The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages
29-31 August 2018, Gurugram, India

Improved Language Identification Using Stacked SDC Features and ResidualNeural Network

Ravi Kumar Vuddagiri, Hari Krishna Vydana, Anil Kumar Vuppala

Speech Processing Laboratory, LTRC, KCIS
International Institute of Information Technology, Hyderabad, India.
ravikumar.v@research.iiit.ac.in,hari.vydana@research.iiit.ac.in,anil.vuppala@iiit.ac.in

Abstract

Language identification (LID) systems, which can model high-level information such as phonotactics have exhibited superior performance. State-of-the-art models use sequential models to capture the high-level information, but these models are sensitive to the length of the utterance and do not equally generalize over variable length utterances. To effectively capture this information, a feature that can model the long-term temporal context is required. This study aims to capture the long-term temporal context by appending successive shifted delta cepstral (SDC) features. Deep neural networks have been explored for developing LID systems. Experiments have been performed using AP17-OLR database. LID systems developed by stacking SDC features have shown significant improvement compared to the system trained with SDC features. The proposed feature with residual connections in the feed-forward networks reduced the equal error rate from 21.04, 18.02, 16.45 to 14.42, 11.14 and 10.11 on the 1-second, 3-seconds and $>$ 3-second test utterances respectively.

Index Terms: language identification system, deep neural network, residual networks

1. Introduction

Growing interest in multilingual dialog systems has brought a lot of scientific attention towards the development of language identification systems (LID). Language identification system refers to a module that can tag the input speech with its language identity [1]. LID system has multiple applications in multilingual dialog systems and information querying systems. Human-computer interaction through speech would be more effective if the communication takes place in multiple languages, and such systems demand a front-end LID system to switch and operate between multiple languages [2, 3, 4].

LID systems can be broadly categorized into two types:

- Explicit LID systems
- Implicit LID systems

Explicit systems convert the acoustic sequences to an intermediate representation such as phones, Senones or tokens and temporal relations among these sequences are exploited for developing LID systems. Implicit approaches directly use the acoustic level information to predict the language identity (language ID) [4]. The LID systems using high-level information such as phonotactics, phone frequency, syntax are highly reliable and robust [5, 6]. Parallel phone recognizers followed by language model (PPRLM) systems have been demonstrated to work as robust LID systems, but these systems need multiple acoustic models to be operated in parallel [7, 2]. An independent phone recognizer followed by language model has been explored for developing an LID system [7, 2]. A language independent acoustic model is employed to convert the acoustic sequence to token sequence, and language models such as SRI language model (SRILM) and recurrent neural network language model (RNNLM) have been explored to model the temporal relations among the tokens for developing large-scale LID systems [5]. An acoustic model is used to convert the acoustic sequence to phones, and the bottleneck representation taken from the acoustic model is used to train a recurrent neural network (RNN) for developing LID system [8]. Though these models have performed better, the performance is sensitive to the front-end acoustic model, a mismatch between the data environment in which the acoustic model is trained and operated degrades the performance severely as the errors get propagated to higher modules. With the availability of large data and recent advances in neural networks have motivated researchers to focus on developing implicit LID systems.

Earliest attempts for training implicit LID systems are inspired from speaker recognition frameworks. Gaussian mixture models (GMM), Gaussian mixture models with a universal background model (GMM-UBM) trained using spectral features have been explored for developing LID systems [9, 10]. Use of DNNs for developing LID systems with spectral features has improved the performance of LID systems when larger sized datasets are available [11, 12]. A kind of aggregation of acoustic level information to model long-term temporal information can be beneficial in developing LID systems. I-vector, which is known to better model the temporal context has been explored for developing LID systems [13]. I-vectors convert the variable length sequence to fixed dimension continuous representation, which is used as a feature for training GMM based LID system. Performances of LID systems depends on the length of the test utterances i.e., i-vectors have been modified for developing LID system which can operate on short duration utterances. Speaker and language ID system have been proposed in [11]. Features from multilingual bottleneck have been explored for developing LID systems [14, 15]. Recent development in neural networks has influenced the performance of LID systems, and have enriched the capabilities of neural networks to process the whole utterance. Such networks have been employed for training LID systems. RNNs and convolutional neural networks (CNNs) have been explored for developing LID systems [16, 17]. Though sequential models like RNN, long short-term memory (LSTM) have performance better, they are not parallelizable. Recently a feed-forward architecture has been proposed in [18, 19], where a self-attention mechanism is used to convert the variable length sequence to a fixed dimension vector and the fixed dimension representation is used to discriminate the languages. The whole network can be trained as a single-framework through back-propagation. Though the LID system modeling the entire utterance has performed bet-

10.21437/SLTU.2018-44

ter they are sensitive to the utterance length, performance of there systems degrades when the utterance length is varying. The performance of the LID system can also influence varying background and mobile environments are explored in [20, 21].

LID systems that have efficiently modeled the long-term temporal information has performed better. To capture this information a feature that can model the long-term temporal information is required. Typical LID systems use SDC features to model temporal dynamics of the speech signal. This study explores the use of stacked SDC features for capturing the long-term temporal context. SDC features from neighboring temporal context have been appended and these stacked SDC features are explored for developing LID systems.

In this study, deep neural networks and deep neural networks with residual connections have been explored for developing LID systems. Use of residual connections in a feed-forward architecture has improved the convergence of the model [22, 23]. Using residual connections, neural networks of extended depth can be trained without gradient problems. Though the residual networks were initially thought to represent the feature hierarchy in an efficient way due to their extended depth, but the latter studies have shown that they work as iterative feature re-estimators [24]. The feature representation in a residual network is maintained closer to the input representation. This study uses deep neural networks and residual neural networks for developing LID systems using stacked SDC features.

The remaining paper is organized as follows. The database used during the study has been described in section 2. Section 3 describes the performances DNN and resnet LID systems. Proposed experiments and results are presented in section 4. Conclusion and future scope are presented in section 5.

2. Speech Corpus

In this study, Oriental Language Recognition(OLR) challenge based on AP17-OLR [25, 26] database is used. The database is offered by Speech Ocean and Multilingual Mino-lingual Automatic Speech Recognition (M2ASR), which is supported by the National Natural Science Foundation of China (NSFC) project. This dataset consists of 10 languages spoken in east, northeast and southeast Asia - Russian in Russia, Korean in Korea, Cantonese in China Mainland and Hong Kong, Uyghur, Kazakh and Mandarin in China, Japanese in Japan, Indonesian in Indonesia, and Vietnamese in Vietnam, respectively. The data is recorded in reading style from various environments like quiet, noisy, recording on mobile e.t.c. Each language has at about 10 hours of speech data. These speech samples are collected in different modes, with a sampling rate of 16 kHz with a size of 16 bits. For each language, 1800 utterances are selected as development set and the rest are used as the training set. The AP17-OLR test set focuses on short utterances. In this work, three test conditions depending on the length of test utterances are used i.e., 1 sec, 3 sec, and full-length conditions.

3. Language identification framework used in this study

3.1. Shifted Delta Features:

Mel-frequency cepstral coefficients are widely used features for speaker recognition tasks, however in an LID task, Shifted Delta Cepstrum (SDC) has been found to exhibit superior performance due to their ability to incorporate additional temporal information spanning multiple frames into the feature vector [1, 27, 28, 29]. In this work, spectral vector is obtained by block processing the whole speech using a 20 ms window with an overlap of 10 ms. From every 20 ms speech, MFCC features are computed using 24 filter bands.

Let $x(t)$ be the static feature vector that is composed of feature elements derived from a frame. SDCs are extracted by concatenating the delta-cepstra computed across multiple frames of speech. Four parameters (N, d, P, and, k) specify the computation of SDC's [27]: N is the number of cepstral coefficients computed from each frame, d is the delta distance between acoustic feature vectors, p is the distance between blocks, and k is the total number of successive blocks of delta coefficients used to formulate a final SDC feature vector.

The delta feature vector for each frame is computed as

$$\delta_d(t,i) = c(t+iP+d) - c(t+iP-d) \tag{1}$$

for $i = 0,1,2,...,k-1$ which covers a block of $2d+1$ frames. The shifted-delta feature vector $SDC(t)$ is obtained by concatenating the delta features from k consecutive blocks as shown in Figure. 1, i.e.,

$$SDC(t) = \begin{pmatrix} \delta_d(t,0) \\ \delta_d(t,1) \\ . \\ . \\ . \\ \delta_d(t,(k-1)) \end{pmatrix} \tag{2}$$

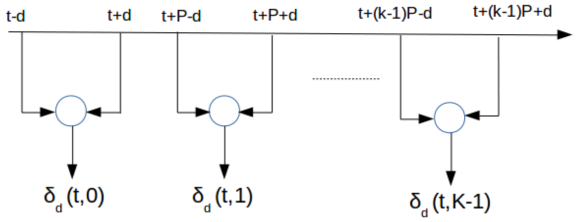

Figure 1: *Computing **Shifted Delta Cepstra (SDC)** feature vector at frame t for parameters $N-d-P-k$.*

This work uses the widely used $7-1-3-7$ configuration for computing shifted delta feature vector. The computed shifted delta feature vector ($\delta_d(t,k-1)$) is appended to static feature vector $x(t)$ to form SDC feature vector, which is of size $(k+1)N$ i.e., 56 Dimensions. The total duration covered per SDC is given by

$$SDC_{Duration} = \frac{F_r * t_f}{2} \tag{3}$$

Where F_r is total number of frames per SDC $k(2d+1)$ (no of blocks * no of frames per block), t_f is each frame duration 20ms with 50% overlap. In [30] recommends a time interval of 90 ms to maintain the transitional information integrated with changes from one phoneme to another. With a time interval from 100 to 160 ms to obtain good estimates an incline of spectral transitions between syllables [31].

3.2. LID systems using Deep Neural Networks

Deep neural networks have been explored for developing LID systems. This network is termed as 9-DNN, consists of nine hidden layers comprising of rectified linear units (ReLU) i.e., (56R-1024R-1024R-1024R-1024R-1024R-1024R-1024R-1024R- 1024R-10S). A softmax output layer with categorical-cross entropy loss function has been used. The network is optimized using ADADELTA optimizer [32]. A successive decrease in the validation accuracy in three successive epochs is considered as an early stopping criterion. Learning rate is halved whenever a minimum increase in validation accuracy is less than 0.5 between successive epochs.

3.3. Residual Neural Network architecture

This work explores the use of residual neural networks (Resnets) for developing LID systems. If a feed-forward deep neural network learns a mapping $H(x)$, with x as input dimension(56) then $H(x) - x$ can also learned by the network with different ease. Then the residual function $F(x)$ can be defined as $F(x) := H(x) - x$. Residual network can be implemented like any deep neural network with a constraint $H(x)$:=$F(x) + x$. The residual block used in this study is presented in the Figure 2.

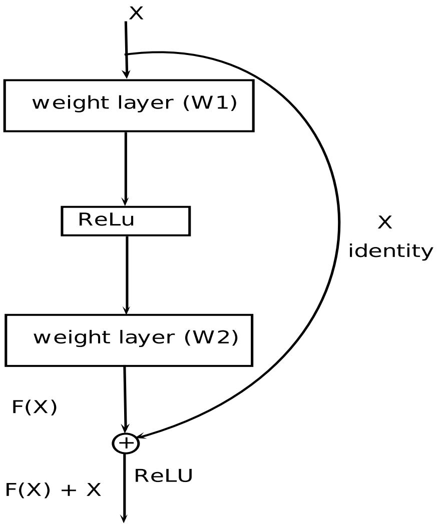

Figure 2: *Residual block used in this study.*

The residual block comprises two weight layers, i.e., W1-ReLU units, W2-ReLU units. During the study, the second weight layer W2 always has the number of units equal to the input dimension(56), so that the output of weight layer W2 can be directly added to input without any zero-padding. The weight layer W1 has 1024 units. A residual network of 9, 4 hidden layers have been used for developing LID system. The Resnet is trained to minimize categorical entropy loss using ADADELTA optimizer and stochastic gradient descent-with-Nesterov momentum as an optimizer. The learning rate and momentum factor are shown in Table. 3, and mini-batch size used during the training is 200. During the work, a continuous decrease in the validation accuracy for three successive epochs is considered as an early stopping criterion. Learning rate is halved upon encountering an epoch whose increase validation accuracy is less than 0.5.

4. Experiments and Results

Performances of various baseline LID systems is presented in Table. 1 [26]. The performance of LID system using i-vectors has been presented in row 1 of Table. 1. The linear discriminant analysis is carried out on the obtained i-vectors and the results are presented in row 2 of Table. 1. Time-delay neural networks can model the temporal context in the input features. Time-delay networks have been explored for developing LID systems and the EER obtained using time delay neural network is presented in row 3 of Table. 1. Long short-term memory networks (LSTM) have been explored for capturing the sequential relations in the input acoustic sequences for developing an LID system, the results are reported in row 4 of Table. 1. The performance of LID is evaluated on the test-set using the test utterances of three different durations i.e., 1 sec, 3 sec and full-length.

Table 1: *Performance of baseline LID systems in terms of EER. (The results presented in the table are taken from [26])*

System	1 sec	3 sec	Full-length
i-vector	15.28	7.59	6.224
i-vector+ LDA	13.30	5.95	4.704
TDNN-LID	15.63	15.43	14.65
LSTM-LID	16.77	16.99	16.03

Performances of LID systems developed using DNNs and Resnets using SDC features have been presented in Table. 2. Deep neural networks trained with 4, 6, 9 hidden layers have been explored and the results have been presented in Table. 2. Similarly, the performance of LID systems using residual networks with 4, 9 hidden layers have been presented in Table. 2. From Table. 2, it has been observed that the performance of LID systems developed using DNNs and Resnets is optimal using a depth of 4 hidden layers. Further, in the study DNNs and Resnets of 4 hidden layers have been used for developing LID systems.

Table 2: *Performance of LID systems developed using SDC features.*

System	1 sec	3 sec	Full-length
DNN_{4H}	21.04	18.02	16.45
DNN_{6H}	21.45	18.78	16.78
DNN_{9H}	22.12	19.95	16.91
$Resnet_{4H}$	20.81	17.59	16.13
$Resnet_{9H}$	21.45	18.24	16.56

4.1. LID systems using stacked SDC features

To better model the temporal context, successive SDC features are stacked with a temporal context. In this study, SDC features are stacked with different temporal context and the performances are presented in Table. 3. But by appending the successive feature vectors increases the redundancy in the feature representation which may over-fit the network quite easily. Training the network with stacked-SDC features using SGD-with-Nesterov momentum as an optimizer, a severe over-fitting has been observed. Use of adaptive optimizers have helped the networks to converge to a better solution. As Adadelta is an adaptive optimizer, with a capability to adjust the learning rates based on the progress of training, networks trained with Adadelta optimizer has performed better. Maintaining a high initial learning rate i.e., 0.1 has resulted in a superior performance. Upon encountering a decrease in validation accuracy the learning rate is halved, the training is progressed till the early stopping is reached, and the performances of LID systems trained with SGD and Adadelta are shown in Table. 3.

Table 3: *Performance of LID systems developed using different optimizers such as SGD-Nesterov and ADADELTA.*

DNN-4H					
	Adadelta			SGD-Nesterov	
Learning rates (η)	0.1	0.01	0.001	0.01	0.001
SDC-168	**11.78**	12.21	12.57	14.63	13.43
SDC-392	**11.19**	12.02	12.22	14.61	13.58

From Table. 3, it can be observed that the using SDC features the networks optimized using Adadelta have performed significantly better. A high initial learning rate has helped the networks to converge to a better solution using Adadelta optimizer. SGD-Nesterov with high initial learning rate i.e., 0.1 has been trained, but the model has not converged.

Stacked SDC features with different temporal context have been explored for developing LID systems. In this study, DNNs and Resnets have been explored for developing LID systems and the performances in terms of EER are tabulated in Table. 4. Column 1 of Table. 4 is the configuration of stacked SDC features and column 2 is the dimensionality of the stacked SDC feature. Columns 3-5, 6-8 are the performances of LID systems using DNN and Resnet. It can be observed that with an increase in the temporal context of stacked SDC an increase in EER can be observed. Use of residual connections has helped the models to converge better, it can be noted that the performance of resnet is superior to DNN, and the margin of improvement is higher for utterances of 1 sec duration. From the Table. 4 it can be observed that the LID systems trained with stacked SDC with a temporal context of 9 frames ie., (4-1-4) have performed significantly better than SDC features.

Table 4: *Performance of LID systems developed using stacked SDC features*

SDC	Feature dimension	DNN			Resnet		
		1s	3s	all	1s	3s	all
1	56D	21.04	18.02	16.45	21.45	17.59	16.13
1-1-1	168D	16.90	12.52	11.78	15.72	12.02	11.33
2-1-2	280D	16.71	11.73	11.54	15.55	11.47	10.19
3-1-3	392D	16.43	11.63	11.19	15.16	11.23	11.02
4-1-4	504D	**15.64**	**11.45**	**10.58**	**14.42**	**11.14**	**10.11**
5-1-5	616D	15.93	13.44	12.52	15.39	12.12	11.37

From Table. 4 it can be observed that the performance of the proposed stacked SDC feature has performed better than SDC features, use of the proposed feature has decreased the EER from 21.04, 18.02, 16.45 to 15.64, 11.45, 10.58 on the test utterances of duration 1 sec, 3 sec and >3sec. Use of residual networks for training LID has further reduced the corresponding EERs to 14.42, 11.14, 10.11. Stacked SDC can better model the temporal context and has led to the better performances in terms of EER. The proposed feature has performed better than the state-of-the-art LID systems such as PTN, LSTM-RNN LID, TDNN-LID, i-vector, i-vector-LDA systems for longer utterances i.e, >3 sec. For the utterances of 3 sec duration, proposed approach performs superior to PTN, LSTM-RNN LID, TDNN-LID and comparable to i-vector based approaches. For 1 sec utterances, the proposed approach performs better than LSTM-RNN LID, TDNN-LID, and i-vectors and the performance is inferior to i-vectors-LDA and PTN LID systems.

5. Summary and Conclusions

Conventional LID systems employ SDC features for developing LID system. To capture the long-term temporal information, the feature that can model the temporal context is required. In this study, the long-term temporal information is captured by stacking successive SDC features. From the experimental analysis, it can be observed that the LID systems developed using stacked SDC features have exhibited superior performance. But stacking successive frames would increase redundancy in the feature and can over-fit the networks, to avoid this we have used adaptive optimizers with a high initial learning rate. Further sophisticated approaches to reduce the redundancy of the stacked features could be explored in future, to reduce the over-fitting nature. The performance of the sequential models has to be explored using Stacked SDC features. Performances of LID systems using stacked SDC features in noisy environments have to be explored. Auto-encoders such as denoising and variational auto-encoders have to be explored to reduce the redundancy in stacked SDC feature. LID systems that can perform efficiently on short utterances have to be explored.

6. Acknowledgements

The authors would like to thank Science & Engineering Research Board (SERB) for funding Language Identification in Practical Environments (YSS/2014/000933) project.

7. References

[1] F. Allen, E. Ambikairajah, and J. Epps, “Language identification using warping and the shifted delta cepstrum,” in *Proc. Multimedia Signal Processing, 7th Workshop on*. IEEE, 2005, pp. 1–4.

[2] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, “Language identification: A tutorial,” *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.

[3] Y. K. Muthusamy, E. Barnard, and R. A. Cole, “Reviewing automatic language identification,” *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, 1994.

[4] M. A. Zissman and K. M. Berkling, “Automatic language identification,” *Speech Communication*, vol. 35, no. 1, pp. 115–124, 2001.

[5] B. M. L. Srivastava, H. Vydana, A. K. Vuppala, and M. Shrivastava, “Significance of neural phonotactic models for large-scale spoken language identification,” in *Proc. Int. Joint Conf. Neural Networks*. IEEE, 2017, pp. 2144–2151.

[6] A. McCree and D. Garcia-Romero, “Dnn senone map multinomial i-vectors for phonotactic language recognition,” in *Proc. Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[7] M. A. Zissman, “Comparison of four approaches to automatic language identification of telephone speech,” *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 1, p. 31, 1996.

[8] Z. Tang, D. Wang, Y. Chen, L. Li, and A. Abel, “Phonetic temporal neural model for language identification,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 134–144, 2018.

[9] P. A. Torres-Carrasquillo, D. A. Reynolds, and J. R. Deller, “Language identification using Gaussian mixture model tokenization,” in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, vol. 1. IEEE, 2002, pp. I–757.

[10] E. Wong and S. Sridharan, “Methods to improve Gaussian mixture model based language identification system,” in *Proc. Seventh International Conference on Spoken Language Processing*, 2002.

[11] F. Richardson, D. Reynolds, and N. Dehak, “A unified deep neural network for speaker and language recognition,” *IEEE Sig. Proc.*, 2015.

[12] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, “Automatic language identification using deep neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*. IEEE, 2014, pp. 5337–5341.

[13] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, “Language recognition via i-vectors and dimensionality reduction,” in *Proc. INTERSPEECH*, 2011.

[14] R. Fér, P. Matějka, F. Grézl, O. Plchot, and J. Černockỳ, “Multilingual bottleneck features for language recognition,” in *Proc. Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] P. Matejka, L. Zhang, T. Ng, H. S. Mallidi, O. Glembek, J. Ma, and B. Zhang, “Neural network bottleneck features for language identification,” *in Proc. IEEE Odyssey*, pp. 299–304, 2014.

[16] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, “Application of convolutional neural networks to language identification in noisy conditions,” *in Proc. Odyssey-14, Joensuu, Finland*, 2014.

[17] W. Geng, W. Wang, Y. Zhao, X. Cai, B. Xu *et al.*, “End-to-end language identification using attention-based recurrent neural networks.” in *Proc. INTERSPEECH*, 2016, pp. 2944–2948.

[18] K. Mounika, S. Achanta, H. Lakshmi, S. V. Gangashetty, and A. K. Vuppala, “An investigation of deep neural network architectures for language recognition in Indian languages.” in *Proc. INTERSPEECH*, 2016, pp. 2930–2933.

[19] C. Raffel and D. P. Ellis, “Feed-forward networks with attention can solve some long-term memory problems,” *arXiv preprint arXiv:1512.08756*, 2015.

[20] V. Ravi Kumar, H. K. Vydana, J. V. Bhupathiraju, S. V. Gangashetty, and A. K. Vuppala, “Improved language identification in presence of speech coding,” in *International Conference on Mining Intelligence and Knowledge Exploration*. Springer, 2015, pp. 312–322.

[21] V. Ravi Kumar, H. K. Vydana, and A. K. Vuppala, “Curriculum learning based approach for noise robust language identification using DNN with attention,” *Expert Systems with Applications*, 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2018.06.004

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

[23] H. K. Vydana and A. K. Vuppala, “Residual neural networks for speech recognition,” in *Proc. European Signal Processing Conference*. IEEE, 2017, pp. 543–547.

[24] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,” *arXiv preprint arXiv:1505.00387*, 2015.

[25] D. Wang, L. Li, D. Tang, and Q. Chen, “AP16-OL7: A multilingual database for oriental languages and a language recognition baseline,” in *Proc. Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific*. IEEE, 2016, pp. 1–5.

[26] Z. Tang, D. Wang, Y. Chen, and Q. Chen, “AP17-OLR challenge: Data, plan, and baseline,” *arXiv preprint arXiv:1706.09742*, 2017.

[27] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr, “Approaches to language identification using Gaussian mixture models and shifted delta cepstral features,” in *Proc. Seventh International Conference on Spoken Language Processing*, 2002.

[28] B. Bielefeld, “Language identification using shifted delta cepstrum,” in *Proc. Fourteenth Annual Speech Research Symposium*. IEEE, 1994.

[29] V. Mounika Kamsali, V. Ravi Kumar, S. V. Gangashetty, and V. Anil Kumar, “Combining evidences from excitation source and vocal tract system features for Indian language identification using deep neural networks,” *International Journal of Speech Technology*, pp. 1–8, 2017.

[30] S. Furui, “Cepstral analysis technique for automatic speaker verification,” *IEEE Trans. Acoustics, Speech, and Language Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[31] F. K. Soong and A. E. Rosenberg, “On the use of instantaneous and transitional spectral information in speaker recognition,” *IEEE Trans. Acoustics, Speech, and Language Processing*, vol. 36, no. 6, pp. 871–879, 1988.

[32] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.