Towards Emotion Independent LanguageIdentification System

by

Priyam Jain, Krishna Gurugubelli, Anil Kumar Vuppala

Report No: IIIT/TR/2020/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
January 2020

Towards Emotion Independent Language Identification System

Priyam Jain, Krishna Gurugubelli, and Anil Kumar Vuppala
Speech Processing Laboratory, LTRC, KCIS
International Institute of Information Technology, Hyderabad, India
{priyam.jain, krishna.gurugubelli}@research.iiit.ac.in, and anil.vuppala@iiit.ac.in

***Abstract*—Language Identification (LID) is an integral part of multilingual speech systems. There are various conditions under which the performance of LID systems are sub-optimal, such as short duration, noise, channel variation, and so on. There has been effort to improve performance under these conditions, but the impact of speaker emotion variation on the performance of LID systems has not been studied. It is observed that the performance of LID systems degrade in the presence of emotional mismatch between train and test conditions. To that effect, we investigated adaptation approaches for improving the performance of LID systems by incorporating emotional utterances in form of adaptation dataset. Hence, we studied a prosody modification technique called Flexible Analysis Synthesis Tool (FAST) to vary the emotional characteristics of an utterance in order to improve the performance, but the results were inconsistent and not satisfactory. In this work, we propose a combination of Recurrent Convolutional Neural Network (RCNN) based architecture with multi stage training methodology, which outperformed state-of-art LID systems such as i-vectors, time delay neural network, long short term memory, and deep neural network x-vector.**

I. Introduction

Language Identification (LID) refers to identifying the language from a spoken utterance. In this technology driven world, LID has found many practical applications [1]. Multilingual speech recognition [2] and speech translation [3] are two such examples. The variety in the number of languages spoken in the world coupled with mobility and interaction of population further adds to the necessity for development in this domain [4]. LID systems can play a role as a front-end or pre-processing component of a speech-enabled application or device. As these speech enabled smart devices reach our homes and offices, the development and improvement in this domain gathers increased interest. Many of these advancements can be attributed to the recent novel approaches in the area of machine learning. LID challenges such as Oriental Language Recognition (OLR) [5] and NIST Language Recognition Evaluation (LRE) [6] have also had a huge impact in the advancement of this domain.

It has been shown in the literature that fixed dimensional embeddings such as i-vector and x-vector give good performance for LID and speaker recognition [7]–[10]. Recurrent Neural Network (RNN), such as Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU) and b-LSTM have also been demonstrated to give good results for Automatic Speech Recognition (ASR) and LID [11], [12]. Their superior performance over Deep Neural Network (DNN) can be attributed to their sequence modelling ability. Time Delay Neural Network (TDNN) is an another neural network architecture that has given good results for LID and ASR [13], [14]. Combining these machine learning architectures with attention mechanism has also resulted in performance improvements [15], [16]. Apart from architecture level developments, there has been work on developing new training methodologies to improve performance. Improving performance of LID by using a student teacher network [17] and joint learning by optimizing two or more loss functions simultaneously are evidence of that [18]. This idea of using multiple loss functions in a single network combined with the way a Gaussian Mixture Model (GMM) is initialised using a Universal Background Model (UBM), motivated us to investigate non contemporary methods of initialising and adapting a neural network.

Most of these architectures have been shown to perform well for computer vision and natural language processing related tasks as well. B-LSTM and GRUs have been used for semantic analysis on textual data [19] and image captioning [20]. This goes to show that these architectures are independent of the modality of data, rather depends on the modelling task. This is also evident from the fact that Convolutional Neural Networks (CNN) are heavily prominent in the field of computer vision but not much in NLP or speech [21]. Recently introduced Recurrent Convolutional Neural Network (RCNN) for image scene labelling task [22] have also given good results for emotion and phoneme recognition tasks on spoken data [23]. This motivated us to investigate them for the LID task.

From recent LID literature, it can be observed that the performance of most LID systems suffer when evaluated with short duration spoken utterance [24], presence of noise [25] and different dialects of a language [26]. However we noticed that performance variation of a LID system under different emotional conditions has not been investigated much. Effect of emotional state has been studied for speaker identification [27], ASR [28] as well as for LID [29]. These studies have shown that the performance degradation can be observed under emotional conditions. At feature level, acoustic feature modification and prosody modification are done to improve performance, whereas MAP based adaptations can be done at the model level [30]. Prosody modification is achieved by altering the pitch, duration and energy of an utterance. In this work, we study the performance of LID systems under emotional condition and propose an approach to improve the

performance. To the best of our knowledge, this is the first time in literature that RCNN with multi stage training methodology has been used for LID.

The rest of this paper is organised as follows. Section II describes the RCNN and training methodology used. Section III describes the dataset used and the setup of our experiments with respect to the baseline systems and RCNN. Section IV outlines the results obtained and finally in Section V, we conclude our findings.

II. Approach

A. Recurrent Convolutional Neural Network

The RCNN uses a Recurrent convolutional layer (RCL), which applies convolution operation to the input feature map over discrete time steps [23]. The notion of time step in RCL layers is different from that in the RNNs. In RNNs, the notion of time step relates to the actual notion of time in a sequential input whereas, in a RCL layer, it denotes the number of times convolution operation is applied iteratively.

The activation of a hidden layer $h^{(t)}$ at position (i, j) due to the input $x^{(t)}$ is given by :

$$h^{(t)}(i,j) = \sigma\Big(\sum_{i'=-s}^{s}\sum_{j'=-s}^{s} w_k^f(i',j')x^{(t)}(i-i',j-j') + \sum_{i'=-s}^{s}\sum_{j'=-s}^{s} w_k^r(i',j')h^{(t-1)}(i-i',j-j') + b\Big) \quad (1)$$

where w_k^f, w_k^r and b denote the forward and recurrent convolution kernels and the added bias respectively. $\sigma(x)$ denotes the ReLU activation function.

The weights in a RCL are shared across time steps, which is evident from equation 1 . The number of time steps T in RCL layer is a hyperparameter, which controls the context received by an activated unit. For example, higher values of T will mean a wider context to the activated unit but also with the same number of parameters as the weights are shared across time step. An example of a RCL with T = 3 is shown in Fig. 1

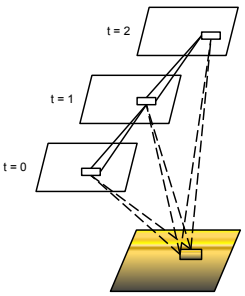

Fig. 1. Hidden states of a single RCL with T = 3. The bottom coloured figure represents input whereas forward and recurrent connections are represented by solid and dash lines respectively.

Input to the RCNN is usually a spectrogram, which has time on x-axis and frequency on y-axis. Hence, there is a dependency along both the axis. The operations applied by the RCL layer leverages on these multidimensional relations. A complete RCNN architecture consists of several RCL layers, with max pooling layers in between, followed by a few fully connected layers. This architectural design is similar to a conventional CNN.

B. Multi Stage Training

The task of identifying language from spoken utterance is a classification task, frequently learnt by optimizing over cross entropy loss function by using language labels as ground truth. Cross entropy is calculated individually for a sample, and hence does not take into account the overall representation of a class. On the other hand triplet loss function, shown in equation 2, forms a triplet with respect to the input sample by taking a positive and a negative sample. It's objective is to minimise the distance between input and positive sample, while maximising the same between input and negative sample. So it is often employed in representation learning tasks [31].

$$L(A,P,N) = max(||f(A) - f(P)||^d - ||f(A) - f(N)||^d + \alpha, 0) \quad (2)$$

In equation 2, $f(A)$, $f(P)$ and $f(N)$ represent the feature vectors of the input, positive and negative samples, $||.||^d$ denotes the distance metric.

A network trained to classify languages can suffer from variations if they are not accounted for during training. Emotion is identified as one such variation which affects the performance of a LID system. It can be understood that degradation is observed due to emotion, causing the representation of utterances to deviate from their representation under neutral emotion, hence deviating from the sample space on which model was trained. The RCL layers can be thought of as feature extractors operating on the raw acoustic features. Hence we can use the triplet loss function over these representations to account for the variability due to emotion. Therefore we first train our network to learn emotion invariant representations, then add the final dense layer with softmax activation for classifying languages. This approach differs slightly from transfer learning as we are training the network on same task but in two stages.

III. Experimental Setup

A. Dataset

Basic details of the dataset used in this work are described in Table I. The complete data used in this work has been pooled from various datasets of the 7 target languages : Basque [32], English [33], [34], German [35], [36], Hindi [34], [37], Serbian (International Standard Language Resource Number: 462-780-920-598-3), Spanish (International Standard Language Resource Number: 477-238-467-792-9) and Telugu [34], [38]. The emotions considered are : Anger, Happy, Neutral and Sad. Train dataset only consists of neutral utterances, while test dataset corresponding to each emotion is created. We have also created a separate adaptation dataset consisting of emotional utterances from all 7 languages. All the speakers in train and adaptation are different from test datasets and so is the spoken content while recording environment and channel characteristics are same among all emotions of an utterance.

TABLE I
DETAILS OF THE DATASET USED IN THIS WORK

	Basque	English	Hindi	German	Serbian	Spanish	Telugu	Total Duration	Total Utterances
Train	1014	960	2380	1208	292	780	1520	12.9	8154
Adaptation	1681	175	1455	1180	291	836	2100	8.6	7718
Test Neutral	140	50	150	90	36	85	150	0.71	701
Test Angry	140	50	150	110	24	76	150	0.66	700
Test Happy	140	50	150	70	24	84	150	0.7	668
Test Sad	140	50	150	70	24	79	150	0.78	663

B. Baseline Systems

The performance of LID systems in the presence of a emotional mismatch has to be studied. We have used the recent as well traditional state-of-art LID models described in the literature as baselines to compare with our approach.

1) i-vector with PLDA: To extract i-vectors, 39D Mel frequency cepstral coefficients (MFCC) are given as input to the UBM-GMM model, which are trained using 1800 and 3600 utterances, respectively. Dimensionality of the i-vectors is set to be 400. Probabilistic linear discriminant analysis (PLDA) is applied to the extracted i-vectors. This is followed by cosine scoring.

2) TDNN and LSTM: Both the TDNN and LSTM are provided with the 40D fbank as inputs. TDNN has 6 hidden layers, with 650 neuron activations in each. The temporal context of the layers are [-2,2], [-1,1], [-1,1] and {-6,-3,0}, where second and last layer have no splicing. In the LSTM network, the first layer is kept similar to that of the TDNN. This is followed by a LSTM layer with 512 cells. Both the TDNN and LSTM have a final layer with dimension equal to the number of classes.

3) DNN x-vector: X-vectors are extracted using a network similar to the one described in [10]. Input is a 40D fbank feature vector and each layer operates on a narrow temporal context coming from the previous layer. The temporal context gets wider as the layer gets deeper and eventually the stat pooling layer gets the complete temporal context. The layer temporal context of the first 5 layers are [-2,2], {-2,1,2}, {-3,1,3}, {1} and {1}. These are then followed by a stat pooling layer, which calculates the mean and standard deviation along the temporal axis, which is then fed to subsequent fully connected layers. At the output there is a softmax activation to perform the classification task.

C. Prosody Modification

We have used Flexible Analysis Synthesis Tool (FAST) [39] to synthesise prosody modified utterances from the adaptation set to be used during training. FAST is based on LP analysis and epoch extraction to get the vocal tract and excitation source characteristics, followed by modification of epoch locations and pitch contour according to the target emotion epoch locations and pitch periods. Prosody modified speech is then synthesized using the LP residual of the source emotion.

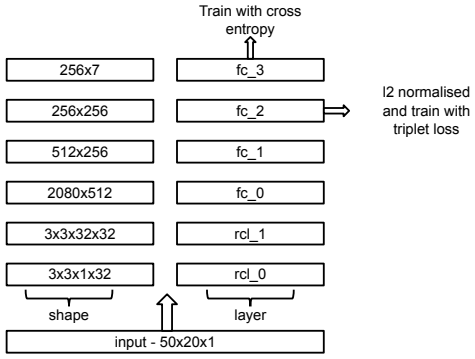

Fig. 2. RCNN architecture used in this work

D. RCNN

We have extracted 20D MFCC from 25ms frames with 10 ms shift. A sample input to the RCNN is of dimension 50 x 20, obtained by stacking 50 frames. We consider the output softmax activation as segment level class probabilities for a segment of 50 frames and average that of all segments of an utterance to get utterance level class probabilities and in turn utterance level predictions.

We empirically found that using convolution filters with small kernel size, such as 3, gave better results. Hence for RCL, we use kernels of size 3 and stride 1 with 32 filters. We use the same kernel size for pooling layers but with stride 2. Also we keep the number of time steps in every RCL as 3. Each RCL is followed by a max pooling layer. The output of last RCL is flattened and fed into the fully connected (FC) layers. Each FC layer is followed by a dropout layer with training drop probability of 0.7. The complete architecture is shown in Fig. 2, the output of fc_2 layer is l2 normalised and the network till this layer is optimised using the triplet loss function. After this, we have another fully connected layer (fc_3) of dimension equal to the number of classes (7) with softmax activation. Apart from the fc_3 layer, all RCL and FC layers have leaky relu activation. The total number of trainable parameters in this architecture are roughly 1.3 million which is $\sim$10 times the number of training segments.

IV. RESULTS AND DISCUSSION

In this work, our focus was to study the degradation in performance of LID systems under emotional mismatch condition

TABLE II
PERFORMANCE (IN TERMS OF EER) OF THE BASELINE SYSTEMS AND RCNN CONSIDERED UNDER DIFFERENT EMOTIONAL CONDITIONS

Model	Neutral	Anger	Happy	Sad	All Emotions
TDNN	7.13	22.14	20.96	15.88	**16.69**
LSTM	17.26	23.86	27.10	19.91	23.68
i-vector + PLDA	15.41	22.00	18.41	14.03	17.77
x-vector	8.42	19.29	21.41	12.22	17.18
RCNN	11.55	22.86	17.99	14.33	16.92

TABLE III
PERFORMANCE (IN TERMS OF EER) OF THE BASELINE SYSTEMS AND RCNN CONSIDERED UNDER DIFFERENT EMOTIONAL CONDITIONS AFTER **PROSODY MODIFICATION**

Model	Neutral	Anger	Happy	Sad	All Emotions
TDNN	8.42	24.81	26.57	21.17	20.51
LSTM	18.97	19.02	22.17	20.70	21.00
i-vector + PLDA	18.26	20.80	22.48	19.91	20.32
x-vector	5.14	13.97	12.42	8.85	**9.99**
RCNN	20.92	19.13	16.76	14.69	18.49

TABLE IV
PERFORMANCE (IN TERMS OF EER) OF THE BASELINE SYSTEMS AND RCNN CONSIDERED UNDER TRAINING WITH ADAPTATION DATA AND MULTI STAGE TRAINING CONDITIONS

Model	Neutral	Anger	Happy	Sad	All Emotions
TDNN	**6.85**	8.86	11.68	**3.02**	7.60
LSTM	13.02	12.67	20.43	10.36	14.12
i-vector + PLDA	11.27	7.86	10.78	7.69	9.33
x-vector	6.28	10.57	10.78	7.69	8.78
RCNN	10.27	11.14	11.24	5.73	9.74
RCNN + MST	9.58	**5.87**	**9.14**	5.43	**7.47**

during train and test, also to come up with architecture level solutions to the problem. The 7 languages considered for this study along with their utterances in 4 emotions are described in the Section III-A. State-of-art LID systems such as i-vector, TDNN, LSTM and x-vectors are considered as baseline for our work. The train set only consists of neutral utterances, while a relatively smaller adaptation set has been prepared containing emotional utterances for all 4 emotions. We evaluate the performance in terms of Equal Error Rate (EER) as it has been the standard evaluation metric for many speech processing and classification tasks. All the results obtained through our experiments are reported in Table II, III and IV. In all these tables, the All Emotions column correspond to test utterances combined from all the emotions.

Table II highlights the degradation in performance under anger, happy and sad utterances compared to the neutral ones. TDNN, i-vector + PLDA, DNN x-vector and RCNN standout from the LSTM model in terms of performance. The degradation in terms of EER is 100% or more between neutral and emotional conditions for TDNN and x-vector while for other systems it stays in the region of 50-80%.

As evident from Table II, degradation is due to the emotional mismatch, since other characteristics of speech are kept same. Change in emotion of an utterance can be linked to it's prosody. Prosody refers to pitch, duration and energy values and hence we considered modifying them to alter the emotion of an utterance using FAST. Since this would also cause some unnecessary changes in the speech environment, we included some prosody modified utterances during training as well. Adaptation set was used to generate prosody modified utterances for training, while the test utterances of all emotions were prosody modified for evaluation. Table III contains the results obtained for this experiment. Very little improvements can be seen for LSTM, while performance under TDNN, i-vectors and RCNN suffered degradation, but the results highlight that RCNN suffered very minimal degradation in performance. Only x-vectors were able to model the prosody modified utterances properly and a huge improvement in performance is obtained for them.

We train the baseline systems and the RCNN network by combining the adaptation set with train set, and the results along with that for the multi stage training are shown in table IV. From that table, we can observe huge improvement in performance for all the systems. This shows that some emotional data must be present during training to account for the variation. Interesting to note the performance improvement when the multi stage training method is used. Our proposed RCNN when subjected to multi stage training has given the best performance for Anger, Happy and All emotions.

From Table IV, the performance of RCNN under the proposed training method can be observed to not deviate much under various emotions, whereas the x-vector system from Table III shows variation for anger and happy compared to sad and neutral, and same can be identified for the TDNN from table IV. In our view RCNN seems to perform well for LID due to being able to model the multidimensional relations in the input spectrogram and the initial training with the adaptation data made it robust to emotional variations.

V. CONCLUSION

In this work, we addressed the degradation in performance of LID due to mismatch in terms of emotional utterances during train and test. Degradation in performance is very high when no emotional data is present during training, which can be reduced by including emotional utterances while training. We demonstrated the usage of prosody modification for this problem, with little to no improvements. To further improve the performance we introduced the Recurrent Convolutional Neutral Network with multi stage training. The multi stage training can be considered as an adaptation technique. As part of future work, we intend to explore RCNN architecture for other areas in speech processing. We conclude by proposing that care must be taken while preparing datasets for LID by accounting for the emotional variation as well.

REFERENCES

[1] Haizhou Li, Bin Ma, and Kong Aik Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[2] Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, "Multilingual speech recognition with a single end-to-end model," in *Proc. ICASSP*. IEEE, 2018, pp. 4904–4908.

[3] Ye Jia, Ron J Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu, "Direct speech-to-speech translation with a sequence-to-sequence model," *arXiv preprint arXiv:1904.06037*, 2019.

[4] Alex Waibel, Petra Geutner, L Mayfield Tomokiyo, Tanja Schultz, and Monika Woszczyna, "Multilinguality in speech and spoken language systems," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1297–1313, 2000.

[5] Zhiyuan Tang, Dong Wang, and Qing Chen, "AP18-OLR challenge: Three tasks and their baselines," in *Proc. APSIPA ASC*. IEEE, 2018, pp. 596–600.

[6] Seyed Omid Sadjadi, Timothee Kheyrkhah, Audrey Tong, Craig S Greenberg, Douglas A Reynolds, Elliot Singer, Lisa P Mason, and Jaime Hernandez-Cordero, "The 2017 NIST language recognition evaluation," in *Proc. Odyssey*, 2018, pp. 82–89.

[7] Noor Salwani Ibrahim and Dzati Athiar Ramli, "I-vector extraction for speaker recognition based on dimensionality reduction," *Proc. Computer Science*, vol. 126, pp. 1534–1540, 2018.

[8] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language recognition via I-vectors and dimensionality reduction," in *12th annual conference of the ISCA*, 2011.

[9] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 5329–5333.

[10] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "Spoken language recognition using X-vectors," in *Proc. Odyssey*, 2018, pp. 105–111.

[11] M. Wöllmer, F. Eyben, B. Schuller, and G. Rigoll, "A multi-stream asr framework for blstm modeling of conversational speech," in *Proc. ICASSP*, May 2011, pp. 4860–4863.

[12] Zhiyuan Tang, Ying Shi, Dong Wang, Yang Feng, and Shiyue Zhang, "Memory visualization for gated recurrent neural networks in speech recognition," in *Proc. ICASSP*. IEEE, 2017, pp. 2736–2740.

[13] Noor Fathima, Tanvina Patel, C Mahima, and Anuroop Iyengar, "TDNN-based multilingual speech recognition system for low resource Indian languages," in *Proc. INTERSPEECH*, 2018, pp. 3197–3201.

[14] Tirusha Mandava and Anil Kumar Vuppala, "Attention based residual-time delay neural network for Indian language identification," in *Proc. Twelfth International Conference on Contemporary Computing (IC3)*. IEEE, 2019, pp. 1–5.

[15] Bharat Padi, Anand Mohan, and Sriram Ganapathy, "Attention based hybrid I-Vector blstm model for language recognition," in *Proc. INTERSPEECH*, 2019, pp. 1263–1267.

[16] Xiaoxiao Miao, Ian McLoughlin, and Yonghong Yan, "A new time-frequency attention mechanism for TDNN and CNN-LSTM-TDNN, with application to language identification," in *Proc. INTERSPEECH*, 2019, pp. 4080–4084.

[17] Peng Shen, Xugang Lu, Sheng Li, and Hisashi Kawai, "Feature representation of short utterances based on knowledge distillation for spoken language identification.," in *Proc. Interspeech*, 2018, pp. 1813–1817.

[18] Yanmin Qian, Maofan Yin, Yongbin You, and Kai Yu, "Multi-task joint-learning of deep neural networks for robust speech recognition," in *Proc. ASRU*. IEEE, 2015, pp. 310–316.

[19] Qianzi Shen, Zijian Wang, and Yaoru Sun, "Sentiment analysis of movie reviews based on cnn-blstm," in *International Conference on Intelligence Science*. Springer, 2017, pp. 164–171.

[20] Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel, "Image captioning with deep bidirectional LSTMs," in *Proc. of the 24th ACM international conference on Multimedia*, 2016, pp. 988–997.

[21] Yann LeCun, Yoshua Bengio, et al., "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[22] Pedro HO Pinheiro and Ronan Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. ICML*, 2014, number CONF.

[23] Yue Zhao, Xingyu Jin, and Xiaolin Hu, "Recurrent convolutional neural network for speech processing," in *Proc. ICASSP*. IEEE, 2017, pp. 5300–5304.

[24] Ruchir Travadi, Maarten Van Segbroeck, and Shrikanth S Narayanan, "Modified-prior i-vector estimation for language identification of short duration utterances," in *Proc. INTERSPEECH*, 2014, pp. 3037–3041.

[25] Ravi Kumar Vuddagiri, Hari Krishna Vydana, and Anil Kumar Vuppala, "Curriculum learning based approach for noise robust language identification using dnn with attention," *Expert Systems with Applications*, vol. 110, pp. 290–297, 2018.

[26] Eliathamby Ambikairajah, Haizhou Li, Liang Wang, Bo Yin, and Vidhyasaharan Sethu, "Language identification: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.

[27] Marius Vasile Ghiurcau, Corneliu Rusu, and Jaakko Astola, "A study of the effect of emotional state upon text-independent speaker identification," in *Proc. ICASSP*. IEEE, 2011, pp. 4944–4947.

[28] Bogdan Vlasenko, Dmytro Prylipko, and Andreas Wendemuth, "Towards robust spontaneous speech recognition with emotional speech adapted acoustic models," in *35th German Conference on Artificial Intelligence (KI-2012), Saarbrücken, Germany (September 2012)*. Citeseer, 2012, pp. 103–107.

[29] Priyam Jain, Krishna Gurugubelli, and Anil Kumar Vuppala, "Study on the effect of emotional speech on language identification," in *Proc. NCC*. IEEE, 2020, pp. 1–6.

[30] Koichi Shinoda, "Speaker adaptation techniques for automatic speech recognition," *Proc. APSIPA ASC*, vol. 2011, 2011.

[31] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proceedings of the iEEE conference on computer vision and pattern recognition*, 2016, pp. 1335–1344.

[32] Ibon Saratxaga, Eva Navas, Inmaculada Hernáez, and Iker Luengo, "Designing and recording an emotional speech database for corpus based synthesis in Basque," in *Proc. LREC*, 2006, pp. 2126–2129.

[33] Sungbok Lee, Serdar Yildirim, Abe Kazemzadeh, and Shrikanth Narayanan, "An articulatory study of emotional speech production," in *Proc. Ninth European Conference on Speech Communication and Technology*, 2005.

[34] Ravi Kumar Vuddagiri, Krishna Gurugubelli, Priyam Jain, Hari Krishna Vydana, and Anil Kumar Vuppala, "IIITH-ILSC speech database for Indain language identification," in *Proc. SLTU*, 2018, pp. 56–60.

[35] Stephan Radeck-Arneth, Benjamin Milde, Arvid Lange, Evandro Gouvêa, Stefan Radomski, Max Mühlhäuser, and Chris Biemann, "Open source German distant speech recognition: Corpus and acoustic model," in *Proc. International Conference on Text, Speech, and Dialogue*. Springer, 2015, pp. 480–488.

[36] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss, "A database of German emotional speech," in *Proc. Ninth European Conference on Speech Communication and Technology*, 2005.

[37] Shashidhar G Koolagudi, Ramu Reddy, Jainath Yadav, and K Sreenivasa Rao, "IITKGP-SEHSC: Hindi speech corpus for emotion analysis," in *Proc. International conference on devices and communications*. IEEE, 2011, pp. 1–5.

[38] Shashidhar G Koolagudi, Sudhamay Maity, Vuppala Anil Kumar, Saswat Chakrabarti, and K Sreenivasa Rao, "IITKGP-SESC: speech database for emotion analysis," in *Proc. International conference on contemporary computing*. Springer, 2009, pp. 485–492.

[39] P Gangamohan, Vinay Kumar Mittal, and Bayya Yegnanarayana, "A Flexible Analysis Synthesis Tool (FAST) for studying the characteristic features of emotion in speech," in *Proc. CCNC*. IEEE, 2012, pp. 250–254.