline and proposed features. Note that the performance differences among STFT-, SFF-, and ZTW-based features are very small and within the error margins.

In summary, the performance of the proposed SFF- and ZTW-based features is comparable or better than baseline STFT-based features for all the four deep neural classifiers. This supports our hypothesis that the high spectral resolutions of SFF and ZTW spectra may help in improving dialect classification and could be an alternative feature representation for dialect discrimination. Among the four deep neural classifiers, TCN, TDNN, and ECAPA-TDNN gave better performance over CNN for many of the baseline and proposed features. This supports our hypothesis that the wider temporal context helped in improving dialect classification. Overall, SFFCCs with ECAPA-TDNN gave best dialect classification with UAR of 85.48%.

Table VII gives the class-wise accuracies of baseline and proposed features with four deep neural classifiers. From the table, it can be observed that results are biased towards the majority classes (AU and US) with lower performance for minority class (UK dialect) for all the features (except SFFCC) and classifiers (except ECAPA-TDNN). In case of CNN classifier, it can be observed that proposed features (especially MFBE–SFF, MFBE–ZTW, and MFCC–ZTW) are more accurate in classification of minority class compared to other proposed and baseline features. In case of TCN and TDNN classifiers, SFFCC features are more accurate in classification of minority class compared to all other features. In the case of ECAPA-TDNN classifier, both SFFCCs and ZTWCCs are more accurate in classification of minority class compared to all other features. Overall, the minority class is classified more accurately for almost all the features with ECAPA-TDNN classifier compared to other classifiers.

Discriminability among the dialect classes are visualized using t-SNE projections of latent features. Figure 7 shows the t-SNE projections of the latent features derived from the second fully connected layer of CNN classifier for the best performing baseline feature (MFCC–STFT) [Fig. 7(a)], proposed SFF-based feature (MFBE–SFF) [Fig. 7(b)], and the proposed ZTW-based feature (MFBE–ZTW) [Fig. 7(c)] (see row 3 of Table VI). It can be observed that all the projections of classes are better separated in MFBE–SFF [Fig. 7(b)] compared to MFCC–STFT [Fig. 7(a)] and MFBE–ZTW [Fig. 7(c)]. Whereas in Figs. 7(a) and 7(c), the projections of classes AU and US are well separated, and the projections of UK class are overlapped with AU and US. These projections

https://doi.org/10.1121/10.0009405

are synchronous with the class-wise accuracies reported in Table VII with CNN.

Figure 8 shows the t-SNE projections of the latent features derived from four deep neural classifiers, CNN [Fig. 8(a)], TCN [Fig. 8(b)], TDNN [Fig. 8(c)], and ECAPA-TDNN [Fig. 8(d)], trained with the best performing feature (SFFCCs). From t-SNE projections of CNN [Fig. 8(a)], it can be observed that the projections of classes AU and US are well separated, and the projections of UK class are overlapped with AU and US. Whereas from t-SNE projections of TCN [Fig. 8(b)], TDNN [Fig. 8(c)], and ECAPA-TDNN [Fig. 8(d)], all the classes are relatively better separated when compared to CNN. These observations are in conformity with the class-wise accuracies reported in Table VII for SFFCC features.

C. Investigation of data-driven learnt spectral filters

Based on the hypothesis that spectral scale depends on the language of dialects for dialect classification, learnable spectral scale filters (as convolution layer) are investigated as discussed in Sec. III A 1 instead of fixed mel-scale spectral filters. Table VIII shows the performances [in UAR (%)] of three spectral representations (i.e., spectrograms of STFT, SFF, and ZTW) integrated with fixed mel-scale filters and learnable-scale filters (represented as convolution layer). From the table, it can be observed that data-driven learnt filters performed better than fixed mel-scale filters for STFT and SFF spectrograms. Whereas in the case of ZTW spectrograms, fixed mel-scale filters performed equally well as learnt filters. It can be concluded that learnt filters retained relevant information required for classification in STFT and SFF spectrograms.

D. Comparison with previous studies

This section compares the results obtained for UT-Podcast corpus by the previous approaches (Wu *et al.*, 2018) that uses DNNs and the current studies (with both baseline and proposed features). In the previous study (Wu *et al.*, 2018), the strength of utterances belonging to the minority class (UK) are re-sampled for training. They investigated six different neural architectures [feed-forward neural network (FFNN), five-layer CNN, AlexNet (Krizhevsky *et al.*, 2012), VGG-11 (Simonyan and Zisserman, 2015), ResNet-18 (He *et al.*, 2016), and FreqCNN] with STFT spectrogram as input. Feed-forward neural network is a small deep neural classifier with three fully connected layers. Five-layer CNN is a deep neural classifier with five 2D convolution layers followed by fully connected layers. AlexNet (Krizhevsky *et al.*, 2012), VGG-11 (Simonyan and Zisserman, 2015), and ResNet (He *et al.*, 2016) are typical deep neural architectures belong to a family of CNNs with varied numbers of convolution layers. FreqCNN is proposed in Wu *et al.* (2018), and its architecture comprises of attention based convolution blocks along with basic convolution blocks. Note that Lu *et al.* (2020) reported UAR of 94% which is due to a mismatch in the experimental protocol, especially in the data distributions of training and testing, as opposed to the initial study of UT-Podcast in Hansen and Liu (2016), where the authors provided train and test sets which are collected from entirely different websites to have open test conditions. Hence, this study (Lu *et al.*, 2020) is not considered for comparison.

For a fair comparison, the UK class is re-sampled as in Wu *et al.* (2018) for the experiments conducted in this section. Table IX shows the results (UAR and class-wise accuracies) from previous studies in Wu *et al.* (2018) that uses different neural networks with SPEC–STFT as input, and the results of proposed and baseline features with CNN-1D classifier. The UAR (%) and class-wise accuracies of the current studies are the mean values from six trails. Among

TABLE VIII. Performance [in mean and standard deviation of UAR (%) from six trails] of CNN classifier trained with spectrograms of STFT, SFF, and ZTW integrated with mel-scale filters and learnable-scale filters (spectral scale as convolution layer).

	Spectral filters	
Feature type	Mel-scale	Learnable-scale
STFT	74.52 ± 0.6	76.60 ± 0.25
SFF	80.10 ± 0.57	81.25 ± 0.44
ZTW	77.95 ± 0.41	77.41 ± 1.21

TABLE IX. Performance in UAR (%) (mean and standard deviation from six trails) and class-wise accuracies (of classes AU, UK, and US) for different deep neural architectures from previous studies and current studies with all the features (STFT, SFF, and ZTW) using CNN classifier (for similar data configurations).

			Class-wise accuracies		
Input feature type	Arch. type	UAR	AU	UK	US
Previous studies (Wu *et al.*, 2018)					
SPEC–STFT	FFNN	61.42	70.78	50.56	62.92
	Five-layer CNN	62.81	64.76	41.57	82.0
	AlexNet	**64.90**	58.43	64.04	74.17
	VGG-11	54.40	55.72	48.31	59.17
	ResNet-18	61.66	69.28	38.20	77.50
	FreqCNN	**79.32**	88.55	71.91	77.50
Current studies: STFT-based features					
SPEC–STFT	CNN	**74.05 ± 0.33**	72.94	77.90	71.60
MFBE–STFT		71.96 ± 0.34	69.23	69.29	76.67
MFCC–STFT		71.58 ± 0.30	70.18	68.73	76.67
Current studies: SFF-based features					
SPEC–SFF	CNN	**80.81 ± 0.30**	82.63	89.89	70.35
SFFCC		79.32 ± 0.34	87.40	71.35	77.57
MFBE–SFF		**80.72 ± 0.20**	87.35	75.84	77.71
MFCC–SFF		**80.38 ± 0.41**	87.20	74.91	77.91
Current studies: ZTW-based features					
SPEC–ZTW	CNN	**79.63 ± 0.22**	83.68	80.15	74.58
ZTWCC		78.72 ± 0.44	79.77	84.27	71.11
MFBE–ZTW		78.69 ± 0.21	86.90	70.97	76.73
MFCC–ZTW		78.33 ± 0.30	86.30	71.72	76.25

the six different DNNs from previous studies (Wu *et al.*, 2018), it can be observed that FreqCNN performed better (with 79.32% UAR) than other classifiers. On the other hand, it can be observed that current studies with all the proposed features (especially SFF-based features) performed better than the previous studies. From the current studies with the baseline STFT-based features, SPEC–STFT (74.05% UAR) performed better than other STFT-based features. The proposed SFF-based features (SPEC–SFF, SFFCC, MFBE–SFF, and MFCC–SFF) outperformed the best performing baseline feature (SPEC–STFT) by 9.1%, 7.1%, 9.0%, and 8.5% (relative UAR), respectively. The ZTW-based features (SPEC–ZTW, ZTWCC, MFBE–ZTW, and MFCC–ZTW) outperformed the best performing baseline feature by 7.5%, 6.3%, 6.3%, and 5.8% (relative UAR), respectively. Overall, it can be observed that performance obtained with the proposed SFF- and ZTW-based features is superior to the baseline features and previous studies (except FreqCNN). Among all, SPEC–SFF with basic CNN gave the highest performance with a UAR of 80.81%, which is 1.9% (relatively) higher than the performance of FreqCNN with SPEC–STFT (79.32%). Investigating proposed features with FreqCNN may further improve the performance.

From the comparison of class-wise accuracies among previous studies, it can be observed that other than AlexNet and FreqCNN, all the classifiers identified the UK dialect with less than 50%. However, AlexNet lacked its performance in identifying the AU dialect. On the other hand, almost all the proposed features identified the UK dialects with accuracy more than 70% without lacking performance in other dialect classes (AU and US). Current studies with both baseline and proposed features outperformed all the architectures (except FreqCNN) of previous studies with similar data configurations.

E. Investigation with VoxCeleb

The effectiveness of proposed features is also investigated for the VoxCeleb1 dataset (Nagrani *et al.*, 2017). This dataset was collected from 1251 celebrities that span a wide range of nationalities. For the dialect classification experiments, speech data of four nationalities [AU, Canadian English (CA), UK, and US] are considered, based on the amount of available data for each class. Table X shows the data statistics of the VoxCeleb1 corpus considered in this study. Details about the data splits are provided (see Kethireddy and Kadiri, 2022).

For a fair comparison, only MFCCs derived from STFT, SFF, and ZTW methods are considered with CNN as a classifier. Table XI shows the performance in UAR (%) and class-wise accuracies. From the results in the table, it can be observed that MFCC–ZTW performed better than MFCC–SFF and MFCC–STFT features. From the class-wise accuracies, it can be observed that the minority classes, CA and AU, are less accurately classified for both baseline and proposed features.

TABLE X. Distribution of #utterances in each dialect class of VoxCeleb1 (AU, CA, UK, and US) for training, validation, and test data.

Data type	AU	CA	UK	US
Train	3223	3941	19 408	66 415
Validation	696	842	4173	14 215
Test	674	819	4192	14 241

TABLE XI. Performance in UAR (%) and class-wise accuracies (AU, CA, UK, and US) for CNN trained with MFCC–STFT, MFCC–SFF, and MFCC–ZTW for VoxCeleb1 corpus.

		Class-wise accuracies			
Feature type	UAR	AU	CA	UK	US
MFCC–STFT	77.6	67.55	60.12	86.74	95.99
MFCC–SFF	75.07	65.73	52.07	86.50	95.99
MFCC–ZTW	78.46	69.78	59.18	89.19	95.67

VI. SUMMARY AND CONCLUSION

This study explored the features derived from high spectro-temporal resolution of SFF and ZTW methods with deep neural classifiers for dialect classification. From SFF/ZTW spectra, four different feature representations (SPEC–SFF/SPEC–ZTW, SFFCC/ZTWCC, MFBE–SFF/MFBE–ZTW, and MFCC–SFF/MFCC–ZTW) were derived. Further, TCN, TDNN, and ECAPA-TDNN deep neural classifiers were investigated along with the traditional CNN.

From initial experiments with CNN classifier, it was found that data augmentation improved the performance of both baseline (STFT-based) and proposed (SFF- and ZTW-based) features. Further, it was found that proposed features outperformed the baseline features in both with and without data augmentation.

From the results with TCN classifier, it was found that SFF-based features, such as SPEC–SFF, SFFCC, and MFCC–SFF, improved their performance relatively by 3.8%, 5.4%, and 3.7%, and ZTW-based features, such as SPEC–ZTW, ZTWCC, and MFCC–ZTW, improved their performance relatively by 4.0%, 2.9%, 1.5%, respectively, compared to the results obtained with CNN classifier. From the results with TDNN classifier, it was found that SFFCC, MFCC–SFF, SPEC–ZTW, and ZTWCC features improved relatively by 5.7%, 4.8%, 3.8%, and 5.0%, respectively, compared to the results obtained with CNN classifier. From the results with ECAPA-TDNN classifier, it was found that SPEC–SFF, SFFCC, MFBE–SFF, MFCC–SFF, SPEC–ZTW, ZTWCC, MFBE–ZTW, and MFCC–ZTWCC features improved relatively by 2.6%, 10.9%, 2.1%, 8.8%, 3.6%, 12.9%, 3.8%, and 0.4%, compared to the results obtained with CNN classifier.

Overall, the proposed SFF- and ZTW-based features gave comparable or better performance over baseline STFT-based features for all the four deep neural classifiers, which supports our hypothesis that the high spectro-temporal resolution of SFF and ZTW spectra helps in improving dialect classification. It was also found that among the four deep neural classifiers, ECAPA-TDNN performed better than

CNN, TCN, and TDNN in many cases. The best dialect classification performance was achieved using SFFCC features with ECAPA-TDNN classifier (85.48% UAR). Further, data-driven learnt spectral scale filters were investigated and found that learnt scale filters performed better than fixed mel-scale filters with STFT and SFF spectrograms. In summary, as the performance of the proposed features is comparable or better than baseline STFT-based features, they can be used as an alternative or complimentary features for similar tasks, such as accent, language, and speaker identification.

ACKNOWLEDGMENTS

R. Kethireddy would like to thank the University Grants Commission India [Project No. 3582/(NET-NOV2017)] for supporting her PhD. S. R. Kadiri would like to thank the Academy of Finland (Project No. 330139) for supporting him as a Research Fellow.

Abdel-Hamid, O., Mohamed, A., Jiang, H., and Penn, G. (**2012**). "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 25–30 March 2012, pp. 4277–4280.

Aneeja, G., and Yegnanarayana, B. (**2015**). "Single frequency filtering approach for discriminating speech and nonspeech," IEEE Trans. Audio Speech Lang. Process. **23**(4), 705–717.

Arslan, L. M., and Hansen, J. H. (**1997**). "A study of temporal features and frequency characteristics in American English foreign accent," J. Acoust. Soc. Am. **102**(1), 28–40..

Bai, S., Kolter, J. Z., and Koltun, V. (**2018**). "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv:1803.01271.

Behravan, H., Hautamäki, V., Siniscalchi, S. M., Kinnunen, T., and Lee, C. (**2016**). "i-vector modeling of speech attributes for automatic foreign accent recognition," IEEE Trans. Audio Speech Lang. Process. **24**(1), 29–41.

Bougrine, S., Cherroun, H., and Ziadi, D. (**2018**). "Prosody-based spoken Algerian Arabic dialect identification," Procedia Comput. Sci. **128**, 9–17.

Cai, W., Cai, D., Huang, S., and Li, M. (**2019**). "Utterance-level end-to-end language identification using attention-based CNN-BLSTM," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 12–17 May 2019, pp. 5991–5995.

Chen, N. F., Shen, W., Campbell, J. P., and Torres-Carrasquillo, P. A. (**2011**). "Informative dialect recognition using context-dependent pronunciation modeling," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 22–27 May 2011, pp. 4396–4399.

Chen, N. F., Tam, S. W., Shen, W., and Campbell, J. P. (**2014**). "Characterizing phonetic transformations and acoustic differences across English dialects," IEEE Trans. Audio Speech Lang. Process. **22**(1), 110–124.

Chennupati, N., Kadiri, S. R., and Yegnanarayana, B. (**2019**). "Spectral and temporal manipulations of sff envelopes for enhancement of speech intelligibility in noise," Comput. Speech Lang. **54**, 86– 105.

Cui, Y., Jia, M., Lin, T., Song, Y., and Belongie, S. (**2019**). "Class-balanced loss based on effective number of samples," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, 16–20 June 2019, pp. 9260–9269.

DeMarco, A., and Cox, S. J. (**2012**). "Iterative classification of regional British accents in i-vector space," in *Proceedings on the Symposium on Machine Learning in Speech and Language Processing (MLSPL)*, Portland, OR, 14 September 2012, pp. 1–4.

Desplanques, B., Thienpondt, J., and Demuynck, K. (**2020**). "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," arXiv:2005.07143.

Dhananjaya, N. (**2011**). "Signal processing for excitation-based analysis of acoustic events in speech," Ph.D. thesis, IIT Madras, Chennai.

Dhananjaya, N., Yegnanarayana, B., and Bhaskararao, P. (**2012**). "Acoustic analysis of trill sounds," J. Acoust. Soc. Am. **131**(4), 3141–3152.

Gao, S., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., and Torr, P. H. (**2019**). "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Hansen, J. H., and Liu, G. (**2016**). "Unsupervised accent classification for deep data fusion of accent and language information," Speech Commun. **78**, 19–33.

He, K., Zhang, X., Ren, S., and Sun, J. (**2016**). "Deep residual learning for image recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 27–30 June 2016, pp. 770–778.

Johnson, R., and Zhang, T. (**2017**). "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the Association for Computational Linguistics (ACL)*, Vancouver, Canada, 31 July–2 August 2017, pp. 562–570.

Kadiri, S. R., and Alku, P. (**2019**). "Mel-frequency cepstral coefficients derived using the zero-time windowing spectrum for classification of phonation types in singing," J. Acoust. Soc. Am. **146**(5), EL418–EL423.

Kadiri, S. R., and Yegnanarayana, B. (**2017**). "Epoch extraction from emotional speech using single frequency filtering approach," Speech Commun. **86**, 52–63.

Kadiri, S. R., and Yegnanarayana, B. (**2018a**). "Analysis and detection of phonation modes in singing voice using excitation source features and single frequency filtering cepstral coefficients (SFFCC)," in *Proceedings of Interspeech*, Hyderabad, India, 2–6 September 2018, pp. 441–445.

Kadiri, S. R., and Yegnanarayana, B. (**2018b**). "Breathy to tense voice discrimination using zero-time windowing cepstral coefficients (ZTWCCs)," in *Proceedings of Interspeech*, Hyderabad, India, 2–6 September 2018, pp. 232–236.

Kat, L. W., and Fung, P. (**1999**). "Fast accent identification and accented speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Phoenix, AZ, 15–19 March 1999, Vol. 1, pp. 221–224.

Kethireddy, R., Kadiri, S. R., Alku, P., and Gangashetty, S. V. (**2020a**). "Mel-weighted single frequency filtering spectrogram for dialect identification," IEEE Access **8**, 174871–174879.

Kethireddy, R., Kadiri, S. R., and Gangashetty, S. V. (**2020b**). "Learning filterbanks from raw waveform for accent classification," in *Proceedings of the International Joint Conference on Neural Networks*, Glasgow, UK, 19–24 July 2020, pp. 1–6.

Kethireddy, R., Kadiri, S. R., Kesiraju, S., and Gangashetty, S. V. (**2020c**). "Zero-time windowing cepstral coefficients for dialect classification," in *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, Tokyo, Japan, 1–5 November 2020, pp. 32–38.

Kethireddy, R., and Kadiri, S. R. (**2022**). "Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations," https://github.com/r39ashmi/e2e_dialect (Last viewed 21 December 2021).

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. (**2015**). "Audio augmentation for speech recognition," in *Proceedings of Interspeech*, Dresden, Germany, 6–10 September 2015, pp. 3586–3589.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (**2012**). "ImageNet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, Lake Tahoe, NV, 3–6 December 2012, pp. 1106–1114.

Lo, S. C. B., Chan, H. P., Lin, J. S., Li, H., Freedman, M. T., and Mun, S. K. (**1995**). "Artificial convolution neural network for medical image pattern recognition," Neural Netw. **8**(7-8), 1201–1214.

Lu, H., Zhang, H., and Nayak, A. (**2020**). "A deep neural network for audio classification with a classifier attention mechanism," arXiv preprint arXiv:2006.09815.

Nagrani, A., Chung, J. S., and Zisserman, A. (**2017**). "Voxceleb: A large-scale speaker identification dataset," arXiv:1706.08612.

Najafian, M., Khurana, S., Shan, S., Ali, A., and Glass, J. (**2018**). "Exploiting convolutional neural networks for phonotactic based dialect identification," in *Proceedings of the International Conference on*

Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018, pp. 5174–5178.

Nellore, B. T., Prasad, R., Kadiri, S. R., Gangashetty, S. V., and Yegnanarayana, B. (**2017**). "Locating burst onsets using SFF envelope and phase information," in *Proceedings of Interspeech*, Stockholm, Sweden, 20–24 August 2017, pp. 3023–3027.

Pandey, A., and Wang, D. (**2019**). "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 12–17 May 2019, pp. 6875–6879.

Pannala, V., Aneeja, G., Kadiri, S. R., and Yegnanarayana, B. (**2016**). "Robust estimation of fundamental frequency using single frequency filtering approach," in *Proceedings of Interspeech*, San Francisco, CA, 8–12 September 2016, pp. 2155–2159.

Peddinti, V., Chen, G., Manohar, V., Ko, T., Povey, D., and Khudanpur, S. (**2015a**). "JHU ASpIRE system: Robust LVCSR with TDNNs, ivector adaptation and RNN-LMS," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, AZ, 13–17 December 2015, pp. 539–546.

Peddinti, V., Povey, D., and Khudanpur, S. (**2015b**). "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, 19–24 April 2015, pp. 3214–3218.

Qi, Z., Ma, Y., Gu, M., Jin, Y., Li, S., Zhang, Q., and Shen, Y. (**2018**). "End-to-end Chinese dialect identification using deep feature model of recurrent neural network," in *Proceedings of the International Conference on Computer and Communications (ICCC)*, Chengdu, China, 7–10 December 2018, pp. 2148–2152.

Rajpal, A., Patel, T. B., Sailor, H. B., Madhavi, M. C., Patil, H. A., and Fujisaki, H. (**2016**). "Native language identification using spectral and source-based features," in *Proceedings of Interspeech*, San Francisco, CA, 8–12 September 2016, pp. 2383–2387.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. (**2021**). "SpeechBrain: A general-purpose speech toolkit," arXiv:2106.04624.

Rouas, J. (**2007**). "Automatic prosodic variations modeling for language and dialect discrimination," IEEE Trans. Audio. Speech, Lang. Process. **15**(6), 1904–1911.

Seki, H., Yamamoto, K., and Nakagawa, S. (**2017**). "A deep neural network integrated with filterbank learning for speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 5–9 March 2017, pp. 5480–5484.

Shon, S., Ali, A., and Glass, J. (**2018a**). "Convolutional neural network and language embeddings for end-to-end dialect recognition," in *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, Les Sables d'Olonne, France, 26–29 June 2018, pp. 98–104.

Shon, S., Hsu, W.-N., and Glass, J. (**2018b**). "Unsupervised representation learning of speech for dialect identification," in *IEEE Spoken Language Technology Workshop*, Athens, Greece, 18–21 December 2018, pp. 105–111.

Siddhant, A., Jyothi, P., and Ganapathy, S. (**2017**). "Leveraging native language speech for accent identification using deep siamese networks," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan, 16-20 December 2017, pp. 621–628.

Simonyan, K., and Zisserman, A. (**2015**). "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, 7–9 May 2015.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (**2018**). "X-vectors: Robust dnn embeddings for speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 15–20 April 2018, pp. 5329–5333.

SoX. (**2021**). "Audio manipulation tool," http://sox.sourceforge.net/ (Last viewed 21 December 2021).

Waibel, A. (**1989**). "Modular construction of time-delay neural networks for speech recognition," Neural Comput. **1**(1), 39–46.

Wu, Y., Mao, H., and Yi, Z. (**2018**). "Audio classification using attention-augmented convolutional neural network," Knowl. Based Syst. **161**, 90–100.

Yegnanarayana, B., and Dhananjaya, N. (**2013**). "Spectro-temporal analysis of speech signals using zero-time windowing and group delay function," Speech Commun. **55**(6), 782–795.

Yu, H., Tan, Z.-H., Zhang, Y., Ma, Z., and Guo, J. (**2017**). "DNN filter bank cepstral coefficients for spoofing detection," IEEE Access **5**, 4779–4787.