Text-Independent Speaker Classification Using Deep Learning

Feras Kiki [1]

Abstract

Automatic Speech recognition (ASR) and automatic speaker identification (SID) have got a lot of interest recently, especially with the increased capabilities of deep learning and AI. Their applications span a wide variety of topics including forensics, social media, education, and daily task user assistance. In this work, we aim to implement an end-to-end speaker classification system based on deep learning techniques such as, RNN, CNN, on common datasets such as VoxCeleb, Librispheech and report the accuracy and comparison with other approaches.

1. Introduction

Natural Language Processing (NLP) is among the most important fields in AI and it includes many sub-fields, including speech and speaker recognition. While the task of transforming speech-to-text is fundamental in understanding human voice input, classifying the speaker can be just as important for a true speech understanding. This is vital for the case when multiple people speak or for security identification. There are 2 common tasks in this approach:

1. Speaker classification/recognition. Here the task is to check if an audio input matches a class the model knows about. There 2 types for this: Text-dependent, where the text of the speech is known beforehand, and text-independent, where the speaker can say anything.

2. Speaker Verification. This task is mainly to check if 2 input audios match the same speaker and it can be regarded as a binary task. Here the new audios can be new and outside the training samples. Common metrics for this task include Equal Error Rate (EER), False Acceptance Rate(FAR), and False Rejection Rate (FRR)

In this work, we are mainly interested in the first (i.e classification) and we leave the other for possible future work.

2. Related Work

Numerous works have added great contributions to this problem in recent. (Anand et al., 2019) reported a few-shot learning model where they experimented with 2 approaches: VGG ResNet, and Capsule net. They reported accuracy on 50 , 200 classes on VoxCeleb1 dataset (Nagraniy et al., 2017) (a difficult dataset) where the highest on the 50 class was ResNet 90.3% and VGG 76.7% and their proposed Capsule Net achieved 67.7% in top-1 accuracy. (An et al., 2019) used a CNN and attention mechanism combination and reported a remarkable top-1 accuracy of 90.8%. Their work however was only marginally better than (Cai et al., 2018) who also tried attention-based mechanisms on CNN and ResNet variants and reported an 89.8% top-1 accuracy on VoxCeleb.

The state-of-the-art (SOTA) reported performance on the VoxCeleb1 dataset was 95.3% achieved by (Niizumi et al., 2023) where they used a masked auto-encoder-based architecture.

(Ding et al., 2020) proposed the AutoSpeech model for speaker recognition tasks. They reported a neural architecture search approach based on CNN and their results outperform other approaches like VGG-M, REsNet-18, and ResNet-34. (India et al., 2019) reported a CNN-based model with self multi-head attention model for speaker classification. (Vaessen & van Leeuwen, 2022) fine-tuned a WAV2VEC2 speech recognition to speaker recognition and reported improved performance and 1.88% EER .

(Vazhenina & Markov, 2020) reported an End-to-End speech recognition based on Fourier and Hilbert spectrum features. They used the Short-Time Fourier Transform (STFT) in combination with Hilbert-Huang transform (HHT) to reduce noise (Vazhenina & Markov, 2020). Their approach utilized CNN and was used on 2 datasets: Wall Street Journal (WSJ) (Paul & Baker, 1992) and CHiME-4 dataset and compared STFT and 2 other hilbert spectrum based decompositions: Emprical Mode Decomposition (EMD) and Variational Mode Decomposition (VMD). STFT performed the highest according to their results (Vazhenina & Markov, 2020). Another approach utilizing STFT and MFCC was an LSTM based work on Chinese Mandarin Corpus dataset (El-Moneim et al., 2020). Their approach reported up to 98.7% on their dataset, but no comparison

[1] Department of Mechanical Engineering. Correspondence to: Feras Kiki <fkiki18@ku.edu.tr>.

COMP541Deep Learning, Fall 2023.

with other datasets were provided.

Another work on LSTM for speaker recognition was done by (Hu et al., 2021) where they worked on a 3DCNN-LSTM approach on a Mandarin open source dataset (Bu et al., 2018) that included 400 speakers and 170 hours. Their work compared 3DCNN, LSTM 3DCNN-LSTM, 3DCNN-BLSTM, they achieved a top accuracy of 90% using 3DCNN-LSTM followed by 3DCNN-BLSTM of 86%. A BiLSTM based network was used by (Alashban & Alotaibi, 2021) to do speaker gender classification in mono and cross language. They used (Moz) Mozzila Common Voice dataset (open source). Their model achieved above 91% in the test set where they used Arabic and English voices in different set mixes. Their dataset included bilingual people who can speak both Arabic and English and they were trying to classify voice across languages. For example, Set A/A was train/test/val Arabic, whereas set A/E was Ar/Ar/En, etc. The accuracy changed among the different sets: lowest was 75% on A/E and highest 91.% on A/A.

Another interesting work was to do an overlapping multi-talker speaker identification (Tran & Tsai, 2020). They used a 1DCNN apprach to classify multi-speakers. For their data, they artificially mixed numerous speakers using an overlapping energy ratio and they compared that also with noise. Their approach only included a closed set (no out of training persons) and they used their own developed dataset.

(Li et al., 2017) focused on speaker verification in a text-independent task where they employed a convolutional time-delay deep neural network (CT-DNN) and focused on the Fisher (fis, 2004) database. Their approach utilized a frequncey approach where they used Mel frequency cepstral coefficients (MFCC). Other common filters such as Linear Prediction Coefficients (LPC) and Linear Prediction Cepstrum Coefficients (LPCC) are also other filtering approaches report (Kabir et al., 2021). Numerous embedding techniques are also reported by numerous works (Kabir et al., 2021), such as i-vector (identity vector) (Dehak et al., 2011), x-vector (Snyder et al., 2017), t-vector (triplet speaker embeddings) (Zhang et al., 2019).

2.1. Common Challenges

There are numerous challenges in speaker classification task(Kabir et al., 2021).

- Dataset Cleanliness. There are clean speech datasets where noise is minimal and there is an in-the-wild dataset (Kabir et al., 2021). The former is collected in controlled conditions where minimal noise is present, the latter doesn't control for any parameter and can contain actual noise, vibration , and intra-speaker variability (Kabir et al., 2021)
- Text dependency. The work can be text dependent where voice matches a given text, or totally variable and text-independent (the focus of the current project)
- open/closed set: Open sets include unknown speakers to the training set in contrast with closed set where all speakers are included in the training set (focus of this project) (Kabir et al., 2021).
- Duration of exposure to the speaker. With more data, it is expected that the model performs better.
- Cross language classification: where the speaker uses more than one language.
- multi-speaker classification: performing the classification when more than one speaker is speaking.

2.2. Datasets

The main datasets that will be used in the project are shown in Table 1 including Librispeech (lib, 2015), which is a big audio-book-based dataset and is relatively clean, and VoxCeleb1 (Nagraniy et al., 2017), which is considered a speaker-in-the-wild (where there can be high variability in quality) audio dataset for 1,251 celebrities and 145,000+ utterances retrieved from Youtube. The VoxCeleb dataset has both clear and noisy audio tracks. The Authors reported a top-1% accuracy of 80.5 and top-5% of 92.1 using a CNN model. The top reported accuracy on Papers with codes was 95.3% achieved by Masked Auto-encoder (MAE) based model (PaperswithCode, 2023). There are other datasets such as VoxCeleb2 (Chung et al., 2018), which is a bigger version that has 7,000+ speakers, and various other speech-based datasets that can be found on (ope) and they contain numerous languages, including Arabic, Chinese, Korean, Hindi, Spanish, etc and dataset types, including text-dependent types.

Table 1. Datasets Information

Information	**VoxCeleb**	**LibriSpeech**
Date Released	2018	2015
Open Access	Yes	Yes
# Speakers	1251	1160+
Difficulty	Hard	Medium
Size	36+ GB	60+ GB
Duration	-	1000+ hours
Gender (Male/Female)	55/45	51/49
Multi-Nationality	Yes	-
Multiple Duration Length Sub-classes	Yes	Yes
Train/Test Split in My Use	80/20	80/20

3. Workflow

In this section, we outline the workflow that is shawn in Figure 3 of our audio classification project, emphasizing

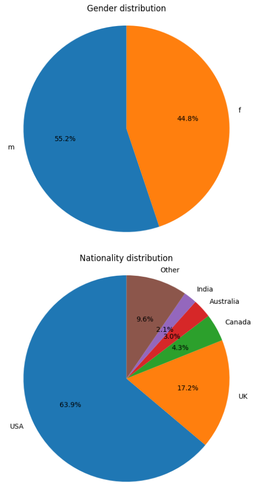

Figure 1. VoxCeleb1 MetaData plots the gender distribution (top) and nationality distribution (bottom)

the essential prepossessing step involving the Short-Time Fourier Transform (STFT). The initial approach for any audio model would be to use the audio signal directly. However, using the temporal signal (the raw audio signal) is not the common method used in the literature. Instead, it is common to use STFT and get the frequency response instead and then use that as features that are input to the model. This can be done using an image format or 2D, and some people use that in the literature. This approach can take advantage of the available models and then do transfer learning on our own task. However, this is not necessary and our approach is to use multi-1D vectors as inputs. We are reporting the results of the 1D approach mainly and future work can compare other variants like 2D-based ones.

3.1. Audio Data Preprocessing

Prepossessing is a common step in all data-based tasks. While filtering and other operations can be done to enhance the signal, we will focus mainly on STFT in this discussion.

Difference between Raw Audio and STFT:

- **Raw Audio Signal:** The raw audio signal represents sound as a continuous waveform, which can be challenging to analyze directly, especially for complex audio classification tasks. It is a one-dimensional time-domain representation of sound.

- **STFT (Short-Time Fourier Transform):** STFT is a technique that dissects the audio signal into its constituent frequencies over small time windows. This results in a two-dimensional representation, with time on one axis and frequency on the other. STFT provides insights into how the spectral content of the audio changes over time, making it more suitable for audio analysis tasks.

3.2. STFT Parameters

When applying STFT to our audio data, we consider the following key parameters:

Window Size (‘n fft‘): This parameter defines the size of the analysis window used in the transformation. It determines the number of samples in each window and plays a crucial role in balancing time and frequency resolution. A larger window size provides finer frequency resolution but may result in coarser time resolution.

Hop Length (‘hop length‘): The hop length controls the overlap between consecutive windows and specifies how much the analysis window shifts along the time axis. A smaller hop length leads to increased overlap, which can enhance time resolution.

Window Function: To address spectral leakage and enhance the accuracy of the transformation, we apply a window function (Hanning is used by default) to each analysis window.

By meticulously selecting these STFT parameters, we can extract informative spectrogram features from our audio data, which serve as the input to our classification model.

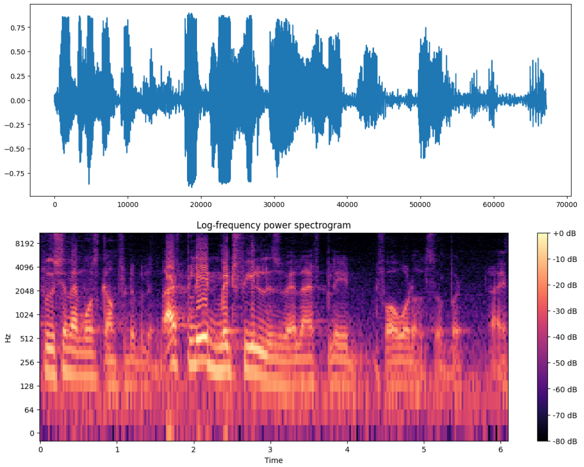

Figure 2. Original Audio Input (Top) and STFT Transform (Bottom)

3.3. STFT Parameters

When applying STFT to our audio data, we used the parameters shown in Table 2.

It is worth noting that the performance can change drastically depending on these values. For example, choosing a small window size will limit the models ability as it didn’t ”hear enough” to make a good classification. This can be seen clearly in Table 3 and Figure 4.

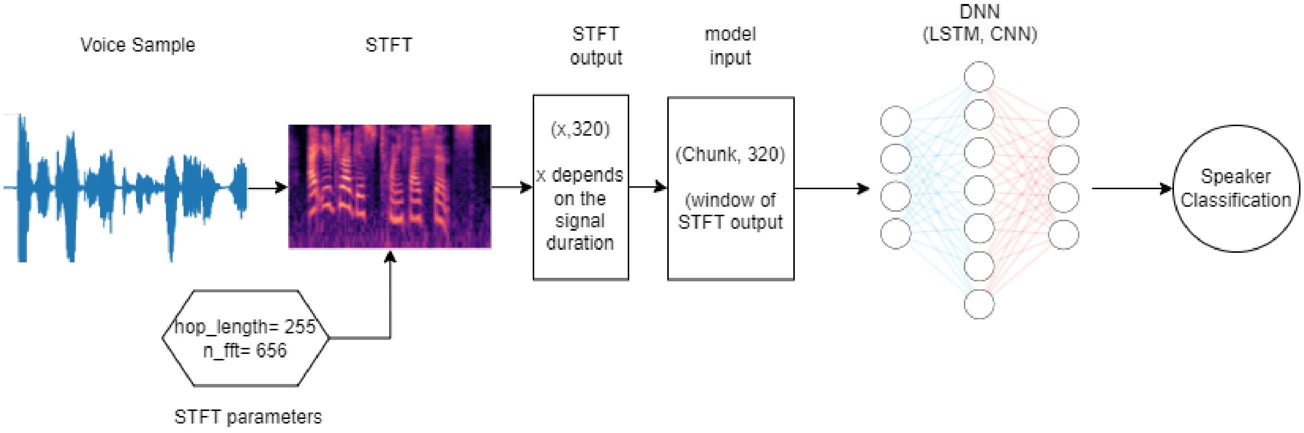

Figure 3. Workflow of the end-to-end classification model

Table 2. STFT chosen parameters values

Parameter	Value
Hop Length (‘hop length‘)	255
Window Size (‘n fft‘)	656
Window Length (‘win length‘)	656

Table 3. Accuracy result based on STFT parameters

height**Parameter**	**Ex1**	**Ex2**
hop_length	128	512
n_fft	256	1024
Test Accuracy	79.68%	89%

We see in Table 3 and Figure 4 how increasing window size and number of points (n fft) results is more coherent segments of frequency. This in turn increases the accuracy (+10% in this example). Another important parameter is the window size of the STFT to be used per forward pass. I am calling this window ”chunk size” to distinguish it from other window sizes in this work. Using a bigger chunk size increases the performance considerably up to a certain size (100 in my case). This is illustrated in Table 4 .

Table 4. Chunk Size Experiment Results (#classes=20)

height**Chunk Size**	**Value**	**Test Accuracy**
20	20	83%
100	100	89%
200	200	85%

We see here that there is a considerable difference (+5%) depending on the chunk size. The value might be different for more classes, but from the current data, 100 is a good value to start with.

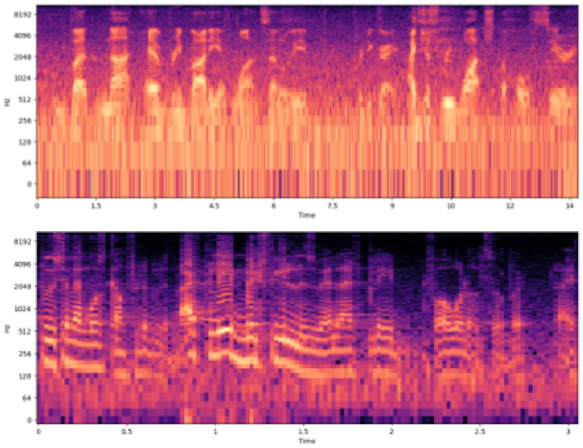

Figure 4. Ex1 (Top) Ex2 (Bottom)

Another important parameter is the number of audio files per speaker (duration of exposure to speaker) and the resultant accuracy. Table 5 shows the performance of the LSTM model according to the number of input files.

Table 5. Audio Files per Speaker Experiment Results (#classes=20)

# Audio Files per Speaker	**Value**	**Test Accuracy**
20	20	70.25%
40	40	78.12%
Up to 80	Up to 80	89%

As Table 5 . shows, a significant performance increase is realized when increasing the data per class. This is not very surprising of course as it is common knowledge that more data means better results, but it is highly important when the number of data files is low to begin with. To put things into perspective, the duration of many of the files in VoxCeleb is nearly 4 seconds, so 20 files result in a couple of minutes.

at most. It is impressive also at the same time to notice the strength of the deep learning approach.

4. Model architecture

There are numerous options to choose from for this task. One particular choice that is commonly used for time series data is RNN, so we decided to try it out initially along with CNN. Since the vanilla RNN sufferers from the vanishing/exploding gradient problem, we went with LSTM, a main variant of RNN (Schuster & Paliwal, 1997). Specifically, we went ahead with Bidirectional LSTM after some experimentation. Additionally, a 1D CNN ResNet based model has also been implemented and tested to the LSTM one. The Implementation is straightforward overall and Table 6. shows our LSTM model, while Table 7. shows our CNN model. A lot of experimentation on the model size has been done especially for the LSTM, but further optimization for the CNN should be done as future work to perform better on the more difficult dataset (VoxCeleb).

BiLSTMClassifier
Layers
LSTM Layer
Input Size: STFT output (ex: 329) Hidden Size: 500 Number of Layers: 2 Batch First: True Dropout: 0.5 Bidirectional: True
Fully Connected Layer
Input Features: 1000 Output Features: num-classes (ex: 201) Bias: True
Attention Layer
Attention Sub-Layer
Input Features: 1000 Output Features: 1 Bias: True

Table 6. BiLSTM Classifier Architecture

5. Performance Results

To start the work, it is better to try to get good performance on a small number of classes then try on a larger set. The results in the performance table show that both datasets saw a good performance on 20 classes, especially the LSTM one where it reached above 97% . We then tried the models on more classes and for Librispeech, the performance is solid even when increasing the number of classes up to 251, which train-100 (referring to 100 hours) in the dataset.

Table 7. Speaker Classifier CNN Model Summary

Layer	**Description**
`conv1`	Conv1D (100 → 16 channels, 3x1 kernel, no bias)
`bn1`	BatchNorm1D (16 channels, eps=1e-05, momentum=0.1, affine, track stats)
`relu`	ReLU Activation (in-place)
`layer1`	Sequential (BasicBlock: 16 → 32 → 32 channels)
`layer2`	Sequential (BasicBlock: 32 → 64 → 64 channels)
`layer3`	Sequential (BasicBlock: 64 → 128 → 128 channels)
`fc1`	Linear (10624 → 128, with bias)
`fc2`	Linear (128 → 64, with bias)
`fc3`	Linear (64 → 201, with bias)

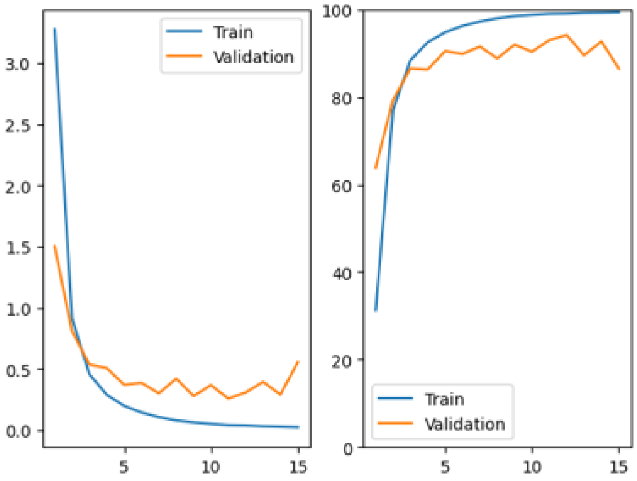

Figure 5. CNN 251 Librispeech accuracy

CNN Figure 5 (LR=1e-04, Batch size= 32) shows a solid performance where accuracy is 94%. The learning is also relatively stable but can be made more stable with different batch sizes. From my experiments on the LSTM model (for classes=20) having a much bigger batch size resulted in faster but less accurate results and I expected similar results for the CNN. The LSTM Figure 5 is also showing a great performance of 97%. The model is also more stable, which depends on the choice of LR and other training parameters (LR= 1e-03). Table 8 shows a full detailed comparison of the results.

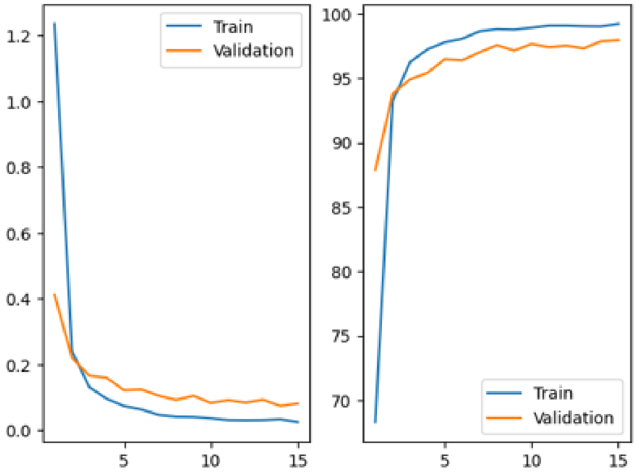

Figure 6. LSTM 251 Librispeech accuracy

In contrast to Librispeech, the VoxCeleb dataset is a harder one, and the results of both models reflect that. The CNN model under-performs for this dataset and it needs much more refinement. This is reflected by the gradients Figure 7 and we see that the loss starts increasing (instead of decreasing) after 12 epochs. The training accuracy seems stable, which can be an indication of the need for further regularization. It is notable the drop-out of 0.5 was used for both models. However, the LSTM model is solid overall. We see accuracy degradation with more classes, which is natural, but the final performance is still good (81% on 200 classes).

A comparison with a published paper (Anand et al., 2019) is provided in Table 8 . Our results outperform the best model in the paper using less than half the number of parameters. This shows that the model is both good performing and efficient. We also see that the CNN model is much smaller than the LSTM and it is one of the reasons it did not perform as good in comparison. It is notable that a similar size of data and classes between the two datasets has such a big difference in performance. The size of 250 classes in Librispeech is 6.21 Gb whereas the same size is only 220

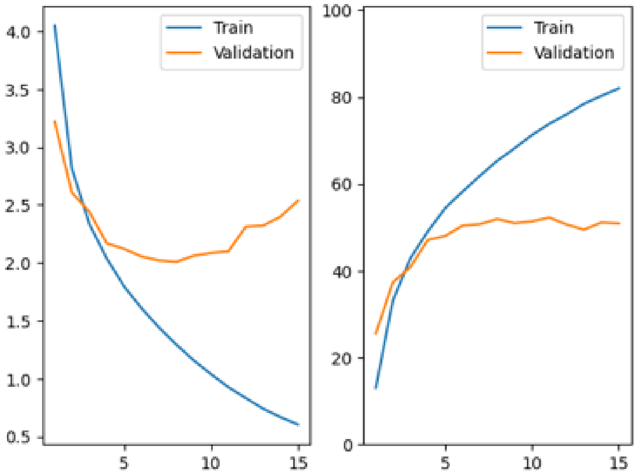

Figure 7. CNN 200 VoxCeleb1 accuracy

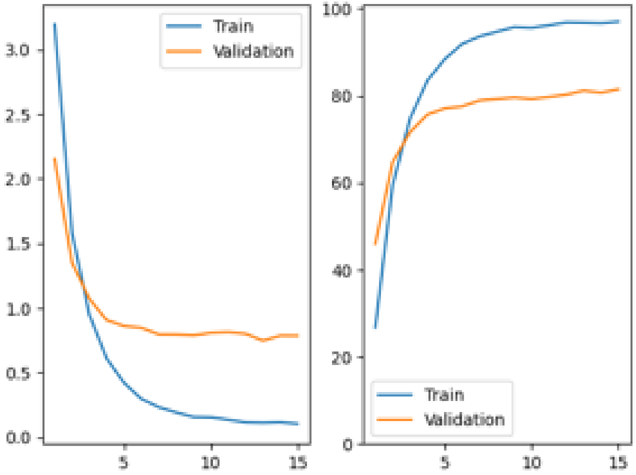

Figure 8. LSTM 200 VoxCeleb1 accuracy

Table 8. Model Performance Comparison

height**Model**	**# Size**	**# Classes**	**LSTM (ours)**	**CNN (ours)**	**(Anand et al., 2019) (VGG-M)**	**(Anand et al., 2019) (ResNet-34)**
NP	-	-	9,534,202	1,495,017	8,291,634	22,354,162
Voxceleb	689 Mb	20	89.00%	81.67%	-	-
Voxceleb	1.5 Gb	50	82.37%	73.47%	76.67	90.37
Voxceleb	5.7 Gb	200	81.47%	52.27%	58.63	71.48
Librispeech	171 Mb	20	97.84%	96.84%	-	-
Librispeech	348 Mb	40	95.55%	96.95%	-	-
Librispeech	6.21 Gb	250	97.94%	94.00%	-	-

classes in VoxCeleb (not shown in the Table 1). There is more than 15% difference in performance in the largest tried number of classes in both. In fact, the performance in Librispeech was relatively the same regardless of the number of classes (both CNN and LSTM). This further illustrates VoxCeleb dataset difficulty level.

6. Conclusion

The project discussed speaker classification using deep learning. Two architectures has been followed: BiLSTM, CNN. Both showed good performance on the task but the BiLSTM was doing better according to the chosen parameters. The approach steps were discussed including the preprocssing and frequency transform. The results were plotted and discussed and compared to a published work where our model outperformed it in both efficiency (less than half the number of parameters for BiLSTM) and accuracy on 2 sub-classes of VoxCeleb (50, 200 classes).

7. Limitations and Future Work

While the work overall shows promising results, further refinement is needed. More importantly, it is interesting to see the performance against state-of-the-art (SOTA) models on the whole dataset. Although a comparison with an available paper was provided (they also worked on subsection of the dataset VoxCeleb1) and our results outperformed their biggest model (Anand et al., 2019) (ResNet-34), it is not enough to prove the model performance on the difficult datasets. Furthermore, other datasets for other languages like Chinese, Arabic, etc can show a different challenge.

There are numerous things that are important to do for the continuation of this project.

- Try the full datasets. This goes for both VoxCeleb and LibriSpeech. This is computationally expensive, so should be done gradually like 100 classes, 500, full size.
- Experiment more with preprocessing like frequency transformation (MFCC, and others), noise, and filtering. Add more noise to LibriSpeech (and try its less clean test-sets), and try more filters and different preprocessing approaches for VoxCeleb1.
- Control for more parameters. While we tried and compared performance under different numbers of files and data subset sizes, we can try more parameters and check the performance.
- Further investigate the results. It is also interesting to see if there is a pattern for when the model fails. Like more noise in a specific class, unclear speaker, accuracy per gender, nationality, accent, etc. The subsample chosen was random, so it is expected to be reflective, but the data is always better.
- Multi-layer classification. In this work, we tried to classify the speakers directly, but gender, age, and other attributes can be used simultaneously.
- Other types of voice classification like text-dependent, multi-talkers, and overlapping talkers are also interesting.
- Doing voice generation and comparing with the available classes. GANs can be a viable approach as it has both a generator and a discriminator.

8. Acknowledgments

This work was done individually by only one author.

References

Mozzila Common Voice. https://commonvoice.mozilla.org/en/datasets.

OpenSLR Resources. https://www.openslr.org/resources.php.

Linguistic Data Consortium, University of Pennsylvania. https://catalog.ldc.upenn.edu/LDC2004S13, 2004.

Librispeech: An asr corpus based on public domain audio books. volume 2015-August, 2015. doi: 10.1109/ICASSP.2015.7178964.

Alashban, A. A. and Alotaibi, Y. A. Speaker gender classification in mono-language and cross-language using blstm network. 2021. doi: 10.1109/TSP52935.2021.9522623.

An, N. N., Thanh, N. Q., and Liu, Y. Deep cnns with self-attention for speaker identification. *IEEE Access*, 7, 2019. ISSN 21693536. doi: 10.1109/ACCESS.2019.2917470.

Anand, P., Singh, A. K., Srivastava, S., and Lall, B. Few shot speaker recognition using deep neural networks. 04 2019.

Bu, H., Du, J., Na, X., Wu, B., and Zheng, H. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. 2018. doi: 10.1109/ICSDA.2017.8384449.

Cai, W., Chen, J., and Li, M. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system, 2018.

Chung, J. S., Nagrani, A., and Zisserman, A. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19, 2011. ISSN 15587916. doi: 10.1109/TASL.2010.2064307.

Ding, S., Chen, T., Gong, X., Zha, W., and Wang, Z. Autospeech: Neural architecture search for speaker recognition. volume 2020-October, 2020. doi: 10.21437/Interspeech.2020-1258.

El-Moneim, S. A., Nassar, M. A., Dessouky, M. I., Ismail, N. A., El-Fishawy, A. S., and El-Samie, F. E. A. Text-independent speaker recognition using lstm-rnn and speech enhancement. *Multimedia Tools and Applications*, 79, 2020. ISSN 15737721. doi: 10.1007/s11042-019-08293-7.

Hu, Z. F., Si, X. T., Luo, Y., Tang, S. S., and Jian, F. Speaker recognition based on 3dcnn-lstm. *Engineering Letters*, 29, 2021. ISSN 18160948.

India, M., Safari, P., and Hernando, J. Self multi-head attention for speaker recognition. volume 2019-September, 2019. doi: 10.21437/Interspeech.2019-2616.

Kabir, M. M., Mridha, M. F., Shin, J., Jahan, I., and Ohi, A. Q. A survey of speaker recognition: Fundamental theories, recognition methods and opportunities. *IEEE Access*, 9, 2021. ISSN 21693536. doi: 10.1109/ACCESS.2021.3084299.

Li, L., Chen, Y., Shi, Y., Tang, Z., and Wang, D. Deep speaker feature learning for text-independent speaker verification. volume 2017-August, 2017. doi: 10.21437/Interspeech.2017-452.

Nagraniy, A., Chungy, J. S., and Zisserman, A. Voxceleb: A large-scale speaker identification dataset. volume 2017-August, 2017. doi: 10.21437/Interspeech.2017-950.

Niizumi, D., Takeuchi, D., Ohishi, Y., Harada, N., and Kashino, K. Masked modeling duo: Learning representations by encouraging both networks to model the input, 2023.

PaperswithCode. Speaker identification on voxceleb1, 2023. URL https://paperswithcode.com/sota/speaker-identification-on-voxceleb1. Accessed: 17 December 2023.

Paul, D. B. and Baker, J. M. The design for the wall street journal-based csr corpus. 1992. doi: 10.3115/1075527.1075614.

Schuster, M. and Paliwal, K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45 (11):2673–2681, 1997. doi: 10.1109/78.650093.

Snyder, D., Garcia-Romero, D., Povey, D., and Khudanpur, S. Deep neural network embeddings for text-independent speaker verification. volume 2017-August, 2017. doi: 10.21437/Interspeech.2017-620.

Tran, V. T. and Tsai, W. H. Speaker identification in multi-talker overlapping speech using neural networks. *IEEE Access*, 8, 2020. ISSN 21693536. doi: 10.1109/ACCESS.2020.3009987.

Vaessen, N. and van Leeuwen, D. A. Fine-tuning wav2vec2 for speaker recognition. volume 2022-May, 2022. doi: 10.1109/ICASSP43922.2022.9746952.

Vazhenina, D. and Markov, K. End-to-end noisy speech recognition using fourier and hilbert spectrum features. *Electronics (Switzerland)*, 9, 2020. ISSN 20799292. doi: 10.3390/electronics9071157.

Zhang, C., Bahmaninezhad, F., Ranjan, S., Dubey, H., Xia, W., and Hansen, J. H. Utd-crss systems for 2018 nist speaker recognition evaluation. volume 2019-May, 2019. doi: 10.1109/ICASSP.2019.8683097.