Cross-Lingual Speaker Identification for Indian Languages

Amaan Rizvi **Anupam Jamatia** **Dwijen Rudrapal** **Kunal Chakma**
Department of Computer Science and Engineering
National Institute of Technology Agartala
Tripura, India
{amaan.rizvi39,anupamjamatia,dwijen.rudrapal,kchax4377}@gmail.com

Björn Gambäck
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
gamback@ntnu.no

Abstract

The paper introduces a cross-lingual speaker identification system for Indian languages, utilising a Long Short-Term Memory dense neural network (LSTM-DNN). The system was trained on audio recordings in English and evaluated on data from Hindi, Kannada, Malayalam, Tamil, and Telugu, with a view to how factors such as phonetic similarity and native accent affect performance. The model was fed with MFCC (mel-frequency cepstral coefficient) features extracted from the audio file. For comparison, the corresponding mel-spectrogram images were also used as input to a ResNet-50 model, while the raw audio was used to train a Siamese network. The LSTM-DNN model outperformed the other two models as well as two more traditional baseline speaker identification models, showing that deep learning models are superior to probabilistic models for capturing low-level speech features and learning speaker characteristics.

1 Introduction

Ascertaining the identities of the writers and speakers are important tasks in language and speech processing. The vocabulary a person uses as well as the ways a person writes and talks can give us information about their identity or their background. Furthermore, people's voices are unique identifiers, just like their retinas and fingerprints, making speaker recognition (the task of recognising the voice of a speaker based on audio input) applicable to building human-to-machine interaction and biometric solutions such as voice assistants, voice-controlled services, and speech-based authentication products (Beigi, 2011). There are two basic speaker recognition tasks:

(i) ***Speaker Verification***: confirm the identity of a speaker.
(ii) ***Speaker Identification***: identify a voice in a set of speakers.

Speaker recognition can be monolingual as well as cross-lingual (Sale et al., 2018). For monolingual tasks, the same language is used to both train and test models. In cross-lingual speaker recognition, a model is trained on one language, e.g., *English*, and tested on a different language, e.g., *Arabic*.

In a multilingual country like India, with more than 120 languages having tens of thousands of speakers and some 50 languages having official status at national or regional level, most citizens speak several languages fluently. Due to this plethora of multilingual speakers, it is not feasible to train a speaker recognition model in one language and re-train the model in a new language. Therefore, the development of cross-language speaker recognition models has become a salient task. Intuitively, language mismatch in training and test language should not be a problem, since a person's vocal traits have nothing to do with what they are saying, but in general, the performance of a speaker recognition system still degrades when a model is trained on one language and verification is done on another (Li et al., 2017b). Probabilistic models like Gaussian Mixture Model (GMM; Reynolds and Rose, 1995) and Gaussian Mixture Model-Universal Background Model (GMM-UBM; Reynolds et al., 2000) have traditionally been used for speaker recognition; however, in recent years deep learning-based approaches have outperformed probabilistic-based ones for both speaker identification and speaker verification.

Proceedings of Recent Advances in Natural Language Processing, pages 979–987
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_105

This paper reports on research conducted on five Indian languages: Hindi, Kannada, Malayalam, Telugu, and Tamil. English was used as the training language for the models. Previous research has shown that extracting features from the audio signal and using them as input to the model will produce much better performance than directly considering raw audio signal as input. Here raw audio, mel-frequency cepstral coefficients (MFCCs; Dave, 2013), and spectrogram images were utilised as input. The impact of language mismatch, the number of speakers, and the duration of utterances were studied while comparing the performances of the three input methods.

The rest of the paper is structured as follows: Section 2 describes related work in the domain, while Section 3 presents the methodology and proposed neural network architecture. Experimental results are discussed in Section 4 and further analysed in Section 5, while Section 6 concludes the observations.

2 Related Work

Cross-lingual speaker recognition has been in focus for researchers for some time because of the abundance of bilingual speakers in the world. Ma and Meng (2004) studied the enrollment-test mismatch and found that it caused significant performance degradation for speaker recognition. Auckenthaler et al. (2001) investigated the mismatch between training and operation, within a GMM-UBM architecture, finding considerable performance degradation if the speech data used to train the Universal Background Model and the data used to validate/test speakers were in different languages. Misra and Hansen (2014) drew similar conclusions when utilizing a model based on i-vectors (Dehak et al., 2010), an intermediate vector representation between Gaussian Mixture Models and MFCC.

Several Deep Neural Network (DNN) models have been proposed for the speaker recognition task, with Li et al. (2017b) arguing that the reason for performance degradation in the cross-lingual environment is the use of probabilistic-based models—as in all the above-mentioned methods—and showing considerable improvement when using a DNN model. Heigold et al. (2016) proposed a text-dependent speaker verification architecture utilising an LSTM to extract d-vectors, i.e., embeddings over the averaged activation from the network's last hidden layer, with Deep Speaker by Li et al. (2017a) showing better results than i-vector based methods.

Snyder et al. (2018) introduced the concept of x-vector embeddings, a model based on a Time-Delay Deep Neural Network architecture that computes speaker embeddings from variable-length acoustic segments. The network consists of layers that operate on speech frames, a statistics pooling layer that aggregates over the frame-level representations, additional layers that operate at the segment level, and finally a softmax output layer. The embeddings are extracted after the statistics pooling layers. Koluguri et al. (2020) described SpeakerNet, an architecture using an x-vector-based statistics pooling layer to map variable-length utterances to a fixed-length embedding. Novoselov et al. (2022) presented a transformer-based speaker recognition system using wav2vec 2.0 (Baevski et al., 2020).

This paper broadly discusses two main approaches to feature extraction: (i) *MFCC-based* and (ii) *Spectrogram-based*. Due to its computational simplicity and robustness to multicollinearity, MFCC is the most popular feature extraction technique among researchers. MFCC yields uncorrelated features which are favorable for linear models like support vector machines (SVM) and Gaussian mixture models. In the MFCC-based approach, filter banks are designed in a manner to operate in a similar way to the human auditory frequency perception. Many fusions of MFCC-based features have been studied. Combining two different sets of features from MFCCs and Perceptual Linear Predictive Coefficients (PLPC) using ensemble classifiers in conjunction with principal component transformation can significantly improve the performance of MFCC-GMM speaker recognition systems (Bose et al., 2017). Combining MFCC features with Residual Phase Cepstrum Coefficients (RPCC) also offers significant overall improvement to the robustness and accuracy of speaker identification tasks (Bo et al., 2014). Ma et al. (2016) used MFCC incorporated into a histogram transform feature for text-independent speaker identification.

Spectrogram images as a feature for convolutional neural network (CNN) models have also been explored (Bunrit et al., 2019; Kadyrov et al., 2021), by extracting spectrogram images from audio files and feeding them to a CNN. The network's performance improved significantly when there were short utterances and a moderate amount of audio files present per speaker.

Language	Speakers			Utterances		
	Male	Female	Total	Male	Female	Total
Hindi	21	8	29	959	395	1354
Kannada	12	6	18	591	263	854
Malayalam	14	6	20	608	289	897
Tamil	14	14	28	604	607	1211
Telugu	15	10	25	639	533	1172
English	76	44	120	4351	1321	5672

Table 1: Gender wise distribution of speakers

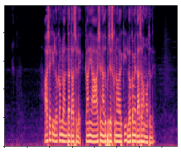


Figure 1: Mel-spectrogram obtained from an audio file

3 Methodology

The National Institute of Technology Karnataka's speaker profiling dataset (NISP; Kalluri et al., 2021) was used for the experiments. It contains recordings of some 4–5 minutes each of speakers talking in both English and their mother tongues. The corpus includes Hindi, which is an Indo-Aryan language, together with four Dravidian languages: Kannada, Malayalam, Tamil, and Telugu. The text prompts used for the recordings were presented in two different sessions to the speakers, in their native language and in English, respectively. The data was sampled at 44.1 kHz with a bitrate of 16 bits per sample. Each speaker's data consists of 30 to 40 audio files in *.wav* format.

A subset of the original NISP dataset was used to train the models, due to limitations of available computing resources. The dataset statistics are summarised in Table 1. The total number of utterances is 5,488 in the native languages and 5,672 in English. Overall, there are 76 male speakers and 44 female speakers, in the age group of 18 to 45.

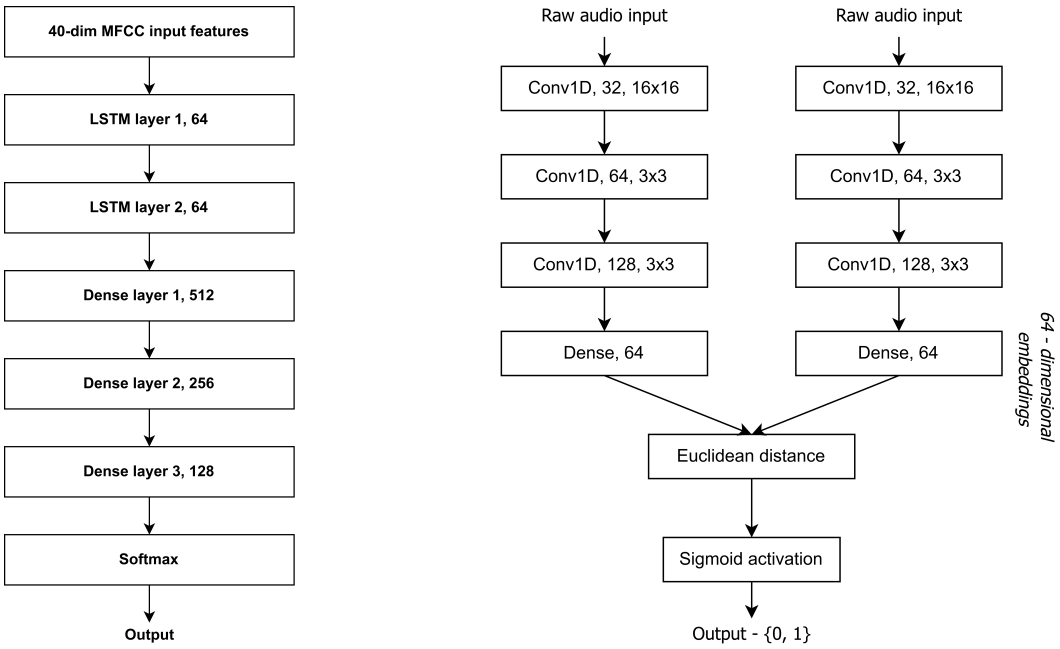
3.1 Feature Extraction

The goal of feature extraction is to transform an input waveform into a sequence of feature vectors that can be fed to a machine-learning model. Each feature vector represents information corresponding to a small time window in a signal. Two feature extraction methods were used, spectrogram images and mel-frequency cepstral coefficients (MFCC).

A *spectrogram* is a visual representation of a signal's strength, as it varies over time at different frequencies. It is basically a three-dimensional graph, where the x-axis represents time, the y-axis represents frequency, and the colour or intensity of the graph at each point represents the magnitude or power of the signal at that frequency and time. A spectrogram image represents the level of energy from light to dark. In case the colour is white or nearly white, there is little or no energy. Conversely, if there is a lot of energy, the colour is black colour or nearly black. A *mel-spectrogram* is obtained by converting a spectrogram to a mel scale. The Python library Librosa was used to extract the mel-spectrogram for each audio file and save it as .png files. An example of the mel-spectrogram image obtained from an audio file is shown in Figure 1. Spectrograms were obtained using a sample rate of 22,050 times/second and an FFT (Fast Fourier Transform) window size of 2,048 samples.

Mel-frequency cepstral coefficients (MFCC) is the most common feature extraction technique. It is based on the idea of cepstrum (Bogert et al., 1963), which is the inverse FT of the logarithm of the estimated signal spectrum. Five steps are involved in deriving MFCC: (i) pre-emphasis, which boosts the amount of energy in high frequencies, since there is more energy at lower frequencies than at higher in spectrum voice segments like vowels; (ii) windowing, which slices the audio waveform into smaller sliding frame windows, assuming the signal in each frame to be stationary; (iii) Discrete Fourier Transform (DFT) is used to extract spectral information (magnitude and phase) from a windowed signal; (iv) mel filter and log, with a set of filters converting the DFT spectrum to a mel-cepstrum and taking the natural logarithm of each mel-cepstrum value; and (v) inverse discrete Fourier transform, which computes the cepstrum as the inverse DFT of the logarithm of the signal spectrum.

For the experiments, 40 MFCC features were extracted using the Librosa library for music and audio analysis. The number of 40 MFCC features extracted for each audio file is a typical value used in speech-processing applications. This is because 40 MFCC features provide a good balance between capturing relevant information and reducing the dimensionality of the data. The function allows for customisation of the number of MFCC features to extract, as well as other parameters such as the sam-



(a) LSTM dense neural network architecture

(b) Siamese network for few-shot learning

Figure 2: Model architectures

pling rate and window size. The MFCC features were extracted frame by frame, with each frame representing a short segment of the audio signal. The frames were then averaged across the different frames for each audio file to obtain a single set of 40 MFCC features for each file.

3.2 Model Architectures

Five different machine learners were evaluated on the cross-lingual speaker identification task. An SVM classifier trained with the 40-dimensional MFCC features was included as a baseline and a GMM-UBM architecture trained on the same features was added for comparison since those two approaches have traditionally been the go-to solutions for speaker identification.

For the main experimental architecture, the MFCC features were used as input to a Long Short-Term Memory-based dense neural network (LSTM-DNN) model, as shown in Figure 2a. The architecture was implemented using Keras and trained on Google Colab, with categorical cross entropy as a loss function and compiled using the Adam optimizer with a 0.001 learning rate. The network has two LSTM layers with 64 units each and a recurrent dropout of 0.2; the output of the last LSTM layer feeds into the first dense layer. Three dense layers are utilised with 512, 256, and 128 units, respectively, and ReLU (Rectified Linear Unit) activation functions. A dropout layer is added after each dense layer with a dropout rate of 0.2. Finally, a softmax layer denotes the number of speakers used for training. The model was trained for 500 epochs with batch sizes of 32 for all datasets.

For comparison, experiments were also carried out with a few-shot learning approach to speaker identification using a Siamese network architecture, shown in Figure 2b. The network consists of two identical encoder modules built with convolution blocks. At the end of the encoder block, a dense layer with 64 units is utilised to get a 64-dimensional embedding of speaker input. Euclidean distance is used to calculate the distance between two embeddings and create a 1-dimensional vector that is then passed to the sigmoid function.

Six audio files were sampled for each speaker to create a dataset of similar pairs with label 1 and dissimilar pairs with label 0. During training, the pair of raw audio inputs were fed into two different encoder blocks. In the first phase, the Siamese model was trained for 50 epochs using batch size 32 and Adam optimizer with a 0.001 learning rate. In the second phase, the training inputs were passed through one encoder block to get the 64-dimensional embeddings, and a softmax function was applied on top of it to output speaker identity. The single encoder block was trained with softmax output for 50 epochs.

Language	GMM-UBM	SVM	LSTM-DNN	Siamese	ResNet-50
Hindi	80.34	89.32	95.17	93.83	**96.67**
Kannada	88.41	97.11	**98.27**	97.89	92.51
Malayalam	49.92	68.12	76.81	**80.59**	72.75
Tamil	81.39	89.20	**95.47**	94.68	77.70
Telugu	81.23	93.43	95.50	**96.79**	94.95

(a) Five speakers per language

Language	GMM-UBM	SVM	LSTM-DNN	Siamese	ResNet-50
Hindi	76.34	85.31	90.07	78.68	**91.76**
Kannada	79.55	89.26	**91.32**	81.52	87.15
Malayalam	38.85	61.96	68.94	65.36	**70.48**
Tamil	73.90	80.40	**83.12**	73.56	69.50
Telugu	72.81	83.67	**84.09**	72.45	**84.10**

(b) All speakers for each language

Language	GMM-UBM	SVM	LSTM-DNN	Siamese	ResNet-50
Hindi	92.06	94.70	98.51	92.01	**98.67**
Kannada	91.05	95.37	**98.15**	90.04	95.01
Malayalam	92.46	95.93	97.67	90.82	**98.26**
Tamil	89.10	92.80	95.10	91.30	**96.04**
Telugu	91.95	94.41	**96.65**	89.35	94.30

(c) Model performance when evaluated in the same language

Table 2: Model accuracies across all languages

As a fifth and final architectural alternative, the ResNet-50 (He et al., 2016) model was trained on mel-spectrogram feature input, again using Google Colab. A dense layer with 256 neurons was added on top of the ResNet-50 model, with a softmax layer as output. The model was trained for 300–400 epochs, the Adam optimizer was employed with exponential learning rate decay, and categorical cross-entropy was selected as the loss function.

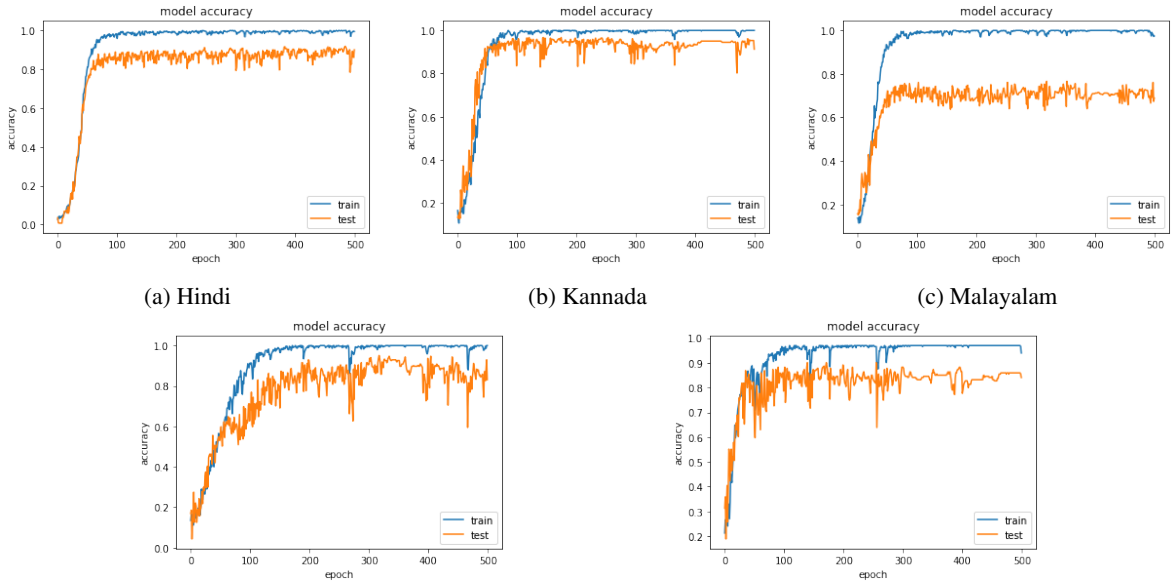
4 Results and Discussion

The results of the experiments are summarised in Table 2, with accuracy as the performance metric. English was used as the training language for all speakers and the trained models were validated on the speakers' native languages. All models were first tested using only five speakers and then on the complete 120-speaker dataset (i.e., with the number of speakers per language as given in the fourth column of Table 1). In addition to the cross-lingual experiments, performance was evaluated also for the mono-lingual case, that is, with the models being trained and evaluated on the same language, on the complete dataset.

The cross-lingual experiments with only five speakers per language (Table 2a) show the few-shot learning-based Siamese network using raw audio input performing better than the ResNet-50 model. However, the limitations of the few-shot learning approach can be observed when the number of speakers is increased; its accuracy drops significantly on all languages when all speakers are included and the Siamese network then performs worse than even the SVM model (Table 2b).

In general, we can notice that the speaker identification accuracy drops for all models when the number of speakers is increased. This means that as the number of speakers in the dataset increases, it becomes more difficult for the models to accurately identify individual speakers. The variations in accuracy over the five languages show the effect of the native accent of speakers and the phonetic similarity (Bradlow et al., 2010) between training and test languages. The native accent of speakers refers to the way in which they pronounce words and phrases based on their regional or cultural background. The phonetic similarity between languages refers to the degree to which the sounds and pronunciation of words in one language are similar to those in another language.

The learning curves in Figure 3 show the performance of the model during training and testing across all five languages. Table 2b shows the accuracy of the model on the test data for each language,



(a) Hindi (b) Kannada (c) Malayalam

(d) Tamil (e) Telugu

Figure 3: Training (blue) and test (orange) learning curves for the LSTM-DNN model

with the poor results for Malayalam most likely due to the overfitting which can be observed in the Malayalam learning curve (Figure 3c). Overall, the learning curves provide insight into the performance of the model during training and testing and can help identify issues such as overfitting that may affect the model's performance on new data.

Table 2c presents the performance of models when trained and evaluated on the same language, with a 97.67% accuracy of the LSTM-DNN model when both trained and tested using Malayalam. Performance degradation can in general be observed when systems are evaluated in cross-lingual environments (Sirsa and Redford, 2013), but the high Malayalam degradation indicates the impact of language mismatch and the speakers' native accents.

Table 3 summarises the model setups and gives their average accuracy performance figures for all speakers, over all five languages. As can be seen, the LSTM-DNN model outperforms the GMM-UBM and SVM systems traditionally used for speaker identification, as well as both the Siamese network and the ResNet-50 model.

Model	Feature extraction	Accuracy
GMM-UBM	MFCC	68.29
SVM	MFCC	80.12
Siamese	Raw audio	74.31
ResNet-50	Mel-spectrograms	80.59
LSTM-DNN	MFCC	**83.51**

Table 3: Summary of all the models

The average speaker identification accuracy for the ResNet-50 model could have been improved by providing more spectrogram images for training. However, as can be seen in Table 2b, for Hindi and Malayalam the ResNet-50 model outperforms the LSTM-DNN and equals it for Telugu when the number of speakers is maximised. CNN-based models rely heavily on the number of images available for training, but in a real-world scenario, it is not feasible to get thousands of speech utterances for an individual speaker.

5 Ablation Study

To evaluate the LSTM-DNN model, several parameter variations were tested, analysing changes in one parameter at the time, while keeping the other parameters constant.

Four groups of ablations were examined. First, different feature extraction techniques. Second, to explore the effects of regularization in LSTM layers, recurring dropout rates were set to none, 0.2, and 0.5, respectively. Third, the impact of reducing the number of LSTM layers. Finally, the learning rates, with two constant learning rates of 0.001 and 0.0001, and an exponential schedule with an initial rate of 0.01 and a decay rate of 0.9.

Ablation		Hindi	Kannada	Malayalam	Tamil	Telugu
Raw audio		67.43	59.11	56.90	67.99	69.56
MFCC		90.07	91.32	68.94	83.12	84.09
Recurrent dropout	none	88.41	87.50	63.96	78.64	82.05
	0.2	90.07	91.32	68.94	83.12	84.09
	0.5	84.29	84.25	59.29	74.83	73.09
LSTM layers	1	84.68	84.31	63.68	82.16	79.99
	2	90.07	91.32	68.94	83.12	84.09
Learning rate	0.001	90.07	91.32	68.94	83.12	84.09
	0.0001	88.87	90.09	70.53	85.02	82.19
	exp	88.47	88.05	65.23	84.64	76.04

Table 4: Feature ablation for the LSTM-DNN model

As the accuracy results in Table 4 show, employing MFCC features as inputs, as opposed to raw audio, considerably enhanced performance. It is crucial to select an adequate recurrent dropout rate since the performance was negatively impacted by setting it too high. Performance was improved by using more dense LSTM layers, although this comes with a higher computational cost.

6 Conclusion

An LSTM dense neural network model for cross-lingual speaker identification is proposed in this work. The model was trained using speaker recordings in English and cross-lingual speaker identification was performed on five Indian languages: Hindi, Kannada, Malayalam, Tamil, and Telugu.

There was a clear variation in speaker identification accuracy across the different languages. Since English was used for training for all speakers, the variation in accuracy is arguably due to variations in phonetic features of the native test languages, as well as any phonetic similarity between those languages and English.

The average classification accuracy on the test data for the LSTM-DNN method was 83.51%, with 68.29% for GMM-UBM, and 80.12% for SVM, with those three learners trained using MFCC (mel-frequency cepstral coefficient) features. A Siamese network using raw audio input reached 74.31% accuracy and a ResNet-50 trained on mel-spectrograms 80.59% accuracy. The LSTM-DNN model thus yielded better average accuracy than the other models, showing the efficiency of an LSTM-DNN trained using MFCC features input under the constraint of limited data.

The Siamese network few-shot learning approach using simple raw audio input is good when there are few speakers but fails to generalise over a significant number of speakers. A complex CNN-based model with spectrogram inputs like ResNet-50 gives better results than MFCC feature extraction when there are sufficient images available to train the model; however, the scarcity of image data is a bottleneck for that approach. Finally, the traditional probabilistic GMM-UBM performed worst of all models in the cross-lingual environment.

While this research focused on speaker identification, the work can also be used as a springboard to develop more advanced frameworks like *x-vectors* for Indian languages and apply the methods to the speaker verification problem.

The models developed can furthermore be utilised in isolation or together with text-based feature extractors for similar digital forensic tasks such as author profiling or native language identification, i.e., to recognize a person's L1 (native language) based on text and speech produced in a foreign language (L2).

Acknowledgements

We would like to express our sincere gratitude to Kalluri et al. for generously making their dataset available to the scientific community. We acknowledge the effort, dedication, and expertise of the researchers involved in the development of the NISP dataset. Their research and dataset have undoubtedly enriched our work and furthered the progress of the scientific community as a whole.

We are furthermore indebted to the anonymous reviewers for their contributions to improving the readability and quality of the paper.

References

Roland Auckenthaler, Michael J. Carey, and John S.D. Mason. 2001. Language dependency in text-independent speaker verification. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 441–444. IEEE.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing systems*, 33:12449–12460.

Homayoon Beigi. 2011. *Fundamentals of speaker recognition*. Springer Science & Business Media.

Cheng Bo, Lan Zhang, Taeho Jung, Junze Han, Xiang-Yang Li, and Yu Wang. 2014. Continuous user identification via touch and movement behavioral biometrics. In *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE.

Bruce P. Bogert, Michael J.R. Healy, and John W. Tukey. 1963. The quefrency analysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking. In *Proceedings of the Symposium on Time Series Analysis*, pages 209–243.

Smarajit Bose, Amita Pal, Anish Mukherjee, and Debasmita Das. 2017. Robust speaker identification using fusion of features and classifiers. *International Journal of Machine Learning and Computing*, 7(5):133–138.

Ann Bradlow, Cynthia Clopper, Rajka Smiljanic, and Mary Ann Walter. 2010. A perceptual phonetic similarity space for languages: Evidence from five native language listener groups. *Speech Communication*, 52(11-12):930–942.

Supaporn Bunrit, Thuttaphol Inkian, Nittaya Kerdprasop, and Kittisak Kerdprasop. 2019. Text-independent speaker identification using deep learning model of convolution neural network. *International Journal of Machine Learning and Computing*, 9(2):143–148.

Namrata Dave. 2013. Feature extraction methods LPC, PLP and MFCC in speech recognition. *International Journal of Advanced Research in Engineering and Technology*, 1(6):1–4.

Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2010. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE.

Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. 2016. End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE.

Shirali Kadyrov, Cemil Turan, Altynbek Amirzhanov, and Cemal Ozdemir. 2021. Speaker recognition from spectrogram images. In *2021 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, pages 1–4. IEEE.

Shareef Babu Kalluri, Deepu Vijayasenan, Sriram Ganapathy, Prashant Krishnan, et al. 2021. NISP: A multi-lingual multi-accent dataset for speaker profiling. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6953–6957. IEEE.

Nithin Rao Koluguri, Jason Li, Vitaly Lavrukhin, and Boris Ginsburg. 2020. SpeakerNet: 1D depth-wise separable convolutional network for text-independent speaker recognition and verification. *arXiv preprint arXiv:2010.12653*.

Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. 2017a. Deep Speaker: An end-to-end neural speaker embedding system. *arXiv preprint arXiv:1705.02304*.

Lantian Li, Dong Wang, Askar Rozi, and Thomas Fang Zheng. 2017b. Cross-lingual speaker verification with deep feature learning. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1040–1044. IEEE.

Bin Ma and Helen Meng. 2004. English–Chinese bilingual text-independent speaker verification. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages V–293. IEEE.

Zhanyu Ma, Hong Yu, Zheng-Hua Tan, and Jun Guo. 2016. Text-independent speaker identification using the histogram transform model. *IEEE Access*, 4:9733–9739.

Abhinav Misra and John H.L. Hansen. 2014. Spoken language mismatch in speaker verification: An investigation with NIST-SRE and CRSS Bi-Ling corpora. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 372–377. IEEE.

Sergey Novoselov, Galina Lavrentyeva, Anastasia Avdeeva, Vladimir Volokhov, and Aleksei Gusev. 2022. Robust speaker recognition with transformers using wav2vec 2.0. *arXiv preprint arXiv:2203.15095*.

Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. 2000. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41.

Douglas A. Reynolds and Richard C. Rose. 1995. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83.

Pritam Limbaji Sale, Spoorti J. Jainar, and B.G. Nagaraja. 2018. A comparison of features for multilingual speaker identification—a review and some experimental results. *International Journal of Recent Technology and Engineering*, 7(4S2):299–304.

Hema Sirsa and Melissa A. Redford. 2013. The effects of native language on Indian English sounds and timing patterns. *Journal of Phonetics*, 41(6):393–406.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE.