The Journal of the Acoustical Society of America

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

--Manuscript Draft--

Manuscript Number:	JASA-07205
Full Title:	Deep neural architectures for dialect classification with single frequency filtering and zero−time windowing feature representations
Article Type:	Regular Article
Section/Category:	Speech Communication
Keywords:	dialect classification; deep neural networks; single frequency filtering (SFF); zero-time windowing (ZTW); spectrogram; MFCCs; CNN; TCNN; TDNN
Abstract:	The goal of this study is to investigate advanced signal processing approaches (SFF and ZTW) with modern DNNs (CNN, TCN, and 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 between dialects, four feature representations are derived from SFF and ZTW methods: SFF/ZTW spectrogram (SPEC-SFF/SPEC-ZTW), SFF/ZTW cepstral coefficients (SFFCC/ZTWCC), mel filter-bank energies from SFF/ZTW (MFBE-SFF/MFBE-ZTW), and MFCCs from SFF/ZTW (MFCC-SFF/MFCC-ZTW). Experiments with and without data-augmentation using CNN classifier revealed that the proposed features performed better than baseline STFT-based features. Even without data-augmentation, all the proposed features gave an improvement of around 15-20% (relative) over best baseline (SPEC-STFT) feature. TCN and TDNN classifiers that captures wider temporal-context further improved the performance for many of the proposed and baseline features. The best performance is achieved with SFFCCs for both TCN (81.30%) and TDNN (81.53%). 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. Further, in comparison to previous DNN-based studies that uses SPEC-STFT, proposed studies with SPEC-SFF and SPEC-ZTW outperformed by 24.5% and 22.7% (relatively).

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

Rashmi Kethireddy,[1, a] Sudarsana Reddy Kadiri,[2] and Suryakanth V. Gangashetty[3]

[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 Green Fields, Vaddeswaram - 522502, Andhra Pradesh, India.*

(Dated: 7 July 2021)

The goal of this study is to investigate advanced signal processing approaches (SFF and ZTW) with modern DNNs (CNN, TCN, and 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 between dialects, four feature representations are derived from SFF and ZTW methods: SFF/ZTW spectrogram (SPEC−SFF/SPEC−ZTW), SFF/ZTW cepstral coefficients (SFFCC/ZTWCC), mel filter−bank energies from SFF/ZTW (MFBE−SFF/MFBE−ZTW), and MFCCs from SFF/ZTW (MFCC−SFF/MFCC−ZTW). Experiments with and without data−augmentation using CNN classifier revealed that the proposed features performed better than baseline STFT−based features. Even without data−augmentation, all the proposed features gave an improvement of around 15-20% (relative) over best baseline (SPEC−STFT) feature. TCN and TDNN classifiers that captures wider temporal−context further improved the performance for many of the proposed and baseline features. The best performance is achieved with SFFCCs for both TCN (81.30%) and TDNN (81.53%). 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. Further, in comparison to previous DNN−based studies that uses SPEC−STFT, proposed studies with SPEC−SFF and SPEC−ZTW outperformed by 24.5% and 22.7% (relatively).

[a] rashmi.kethireddy@research.iiit.ac.in

I. INTRODUCTION

Identifying the regional origin of a speaker from the acoustic characteristics of speech refers to dialect identification. The task of dialect identification is usually considered as a sub−class of language identification. However, dialect discrimination is a bit more challenging than language identification due to low variability between dialects.

Dialect information in speech reflects 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 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, 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 a low−dimensional and uncorrelated utterance level representations, machine learning approaches such as 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 (FH-VAE) model (Shon *et al.*, 2018b) were investigated.

Further for better classification, linear classifiers such as support vector machine (SVM) and linear discriminant analysis (LDA), and non−linear classifiers such as feed−forward neural networks (FFNNs) (DeMarco and Cox, 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, QDA, and iterative QDA) for classification of British English dialects. Out of them, iterative LDA classifier 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 amount of data for training. To over this, different data−augmentation approaches are investigated in this study. Different weight initialization of 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 Melspectrogram as input) performed better by 10% absolute in 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.

From the early studies on accent classification (Kat and Fung, 1999; Levent 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.*, 2020), it was found that the distribution of learnt frequency bands are different from standard mel−scale distribution. It was observed that learnt scale shown 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. 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) methods. These methods were shown to provide higher spectro−temporal resolution compared to STFT (Aneeja and Yegnanarayana, 2015; Yegnanarayana and Dhananjaya, 2013). 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 it, mel filter−bank energies derived from SFF (MFBE−SFF) were investigated with SVM classifier in our previous studies (Kethireddy *et al.*, 2020), which showed promising results in identifying dialects compared to conventional STFT representations such as mel−spectrogram and MFCCs. In extension to the preliminary studies (Kethireddy *et al.*, 2020), this study proposes to derive four different feature representations: namely, (1) SFF spectrogram (referred as SPEC−SFF), (2) single frequency filtered cepstral coefficients (referred as SFFCCs), (3) mel filter−bank energies derived from SFF spectrum (referred as MFBE−SFF), and (4) mel−frequency cepstral coefficients derived from SFF spectrum (referred as 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.*, 2020) 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 (referred as SPEC−ZTW), (2) zero−time windowed cepstral coefficients (referred as ZTWCCs), (3) mel filter−bank energies derived from ZTW spectrum (referred as MFBE−ZTW), and (4) mel−frequency cepstral coefficients derived from ZTW spectrum (referred as MFCC−ZTW). These four feature representations derived from each method are used as input to advanced deep neural classifiers for dialect classification. To assist related work, we have made code available at: `https://github.com/r39ashmi/e2e_dialect`.

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 (TCNs and TDNNs) that provide long temporal context, along with traditional CNNs 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 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: Section II describes the SFF and ZTW methods along with the proposed feature representations derived from 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 Section IV. Results of the experiments with analysis are provided in Section V. Finally, Section 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 of the selected frequency. In this method, the amplitude envelope at 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 $\hat{\omega}_k = \pi - \frac{2\pi f_k}{f_s}$, 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: $H(z) = \frac{1}{1+rz^{-1}}$. 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:

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

where $K = \frac{(f_s/2)}{\Delta f}$. 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 particular time instant. However in this study, 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.

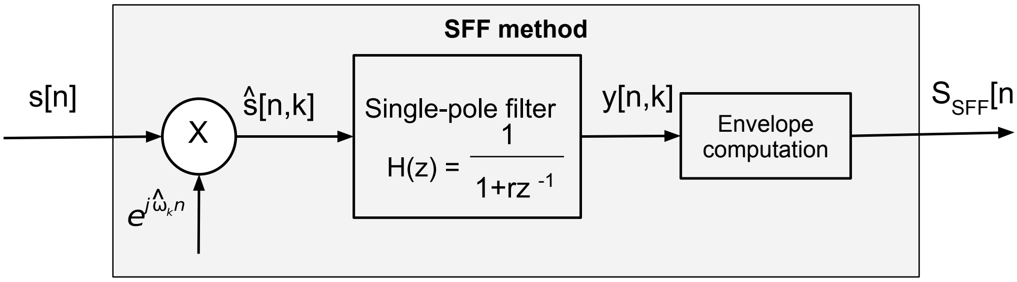


FIG. 1. Schematic block diagram describing the steps involved in the computation of SFF spectrum.

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 speech production mechanism can be captured. In this method, speech signal is windowed with a heavily decaying window (unlike 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:

- 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

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

N is the number of points used in the computation of discrete Fourier transform (DFT) ($N >> M$). 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

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

where $X_R[k]$ is the real and $X_I[k]$ is 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, 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.

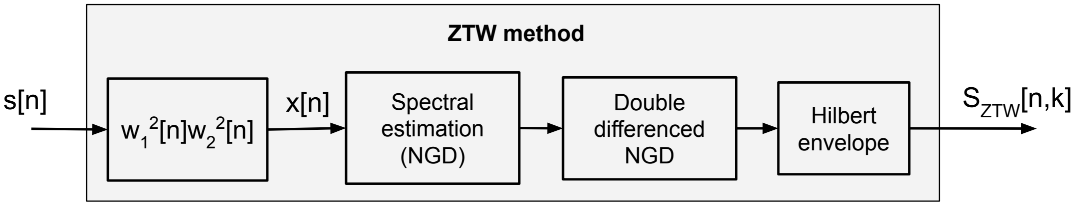


FIG. 2. Schematic block diagram describing the steps involved in the computation of ZTW spectrum.

C. Extraction of feature representations from SFF/ZTW methods

This study propose to derive four types of features from both SFF and ZTW spectra. They are: (1) SFF/ZTW spectrogram (referred as SPEC−SFF/SPEC−ZTW), (2) cepstral coefficients derived from SFF/ZTW spectrum (referred as SFFCC/ZTWCC), (3) mel filter−bank energies derived from SFF/ZTW spectrum (referred as MFBE−SFF/MFBE−ZTW), and (4) mel−frequency cepstral coefficients derived from SFF/ZTW spectrum (referred as MFCC−SFF/MFCC−ZTW). Out of four features derived from SFF spectrum, only MFBE−SFF was investigated for dialect identification in Kethireddy *et al.* (2020) and out of four feature derived from ZTW spectrum, only ZTWCC was investigated for dialect classification in Kethireddy *et al.* (2020). As per our knowledge, this is the first attempt to propose to use these feature representations for dialect classification.

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 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,b):

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

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. 3(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. 3(b). The MFBE extraction involves the computa-

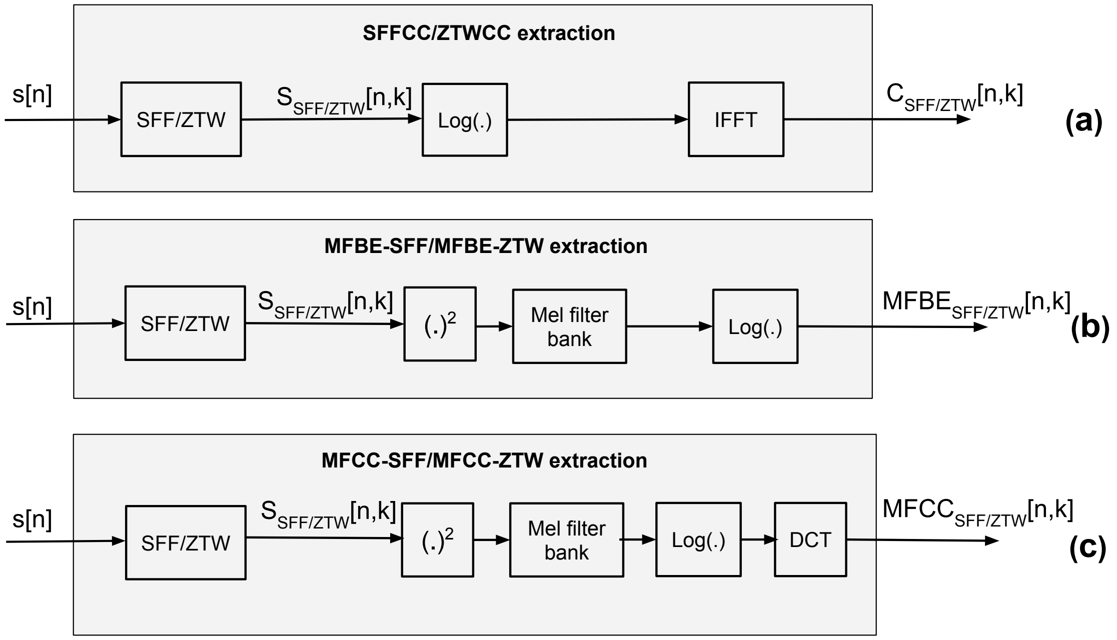


FIG. 3. Schematic block diagrams describing the steps involved in the extraction of features from SFF/ZTW method. (a) Steps involved in the extraction of SFFCC/ZTWCC. (b) Steps involved in the extraction of MFBE−SFF/MFBE−ZTW. (c) Steps involved in the extraction of MFCC−SFF/MFCC−ZTW.

tion 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:

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

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 ZTW/SFF 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. 3(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):

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

where $MFCC_{SFF/ZTW}[n,k]$ denotes the mel-cepstrum. The resulting cepstral coefficients are referred 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 4 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), and time−delay neural network (TDNN) are explored. Deep neural classifiers are trained with frame−level features from an entire utterance. This section gives the details of network architectures of CNN, TCN, and TDNN.

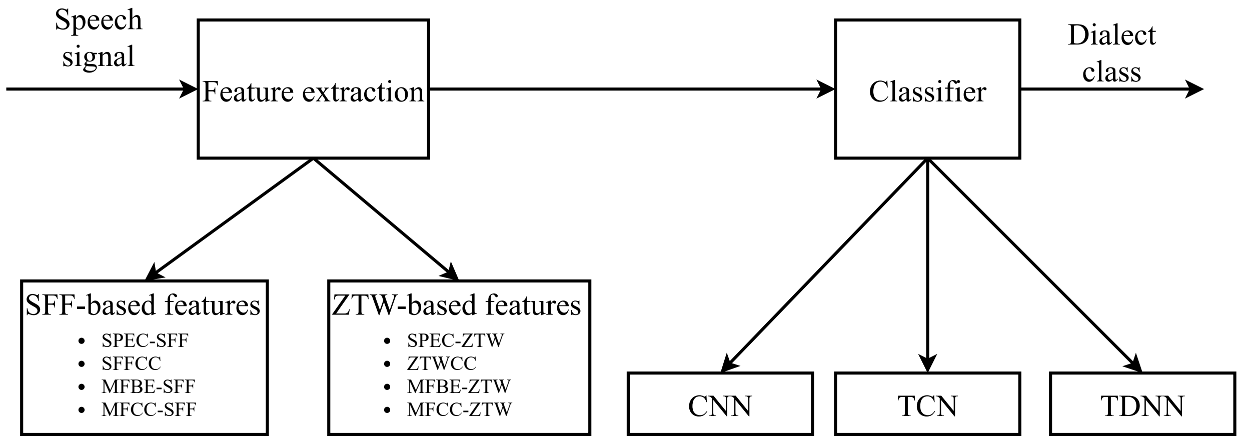


FIG. 4. A schematic block diagram of the proposed dialect classification system with proposed feature extraction methods and deep neural classifiers.

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 1D convolutions (Shon *et al.*, 2018a) and 2D convolutions (Wu *et al.*, 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. 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 represents the layers of the CNN with configurations defined along rows. Convolution layers and max−pooling layers are segmental layers, and the layers after L2 pool processes on utterance level representations. Rectified linear unit (ReLU) activation is commonly applied in all the layers.

TABLE I. End-to-end CNN architecture for dialect classification.

Layers:	Conv1	Conv2	Max pool	Conv3	Conv4	L2 pool	FC1	FC2	FC3
No. filters/output dim.	500	500	-	3000	3000	3000	1500	600	3
Kernel size	5	3	10	5	3	-	-	-	-
Stride	1	1	10	1	1	-	-	-	-

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 is well known and previously used for speech recognition (Seki *et al.*, 2017), spoofing detection (Yu *et al.*, 2017), and accent classification (Kethireddy *et al.*, 2020). As per our knowledge, this is the first attempt to propose to use learnt spectral scale filters for dialect classification. Figure 5 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.

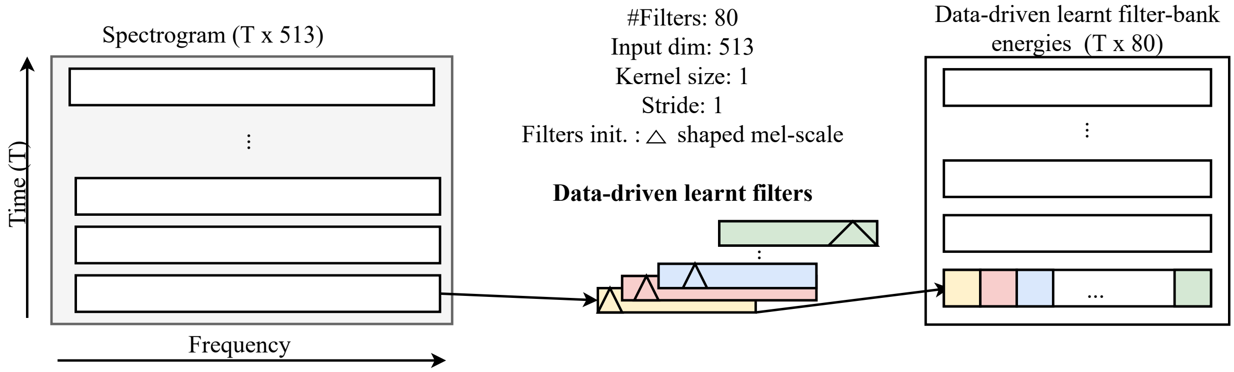

Spectrogram (T x 513)
Time (T)
Frequency
#Filters: 80
Input dim: 513
Kernel size: 1
Stride: 1
Filters init. : △ shaped mel-scale
Data-driven learnt filters
Data-driven learnt filter-bank energies (T x 80)
...

FIG. 5. A schematic block diagram showing learnable spectral filters as convolution layer initialized with mel−scaled triangular−shaped filters.

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 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.

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 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 hyperparameters that define the TConv layer are number of filters (#filters), kernel size, stride, and dilation. The layers after L2 pool processes the dependencies across entire utterance.

TABLE II. End-to-end TCN architecture for dialect classification.

Layers:	TConv1	TConv2	Max1	TConv3	TConv4	L2 pool	FC1	FC2	FC3
No. filters/Output dim.	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	-	-	-	-

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 doesn't loose 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 fully connected (FC) layers as in CNNs. The hyper-parameters that define TD layer are input dimension, output dimension, and context. Along with them cumulative context of the layer is also defined

TABLE III. End-to-end TDNN architecture for dialect classification.

Layers:	TD1	TD2	TD3	TD4	TD5	L2 pool	FC1	FC2	FC3
Input dim.	(feat. dim.)*5	1536	1536	512	512	1500T	1500	1500	600
Output dim.	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

in the table as total context. The first five TD layers process acoustic dependencies at segmental level, while the layers after L2 pooling processes the utterance level dependencies. The TD layers of TDNN used in this study is similar to the architecture defined in Snyder *et al.* (2018) for speaker embeddings.

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 (referred as SPEC−STFT), (2) mel filter−bank energies derived from STFT spectrum (referred as MFBE−STFT), and (3) mel−frequency cepstral coefficients derived from STFT (referred as 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 as MFCC−STFT.

In this study, speech signal is segmented with Hamming window of length 25 msec 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 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 length 25 msec 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 are 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. Number of training epochs are decided approximately based on the loss convergence and over fitting. CNN and TCN models are trained for 50 epochs, while TDNN is trained for 70 epochs. Models are trained to reduce cross−entropy loss with gradient descent optimizer with a learning rate of 0.001. 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 & 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 class label that takes values between 0 to C, n_y is class strength for class y, $\beta = \frac{N-1}{N}$, and N is 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 was collected from adults with 127 male and 104 female speakers. Variations in pronunciation, vocabulary, and grammar that are common to group of people are considered as dialect. 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. Number of utterances available for training in each of the dialect before data−augmentation are, AU:449, UK:246, and US:406. Data is 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 factor resulted in thrice the size of the corpus. Perturbations are implemented using SoX audio manipulation tool (SoX). The sampling frequency of the corpus is 8 kHz.

TABLE IV. Distribution of #utterances in each dialect class of UT−Podcast (AU: Australian English, UK: Britain English, and US: American English) before data−augmentation and after data−augmentation for train data, and test data utterances.

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

V. RESULTS AND DISCUSSION

This section report the dialect classification experimental results and analysis of them. First, the effect of data−augmentation (speed and volume perturbations) to increase the training space for CNN classifier is investigated in Section V A. Secondly, the baseline feature representations (derived from STFT spectrum) and proposed feature representations (derived from SFF and ZTW spectra) are investigated for dialect classification with three deep neural classifiers (CNN, TCN, and TDNN) in Section V B. Further, to better understand the performance of dialect classification systems with respect to each class, class−wise accuracies are also discussed in Section V B. Thirdly, 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 Section V C. Finally in Section V D, the proposed feature representations with deep neural classifiers are compared to the previous approaches in the literature that uses the deep neural classifiers. Unweighted average recall (UAR) is used as primary metric to evaluate the imbalanced classes better, as present in the UT−Podcast corpus. For all the experiments, networks are trained six times to mitigate the side−effect of neural network weights initialization, and tested against each trained model. The performance is averaged across all models, and mean & standard deviation of UAR [%] are reported for all the experiments.

A. Effect of data−augmentation

DNN architectures are constrained to have sufficiently large amount 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, third column (NP: no perturbation) reports the results without any data−augmentation, fourth column (SP) and fifth column (VP) report the results with speed perturbation and volume perturbation respectively, and final column (SVP) reports the results with combination of speed and volume perturbations. Experiments were 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.

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 combination of both speed & volume perturbations (SVP).

	Feat. type	**NP**	**SP**	**VP**	**SVP**
STFT−based features (baseline features)	**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 features)	**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 features)	**SPEC−ZTW**	74.31±1.65	73.50±0.80	78.60±1.56	75.87±0.24
	ZTWCC	72.72±0.582	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

The mean and standard deviation of UAR [%] from six trails are reported in the table. From the standard deviation values, it be can 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 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 combination of both speed and volume perturbations (SVP) gave better performance for all the feature representations (baseline and proposed). Hence through out 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 three deep neural classifiers (CNN, TCN, and 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. Illustrations of latent features derived from three deep neural classifiers (trained with best performing feature (SFFCC)) are shown Fig. 6.

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, from the standard deviation values it can be observed 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 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 best baseline feature MFCC−STFT (76.70% UAR).

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

	STFT−based features (Baseline)			**SFF−based Features (Proposed)**				**ZTW−based Features (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

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 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 best performance (with 81.30 UAR %). Among the ZTW−based features, SPEC−ZTW gave best performance (with 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 %).

In summary, the proposed SFF and ZTW−based features gave better performance over baseline STFT−based features for all the three deep neural classifiers. This supports our hypothesis that the high spectral resolutions of SFF and ZTW spectra help in improving dialect classification and could be an alternative feature representations for dialect discrimination. Among the three deep neural classifiers, TCN and TDNN gave better performance over CNN for many of the baseline and proposed features. This supports our hypothesis that the wider temporal context of TDNN and TCN helped in improving dialect classification. Overall SFFCCs with TDNN gave best dialect classification with UAR of 81.53 %.

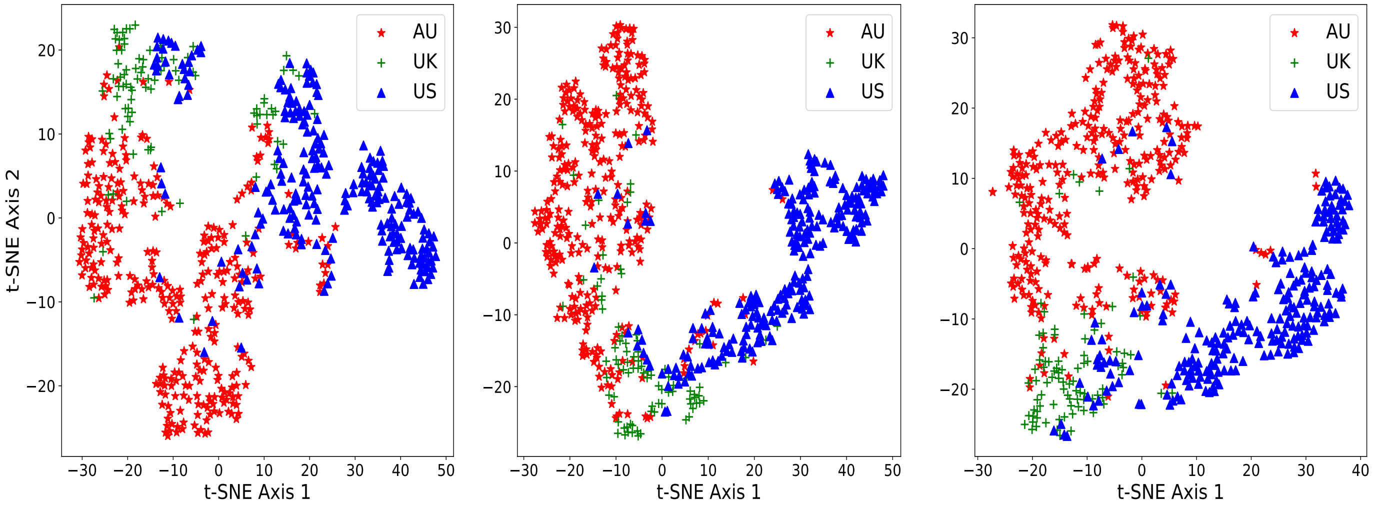
Table VII gives the class−wise accuracies of baseline and proposed features with three deep neural classifiers. From the table, it can be clearly observed that baseline results are biased towards the majority classes (AU and US) with lower performance for minority class (UK dialect). On the other hand, the many of the proposed features (especially SFF-

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

based features) are less biased to the majority classes, and gave better performance for minority class (UK) compared to the baseline features. 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 features. In case of TCN and TDNN classifiers, SFFCC features are more accurate in classification of minority class compared to all other features.

Figure 6 shows the non−linear t−distributed stochastic neighbor embedding (t−SNE) projections of the utterance level feature representations derived from second fully connected layer of CNN (Fig. 6(a)), TCN (Fig. 6(b)), and TDNN (Fig. 6(c)). Latent features derived from SFFCCs are analyzed as they are the best performing features (see Table VI). From t−SNE projections of the latent representations of CNN shown in Fig. 6(a), it can be



(a) CNN

(b) TCN

(c) TDNN

FIG. 6. (color online) Plots showing t−SNE projections of the latent representations from second fully connected layer (FC2, see Section III) of CNN (a), TCN (b), and TDNN (c) for SFFCC features. Projections are color coded by their dialect class (AU:Red(∗), UK:Green(+), and US:Blue(∆)).

observed that the projections of classes AU and US are linearly separated, and the projections of UK class are overlapped with AU and US. Whereas from t−SNE projections of the latent representations of TCN and TDNN shown in Figs. 6(b) and (c), all the classes are relatively better separated when compared to Fig. 6(a). 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 dis-

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	
Feat. 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**

cussed in Section 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 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 minority class (UK) are re−sampled for training. They investigated five different neural architectures (feed−forward neural network (FFNN), five−layer CNN, AlexNet, VGG-11, and ResNet-18) 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 family of CNNs with varied number of convolution layers.

For a fair comparison, 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 the five different DNNs from previous studies (Wu *et al.*, 2018), it can be observed that AlexNet performed better (with 64.90% UAR) than other classifiers. On the other hand, it can be observed that current studies with all the baseline and proposed features outperformed 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,

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 Feat. 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
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

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.

Further comparing the class−wise accuracies among previous studies, it can be observed that other than AlexNet all the classifiers identified UK dialect with less than 50%. However, AlexNet lacked its performance in identifying AU dialect. On the other hand, all the proposed features identified 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 architecture of previous studies with similar data configurations.

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 and 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 proposed SFF−based features such as SPEC−SFF, SFFCC, and MFCC−SFF improved their performance relatively by 3.8%, 5.4%, and 3.7%, and proposed 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 of proposed features improved relatively by 5.7%, 4.8%, 3.8%, and 5.0% respectively, compared to the results obtained with CNN classifier.

Overall, the proposed SFF and ZTW−based features gave better performance over baseline STFT−based features for all the three deep neural classifiers, which supports our hypothesis that the high spectro−temporal resolution of SFF and and ZTW spectra help in improving dialect classification. Between SFF and ZTW−based features, SFF− based features performed better than ZTW−based features. It was also noticed that among the three deep neural classifiers, TCN and TDNN performed better than CNN in many cases. The best dialect classification performance was achieved using SFFCC features with TDNN classifier (81.53% 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 comparison to previous deep neural classifiers (Wu *et al.*, 2018) with STFT−spectrogram as input, current studies with SPEC−STFT, SPEC−SFF, and SPEC−ZTW outperformed by UAR of 14.1%, 24.5%, and 22.7% (relative) respectively. As the proposed features (especially SFF−based features) performed 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

The first author would like to thank the University Grants Commission India (Project No. 3582/(NET-NOV2017)) for supporting her PhD. The second author would like to thank the Academy of Finland (Project No. 330139) for supporting his stay in Finland as a Postdoctoral Researcher.

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 *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, pp. 4277–4280.

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

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, and Language Processing **24**(1), 29–41.

Bougrine, S., Cherroun, H., and Ziadi, D. (**2018**). “Prosody-based spoken Algerian Arabic dialect identification,” Procedia Computer Science **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 *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 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 *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 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, and Language Processing **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,” Computer Speech & Language **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 *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 9260–9269.

DeMarco, A., and Cox, S. J. (**2012**). “Iterative classification of regional British accents in i-vector space,” in *Proc. Symposium on Machine Learning in Speech and Language Processing*, pp. 1–4.

Dhananjaya, N. (**2011**). "Signal processing for excitation-based analysis of acoustic events in speech," Ph.D. thesis, Dept. of Computer Science and Engineering, IIT Madras, Chennai, speech.iiit.ac.in/svlpubs/phdthesis/dhanu-phd-2011.pdf.

Dhananjaya, N., Yegnanarayana, B., and Bhaskararao, P. (**2012**). "Acoustic analysis of trill sounds," The Journal of the Acoustical Society of America **131**(4), 3141–3152.

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

He, K., Zhang, X., Ren, S., and Sun, J. (**2016**). "Deep residual learning for image recognition," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Johnson, R., and Zhang, T. (**2017**). "Deep pyramid convolutional neural networks for text categorization," in *Proc. Association for Computational Linguistics*, 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," The Journal of the Acoustical Society of America **146**(5), EL418–EL423.

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

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

Kadiri, S. R., and Yegnanarayana, B. (**2018**b). "Breathy to tense voice discrimination using zero-time windowing cepstral coefficients (ZTWCCs)," in *Proc. INTERSPEECH*, pp. 232–236.

Kat, L. W., and Fung, P. (**1999**). "Fast accent identification and accented speech recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 221–224.

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

Kethireddy, R., Kadiri, S. R., and Gangashetty, S. V. (**2020**). "Learning filterbanks from raw waveform for accent classification," in *Proc. Int. Joint Conf. Neural Networks*, pp. 1–6.

Kethireddy, R., Kadiri, S. R., Kesiraju, S., and Gangashetty, S. V. (**2020**). "Zero-time windowing cepstral coefficients for dialect classification," in *Proc. ODYSSEY*, pp. 32–38.

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. (**2015**). "Audio augmentation for speech recognition," in *Proc. Interspeech*, pp. 3586–3589.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (**2012**). "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems **25**, 1097–1105.

Levent, A., and Hansen, J. H. (**1997**). "A study of temporal features and frequency characteristics in American English foreign accent," The Journal of the Acoustical Society of America **102**(1), 28–40.

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 networks **8**(7-8), 1201–1214.

Najafian, M., Khurana, S., Shan, S., Ali, A., and Glass, J. (**2018**). "Exploiting convolutional neural networks for phonotactic based dialect identification," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 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 *Proc. Interspeech*, pp. 3023–3027.

Pandey, A., and Wang, D. (**2019**). "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 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 *Proc. Interspeech*, pp. 2155–2159.

Peddinti, V., Chen, G., Manohar, V., Ko, T., Povey, D., and Khudanpur, S. (**2015**a). "JHU ASpIRE system: Robust LVCSR with TDNNs, ivector adaptation and RNN-LMS," in *Proc. Automatic Speech Recognition and Understanding Workshop*, pp. 539–546.

Peddinti, V., Povey, D., and Khudanpur, S. (**2015**b). "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 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 *Proc. Conference on Computer and Communications (ICCC)*, 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 *Proc. Interspeech*, pp. 2383–2387.

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

Seki, H., Yamamoto, K., and Nakagawa, S. (**2017**). "A deep neural network integrated with filterbank learning for speech recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, pp. 5480–5484.

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

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

Siddhant, A., Jyothi, P., and Ganapathy, S. (**2017**). "Leveraging native language speech for accent identification using deep siamese networks," in *Proc. Automatic Speech Recognition and Understanding Workshop*, pp. 621–628.

Simonyan, K., and Zisserman, A. (**2015**). "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations.*

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (**2018**). "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, pp. 5329–5333.

SoX. “Audio manipulation tool” http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/, [Online] Available.

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

Wu, Y., Mao, H., and Yi, Z. (**2018**). “Audio classification using attention-augmented convolutional neural network,” Knowledge-Based Systems **161**, 90–100.

Yegnanarayana, B., and Dhananjaya, N. (**2013**). “Spectro-temporal analysis of speech signals using zero-time windowing and group delay function,” Speech Communication **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.