The Speaker and Language Recognition Workshop (Odyssey 2022)
28 June - 1 July 2022, Beijing, China

Hybrid Neural Network-based Deep Embedding Extractors for Text-Independent Speaker Verification

Jahangir Alam, Woo Hyun Kang, Abderrahim Fathan

Computer Research Institute of Montreal (CRIM)
jahangir.alam,woohyun.kang,abderrahim.fathan@crim.ca

Abstract

In this contribution, we propose a multi-stream hybrid neural network for extracting speaker discriminant utterance-level embedding vectors. In this approach, an input acoustic feature frame is processed in multiple parallel pipelines where each stream has a unique dilation rate for incorporating diversity of temporal resolution in embedding processing. In order to aggregate the speaker information within short time-span and utterance-level context, proposed extractor employs multi-level global-local statistics pooling. In addition, we also propose an ensemble embedding extractor that employs both a hybrid neural network (HNN) and an extended time delay neural network - Long short-term memory (ETDNN-LSTM) hybrid modules for including diversified temporal resolution and for capturing complementarity. In order to evaluate the proposed systems, a set of experiments on the CNCeleb corpus were conducted, and the proposed multi-stream hybrid network outperformed the conventional approaches trained on the same dataset. The ensemble approach is found to provide the best results in terms of all considered evaluation metrics.

1. Introduction

Speaker verification is the task of verifying the claimed speaker identity based on the given speech samples and has become a key technology for personal authentication in various applications. Usually, utterance-level fixed-dimensional vectors (i.e., embedding vector) are extracted from the enrolled and test speech recordings and then fed into a scoring algorithm (e.g., cosine similarity) to measure their similarity.

To efficiently capture the speaker-dependent information from the given speech, a variety of methods have been proposed utilizing deep learning architectures for learning the latent frame-level representations, which are then aggregated to obtain an utterance-level embedding [1, 2]. One of the most popular methods is the x-vector framework [2], which uses a time-delay neural network (TDNN) architecture and statistics pooling for extracting a speaker discriminant utterance-level embedding. This framework has shown great performance in text-independent speaker verification [2].

The Extended TDNN (ETDNN) [3] and Factored TDNN (FTDNN) [4] architectures are the two extensions of x-vector framework proposed for improving speaker verification performance. For the past several years, there have been lots of attempts on employing the residual network (ResNet) architecture for speaker embedding extraction [5, 6], which have proven to be the dominant approach in the image classification field [7]. Moreover, to exploit the speaker-dependent information within the temporal variability of the speech sequence, many research also focused on employing a recurrent neural network such as the long short-term network (LSTM) for speaker embedding extraction [8, 9].

Although these methods have all shown reasonable performance in terms of speaker verification, most of them rely on a single network architecture (e.g., TDNN, ResNet, LSTM). However, different network architectures are known to learn complementary information about the input representation. For example, TDNN and Convolutional Neural Network (CNN) architectures are adept at reducing changes in frequency, while LSTMs are good for sequential & temporal modeling. On the other hand, DNN is suitable for mapping input features to more separable regions. To take advantage of the complementary speaker information encoded by different network architectures, various hybrid approaches have been proposed, which employ non-TDNN modules such as LSTM or CNN to the x-vector framework, and have achieved noticeable enhancements of the speaker verification performance. Especially in [10–12], it was observed that the robustness of x-vectors can be improved by adding residual connections between the frame-level layers. Moreover, [13] proposed a multi-level pooling scheme for considering the statistics from different modules (e.g., TDNN, LSTM), and showed promising performance in speaker verification.

In light of this, a large-scale hybrid neural network (HNN) was recently proposed for speaker embedding extraction [14, 15], which not only employs different types of network architectures (i.e., TDNN, 2D-CNN, LSTM), but also exploits the short-durational statistics of the hidden representations to capture the instantaneous speaker-dependent information. Attributed to these characteristics, the HNN system have shown promising results in various challenging datasets, including NIST Speaker Recognition Evaluation (NIST-SRE) 2016, Short duration Speaker Verification (SdSV) Challenge 2021 [16], and VoxCeleb [17].

In this paper, we aim to further expand the HNN framework to fully extract the speaker information latent in the input speech. More specifically, we propose two new HNN-based architectures for extracting the speaker embedding. Our first proposed model is the multi-stream hybrid neural network (MSHNN), which incorporates the multi-streaming framework into the HNN speaker embedding network. The multi-streaming scheme was originally adapted for automatic speech recognition (ASR) [18], where the network processes the input speech with multiple temporal resolutions by applying different dilation rates to the CNN. Since the speaker-dependent information can be latent in different temporal contexts, we can assume that the speaker embedding extraction network can greatly benefit from analyzing the multiple resolutions of the given speech. Our second proposed model is an ensembled HNN architecture, which employs both the standard HNN and an extended time delay neural network & Long short-term memory (ETDNN-LSTM)-based hybrid network for

10.21437/Odyssey.2022-5

further diversifying the complementarity of different network architectures. In order to evaluate the speaker verification performance of our proposed hybrid systems, we have conducted a set of experiments on the CNCeleb dataset [19, 20]. From our results, it could be seen that the proposed MSHNN can outperform the conventional approaches. Moreover, ensembled hybrid system showed the best performance. The contribution of this paper is as follows:

- We propose a novel embedding architecture for speaker verification, which incorporates the multi-streaming scheme into the HNN embedding extractor.
- We additionally propose an ensembled hybrid system for speaker verification, which employs two different hybrid backbone architectures namely, HNN and ETDNN-LSTM.
- We perform evaluation of our proposed approaches on the CNCeleb dataset and compare performances with the previously proposed approaches.
- To the best of our knowledge, proposed approach is new and till now, nobody has tried this approach for text-independent speaker verification task.

The rest of this paper is organized as follows: Section 2 provides a brief overview of x-vector framework and its extensions. The detailed description of the conventional HNN system is described in Section 3. The proposed systems are described in Section 4. In Section 5, detailed setting for the experimentation and the results are presented, and Section 6 concludes the paper.

2. X-vector Extractor and Its Extensions

In this section, we provide a brief description of the popular x-vector framework and some of its extensions which are widely used by the speaker recognition community for the speaker verification task.

The x-vector extractor framework [1], as depicted as Table 1, comprised of a time-delay neural network (TDNN)-based frame-level network (1–5 layers, also known as backbone architecture), statistics pooling layer which concatenates component-wise first- and second-order statistics over the time axis, two fully connected layers and output softmax layer to yield outputs corresponding to conditional log-probabilities over the set of training speakers. This framework demonstrated great performance in text-independent speaker verification [2] and other speech applications.

The Extended TDNN (ETDNN) was introduced in [3, 21] with wider temporal context than TDNN for improving speaker recognition performance. As summarized in Table 2 the extended TDNN architecture utilizes interleaving dense layers between TDNN layers and slightly wider temporal context due to the 7th TDNN layer.

The Factored TDNN (FTDNN) [22] architecture is another extension of the x-vector framework proposed for improving speaker verification performance. The FTDNN is obtained by simply replacing 2nd to 9th layers in the ETDNN architecture with a factorized TDNN with skip connections [22, 23]. In FTDNN, the weight matrix is factorized into the product of two low-rank matrices one of which is constrained to be semi-orthogonal [23].

The other simple improvement of x-vector was introduced in [24] where the 2nd TDNN layer in the x-vector architecture was replaced with a LSTM (Long Short-Term Memory) layer and demonstrated better speaker verification performance.

Table 1: TDNN-based x-vector extractor architecture. T indicates the duration of features in number of frames and d the feature vector dimensionality. The last column indicates the size of input and output in each layer.

Layer	*Layer Type*	*Context*	*Input* $\rightarrow$ *Output*
1	*TDNN-ReLU*	t-2:t+2	d $\times$ T $\rightarrow$ 512 $\times$ T
2	*TDNN-ReLU*	t-2,t,t+2	512 $\times$ T $\rightarrow$ 512 $\times$ T
3	*TDNN-ReLU*	t-3,t,t+3	512 $\times$ T $\rightarrow$ 512 $\times$ T
4	*Dense-ReLU*	t	512 $\times$ T $\rightarrow$ 512 $\times$ T
5	*Dense-ReLU*	t	512 $\times$ T $\rightarrow$ 1500 $\times$ T
6	*Pooling (mean + stddev)*	t	1500 $\times$ T $\rightarrow$ 3000
7	*Dense-ReLU*		3000 $\rightarrow$ 512
8	*Dense-ReLU*		512 $\rightarrow$ 512
9	*Softmax*		512 $\rightarrow$ # speakers

Table 2: Improved x-vector extractor architecture based on the extended TDNN (ETDNN) backbone (1-9 layers). T indicates the duration of features in number of frames and d the feature vector dimensionality. The last column indicates the size of input and output in each layer.

Layer	*Layer Type*	*Context*	*Input* $\rightarrow$ *Output*
1	*TDNN-ReLU*	t-2:t+2	d $\times$ T $\rightarrow$ 512 $\times$ T
2	*Dense-ReLU*	t	512 $\times$ T $\rightarrow$ 512 $\times$ T
3	*TDNN-ReLU*	t-2,t,t+2	512 $\times$ T $\rightarrow$ 512 $\times$ T
4	*Dense-ReLU*	t	512 $\times$ T $\rightarrow$ 512 $\times$ T
5	*TDNN-ReLU*	t-3,t,t+3	512 $\times$ T $\rightarrow$ 1500 $\times$ T
6	*Dense-ReLU*	t	512 $\times$ T $\rightarrow$ 512 $\times$ T
7	*TDNN-ReLU*	t-4,t,t+4	512 $\times$ T $\rightarrow$ 512 $\times$ T
8	*Dense-ReLU*	t	512 $\times$ T $\rightarrow$ 512 $\times$ T
9	*Dense-ReLU*	t	512 $\times$ T $\rightarrow$ 1500 $\times$ T
10	*Pooling (mean + stddev)*	t	1500 $\times$ T $\rightarrow$ 3000
11	*Dense-ReLU*		3000 $\rightarrow$ 512
12	*Dense-ReLU*		512 $\rightarrow$ 512
13	*Softmax*		512 $\rightarrow$ # speakers

3. Hybrid Neural Network (HNN) Embeddings Extractor

In this section, we describe the backbone architecture for the HNN embedding extractor [14, 15]. An overview of the HNN embedding extractor is presented in Figure 1 that employs CNN, TDNN, LSTM networks and global-local statistics pooling layers. The key motivation behind using hybrid networks in numerous speech processing applications is to catch the complementary information that exists among CNN, LSTM, TDNN, and DNN modules. In Figure 1, the HNN backbone architecture is depicted inside the red dotted rectangle.

3.1. 2D-CNN-based feature extraction module

In order to make sure that the hybrid network can capture the temporal-spectral correlations within the speech, the HNN uses 2D-CNNs to process the input Mel-FilterBank (MFB) features over which SpecAugment [25, 26] is applied on the fly, where both time and frequency masking are performed. By passing the input augmented MFB features (after applying SpecAugment) through a stack of 5 2D-CNN layers [18], frame-level representations with information on not only the relation between the local frames, but also the local frequency bins could be obtained.

3.2. TDNN-LSTM-based frame-level network

The 2D-CNN module is then followed by a frame-level network which is composed of TDNN and LSTM layers, to extract local descriptors with sufficient temporal information for speaker discrimination. In Figure 1, the TDNN-LSTM-based frame-level

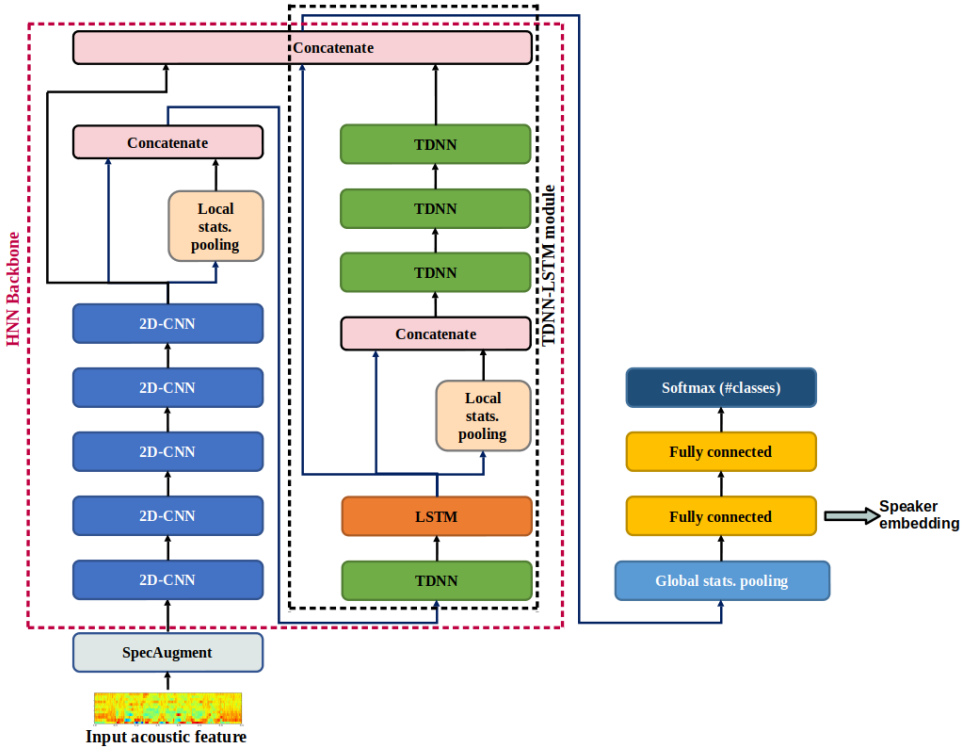

Figure 1: Schematic diagram of the hybrid neural network (HNN) architecture as embeddings extractor for automatic speaker verification task [14,15]. The HNN backbone architecture is shown inside the red dotted rectangle. Inside the black dotted rectangle the TDNN-LSTM module is presented. Here, the acronyms CNN, TDNN and LSTM stand for Convolutional Neural Network, Time Delay Neural Network and Long Short-Term Memory, respectively.

network is shown inside the black dotted rectangle. The frame-level network used in the HNN is similar to the TDNN-LSTM approach presented in [24], where the second TDNN layer of the standard x-vector [2] is replaced with a LSTM layer.

3.3. Multi-level global-local statistics pooling

In the HNN architecture, a multi-level statistics pooling (MLSP) [27] was employed for aggregating statistics from the last layers of CNN, LSTM and TDNN blocks in order to capture speaker specific information from different spaces and learn more discriminative utterance-level representations by bagging complementarity available in CNN, LSTM and TDNN networks. Similar to the standard x-vector, HNN extracts the first- and second-order statistics. However, unlike the conventional x-vector, HNN extracts the statistics not only globally, but also locally to exploit the short-durational correlation. While the global statistics pooling is done in the same manner with the standard x-vector, the local statistics pooling is performed within a short durational moving window similarly to the speech activity detection proposed in [28]. Each module (i.e., TDNN, LSTM) takes both the frame-level outputs from the previous model, and the local statistics extracted from them as input. During the local statistics pooling operation, the input sequences are resampled and the pooling window shift rates are adjusted to match the sequence length with the frame-level features.

After propagating the input features to the frame-level network, a global statistics pooling is performed to aggregate the local descriptors obtained from the TDNN and LSTM blocks. The global first- and second- order statistics are concatenated to a fixed-dimensional utterance-level representation.

The pooled statistics are then projected into a 512-dimensional embedding vector via two fully-connected layers. Once the training is completed, the embeddings are extracted from the fully-connected layer close to the global statistics pooling layer.

4. Proposed methods

This section provides a detailed description of the proposed embedding extractor frameworks.

4.1. Multi-Stream Hybrid Neural Network (MSHNN) Embeddings Extractor

In this section, we describe the proposed Multi-Stream Hybrid Neural Network (MSHNN) Embeddings Extractor architecture. As presented in Figure 2, the proposed MSHNN system follows a similar backbone structure with the conventional HNN, where the network is composed of 2D-CNN, TDNN-LSTM, and TDNN blocks. However, unlike the standard HNN, which only consists of one TDNN-LSTM block, the MSHNN employs multiple TDNN-LSTM blocks to capture the speaker information latent in different temporal resolutions. More specifically, after processing the input acoustic feature with 5 layers of 2D-CNN layers, the CNN output along with its local statistics are branched out to 3 different streams, where each stream process consists of a TDNN-LSTM block with a unique dilation rate. The outputs from the different streams are then concatenated to each other, and then fed into the following TDNN layers as in the standard HNN framework.

Like the HNN architecture, a multi-level statistics pooling

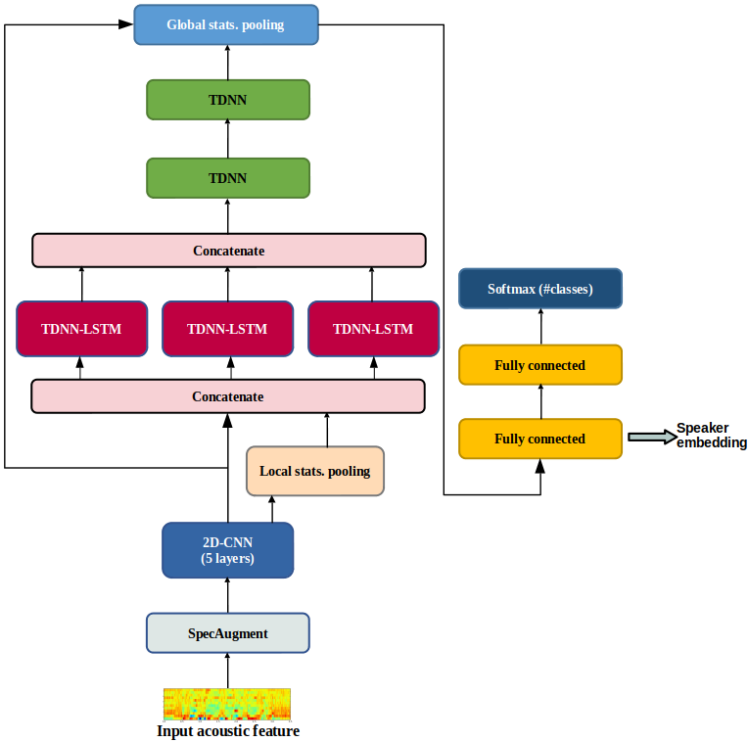

Figure 2: Schematic diagram of the Multi-Stream Hybrid Neural Network (MSHNN) architecture as embeddings extractor for automatic speaker verification task. Our proposed MSHNN embedding extractor employs three HNN backbone architectures in parallel, each with unique dilation rate for incorporating diversity of temporal resolution while processing embeddings. The acronyms CNN, TDNN and LSTM stand for Convolutional Neural Network, Time Delay Neural Network and Long Short-Term Memory, respectively.

(MLSP) [27] is also employed in the MSHNN framework for pooling statistics from the last layers of CNN, TDNN blocks, as shown in Figure 2, to capture speaker specific information from different spaces and learn more discriminative utterance-level representations by bagging complementarity available in different networks. The global first- and second- order statistics are concatenated to obtain fixed-dimensional utterance-level representations which are then projected into a 512-dimensional embedding vector via two fully-connected layers. When training is completed, the embeddings are extracted from the fully-connected layer adjacent to the global statistics pooling layer.

4.2. The Ensemble Embeddings Extractor

In this section, we describe the proposed ensemble embedding extractor architecture, denoted as ENSEMBLE, that incorporates two hybrid backbone networks in parallel manner. These two backbone networks are – The HNN backbone described in Section 3 and the extended TDNN-LSTM (ETDNN-LSTM) backbone architectures.

As shown in Figure 3, the proposed embedding extractor architecture ensembles two different hybrid architectures:

- Standard HNN: which is the conventional HNN backbone, the description of which has already been provided in Section 3. In Figure 3, this standard HNN backbone is highlighted using a red dashed rectangle.
- ETDNN-LSTM_CA: This hybrid backbone is similar to the standard HNN, but is based on the extended TDNN (ETDNN) [3] and does not employ any 2D-CNN layers. From the ETDNN architecture, which is known to have a wider temporal context than the standard TDNN, the second layer was replaced with the LSTM layer and the local statistics are appended as in the HNN. Moreover, unlike the standard HNN (shown in Figure 1) which applies SpecAugment to the Mel FilterBank (MFB) features, the ETDNN-LSTM_CA applies masking directly to the MFCC and we denote this augmentation as CepAugment.

While the standard HNN can capture the time-frequency correlation via the 2D-CNN module, the ETDNN-LSTM_CA system is expected to focus more on the temporal information attributed to the large number of TDNN layers. Therefore, the proposed ensemble architecture may exploit the complementary information captured by these two different hybrid architectures.

The global multi-level statistics pooling is performed from the last layers of the two hybrid backbone architectures, as shown in Figure 3, to capture speaker specific information from different modules and learn more discriminative utterance-level representations by bagging complementarity available in these parallel backbone networks.

The global first- and second- order statistics are concatenated to obtain fixed-dimensional utterance-level representations which are then projected into a 512-dimensional embedding via two fully-connected layers. The embeddings are normally extracted from the fully-connected layer near the global statistics pooling layer.

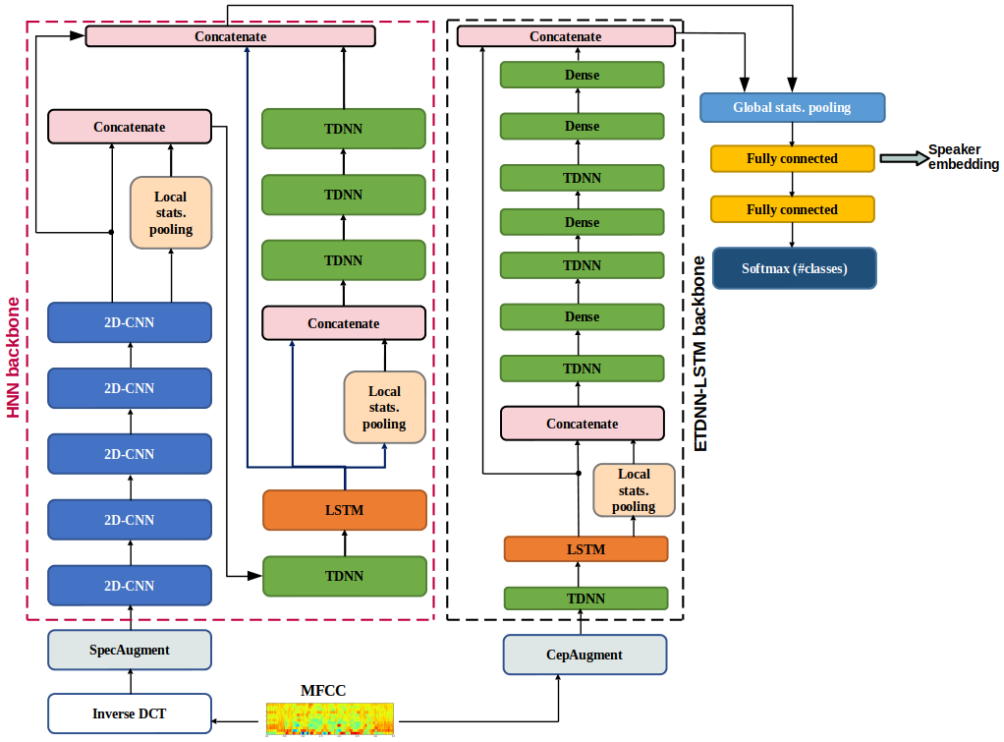

Figure 3: Schematic diagram of the ensemble embeddings extractor for automatic speaker verification task. Here, we denote this system as ENSEMBLE. This embedding extractor employs two hybrid backbone architectures, namely the HNN (hybrid neural network) backbone and the extended TDNN-LSTM (ETDNN-LSTM) backbone, in parallel fashion. In this Figure the HNN backbone module is marked by the red dashed rectangle and inside the black dashed rectangle is the ETDNN-LSTM backbone architecture.

5. Experiments

5.1. Dataset

In order to carry out simulation experiments, we use CNCeleb corpus [19, 20] which is comprised of CN-Celeb 1 [19] and CN-Celeb 2 [20] subsets and sampled at 16kHz with 16-bit precision. Statistics of CNCeleb corpus is presented in Table 3 in terms of train - eval splits, number of speakers, recordings and evaluation trials.

For training our developed systems, we have used the training portions of the CNCeleb 1 & 2 datasets (i.e., CNCeleb_train as presented in Table 3), which consists of approximately 2,800 speakers with 11 different genres (i.e., advertisement, drama, entertainment, interview, live broadcast, movie, play, recitation, singing, speech, vlog) [20]. For a detailed information about the CNCeleb dataset please see [19, 20]. The evaluation subset of the CN-Celeb 1 [19] dataset is used for reporting results.

5.2. Offline Data Augmentation

The use of data augmentation in deep learning-based classification task is ubiquitous. Data augmentation helps to increase the size and diversity in the training data. It also helps the network to achieve better generalization capability to unseen data. In order to increase the robustness and generalization capability of the embedding extraction network, multiple offline and on the fly data augmentation techniques are applied before being fed into the network. The offline data augmentation (on waveform-level) generates supplementary data using the following strategies:

- Reverb: Artificially reverberate via convolution with simulated RIRs from the AIR dataset[1].
- Music: A single music file (without vocals) is randomly selected from MUSAN[2], trimmed or repeated as necessary to match duration, and added to the original signal (5–15 dB SNR).
- Noise: MUSAN noises are added at one second intervals throughout the recording (5–15 dB SNR).
- Babble: Three to seven speakers are randomly picked from MUSAN speech data, summed together, then added to the original signal (13-20 dB SNR).

For all the trained systems, we commonly apply above mentioned offline augmentation to the input speech prior to the MFCC extraction process [2]. So, all our developed systems use the augmented CNCeleb_train data (original CNCeleb_train + supplementary data generated over CNCeleb_train).

5.3. Acoustic Features and Backend

40-dimensional Mel-frequency cepstral coefficients (MFCC) are extracted using an analysis window of 25 msec with a frame shift of 10 msec. Features are normalized using cepstral mean normalization over a window of 300 frames. In order to get rid of non-speech frames, energy-based speech activity detector (SAD) is used. To perform verification scoring, a Probabilistic linear discriminant analysis (PLDA) backend is used following kaldi recipe[3].

[1] https://www.openslr.org/20/
[2] https://www.openslr.org/17/
[3] https://github.com/kaldi-asr/kaldi/tree/master/egs/cnceleb/v2

5.4. Evaluation Metrics

The equal error rate (EER) and minimum detection cost functions minDCF (p-target=0.01) & minDCF (p-target=0.001) are used as metrics to evaluate the performances of the proposed and other conventional methods considered in this work for comparison purpose.

5.5. Experimental setup

Although offline data augmentation based on the noise and reverberation augmentation was proven to be effective in previous researches, the performance gain may be limited since the network may overfit to the predefined noise and room impulse response (RIR) sets used for augmentation.

Therefore in addition to the typical noise and RIR augmentation, we have explored the use of the SpecAugment technique [26], which has shown great performance in speech recognition. Our developed systems ETDNN_CA, ETDNN-LSTM_CA, HNN, MSHNN and ENSEMBLE (as presented in Table 4) use both offline and on-the-fly data augmentation using SpecAugment technique [25].

Since SpecAugment is applicable on the Mel-FilterBank (MFB) features, the MFCC features are passed through the IDCT-layer (inverse discrete cosine transform) to obtain d-dimensional Mel-Filterbank (MFB) features. Here $d = 40$. The MFCC features, being decorrelated, are more easily compressible without any information loss and therefore, take less storage space than MFB features.

Over the MFB features, SpecAugment is applied on the fly, where both time and frequency masking are performed. For a MFB feature sequence with n frames, the policy for time and frequency masking are as follows:

- Frequency masking: for a randomly sampled $f \sim \text{unif}(0, F)$ and $f_0 \sim \text{unif}(0, d - f)$, the Mel-frequency channels $[f_0, f_0 + f)$ are masked, where F is the frequency mask parameter.
- Time masking: for a randomly sampled $t \sim \text{unif}(0, T)$ and $t_0 \sim \text{unif}(0, n - t)$, the time steps $[t_0, t_0 + t)$ are masked, where T is the time mask parameter.

SpecAugment is known to prevent deep learning models from being overfit thus enable them to become more robust to unseen testing data [25]. Though SpecAugment augmentation is normally applied on spectral features (e.g., Mel FilterBank) the ETDNN_CA, ETDNN-LSTM_CA, and ENSEMBLE systems apply masking directly to the MFCC features and in this work, we denote this augmentation as CepAugment.

All systems are trained on the augmented CNCeleb_train data. All reported results in Table 4 are on the CNCeleb1_eval data consisting of 17755 target and 3466537 non-target trials.

Implementation of all experimented systems was performed using the Kaldi toolkit [29] following the egs/cnceleb/v2 speaker verification recipe.

5.6. Experimental results

In this experiment, we compare the speaker verification performance of different HNN-based systems and other conventional systems on the CNCeleb dataset. Table 4 shows the results of the experimented systems. As shown in the results from E-TDNN and TDNN, increasing the temporal context can significantly improve the performance. Using CepAugment with the ETDNN (i.e., ETDNN_CA) further improved the performance, which achieved an EER of 11.09%.

Table 3: Statistics of CNCeleb 1 & 2 data in terms of numbers of speakers, recordings, total number of trials and target trials.

Train/Test sets	# Speakers	# Recordings	# Trials	# Target trials
CNCeleb1_train	797	107953	N/A	N/A
CNCeleb2_train	1996	524787	N/A	N/A
CNCeleb_train (1 & 2)	2793	632740	N/A	N/A
CNCeleb1_Eval	200	17973	3484292	17755

The HNN and the ETDNN-LSTM_CA systems outperformed the conventional TDNN-based architectures (i.e., TDNN, ETDNN). Especially the standard HNN outperformed the conventional TDNN with a relative improvement of 25.95% in terms of EER. This may be attributed to the complementary information learned by different network architectures (e.g., LSTM, TDNN, CNN). Moreover, from these results, we can assume that capturing the time-frequency correlation and increasing the temporal context can both improve the embedding networks ability to capture the speaker discriminative footprints.

The best performance was observed from the ensemble system (i.e., ENSEMBLE), which outperformed the standard HNN with a relative improvement of 2.58% in terms of EER. This indicates that the standard HNN and the ETDNN-LSTM_CA can learn complementary speaker-dependent information to each other.

Interestingly, although the proposed MSHNN is not ensembled with other network architectures, it was also able to achieve comparable performance to the ENSEMBLE system. By only adding a multi-streaming scheme, the MSHNN was able to outperform the HNN with a relative improvement of 1.36% in terms of EER. Such performance improvement may be attributed to the capability of the MSHNN to extract the speaker information latent in different temporal resolutions.

6. Conclusion

In this work, we proposed two new hybrid neural network (HNN)-based speaker embedding extraction schemes for effectively extracting the speaker-dependent information from the input speech. We first proposed a multi-stream HNN (MSHNN), which processes the input speech in multiple streams of TDNN-LSTM blocks to capture the speaker information latent in different temporal resolutions. Additionally, we proposed an ensemble embedding extractor, which combines two different hybrid backbone architectures (HNN and ETDNN-LSTM) to exploit the complementary information learned by both of them. Our experimental results on the CNCeleb corpus showed that both of our proposed methods can outperform the conventional methods.

Our future study will further explore the proposed HNN-based systems to analyze their performance on larger datasets and incorporate training strategies for training the systems. Moreover, we will investigate the combination of the two proposed methods, which will ensemble the multi-stream and single-stream HNN-based systems.

7. Acknowledgment

The authors wish to acknowledge the funding from the Government of Canada's New Frontiers in Research Fund (NFRF) through grant NFRFR-2021-00338 and Natural Sciences and Engineering Research Council of Canada (NSERC) through grant RGPIN-2019-05381. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NFRF & NSERC.

Table 4: The experimental results of the proposed Hybrid Neural Network (HNN) -based systems and conventional methods on the CNCeleb1 evaluation sets in terms of EER (equal error rates) and minimum detection cost functions minDCF (p-target=0.01) & minDCF (p-target=0.001). The best results of each evaluation metric are highlighted in bold font.

System	Backend	EER	minDCF (p-target=0.01)	minDCF (p-target=0.001)
TDNN [20]	PLDA	12.39	0.6011	0.7353
TDNN	PLDA	11.33	0.5704	0.7200
ETDNN	PLDA	11.09	0.5626	0.7190
ETDNN_CA	PLDA	10.88	0.5588	0.7105
ETDNN-LSTM_CA	PLDA	10.30	0.5459	0.6828
HNN	PLDA	9.18	0.4994	0.6441
MSHNN	PLDA	**9.05**	**0.4874**	**0.6425**
ENSEMBLE	PLDA	**8.94**	**0.4822**	**0.6389**

8. References

[1] Yunqi Cai, Lantian Li, Dong Wang, and Andrew Abel, "Deep Speaker Vector Normalization with Maximum Gaussianality Training," 2020.

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

[3] David Snyder, Jesús Villalba, Nanxin Chen, Daniel Povey, Gregory Sell, Najim Dehak, and Sanjeev Khudanpur, "The JHU Speaker Recognition System for the VOiCES 2019 Challenge," in *Proc. Interspeech 2019*, 2019, pp. 2468–2472.

[4] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Fred Richardson, Suwon Shon, François Grondin, Réda Dehak, Leibny Paola García-Perera, Daniel Povey, Pedro A. Torres-Carrasquillo, Sanjeev Khudanpur, and Najim Dehak, "State-of-the-Art Speaker Recognition for Telephone and Video Speech: The JHU-MIT Submission for NIST SRE18," in *Proc. Interspeech 2019*, 2019, pp. 1488–1492.

[5] Tianyan Zhou, Yong Zhao, and Jian Wu, "ResNeXt and Res2Net Structures for Speaker Verification," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 301–307.

[6] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[8] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, p. 5115–5119, IEEE Press.

[9] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.

[10] Sergey Novoselov, Andrey Shulipa, Ivan Kremnev, Alexander Kozlov, and Vadim Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition," *CoRR*, vol. abs/1804.10080, 2018.

[11] Aleksei Gusev, Vladimir Volokhov, Tseren Andzhukaev, Sergey Novoselov, Galina Lavrentyeva, Marina Volkova, Alice Gazizullina, Andrey Shulipa, Artem Gorlanov, Anastasia Avdeeva, Artem Ivanov, Alexander Kozlov, Timur Pekhovsky, and Yuri Matveev, "Deep speaker embeddings for far-field speaker recognition on short utterances," 2020.

[12] W. Lu L. Wang M. Liu L. Zhang J. Jin J. Xu R. Zhang, J. Wei, "Aret: Aggregated residual extended time-delay neural networks for speaker verification," in *Proc. Interspeech 2020*, 2020, pp. 946–950.

[13] Yun Tang, Guohong Ding, Jing Huang, Xiaodong He, and Bowen Zhou, "Deep speaker embedding learning with multi-level pooling for text-independent speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6116–6120.

[14] Jahangir Alam, Abderrahim Fathan, and Woo Hyun Kang, "Text-Independent Speaker Verification Employing CNN-LSTM-TDNN Hybrid Networks," in *23rd International Conference on Speech and Computer (SPECOM), Lecture Notes in Computer Science, Springer, Cham*, 2021, vol. 12997, pp. 1–13.

[15] Woo Hyun Kang, Jahangir Alam, and Abderrahim Fathan, "Hybrid network with multi-level global-local statistics pooling for robust text-independent speaker recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, vol. accepted for publication.

[16] Hossein Zeinali, Kong Aik Lee, Jahangir Alam, and Lukas Burget, "Short-duration Speaker Verification (SdSV) Challenge 2021: the Challenge Evaluation Plan," 2021.

[17] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech Language*, vol. 60, pp. 101027, 2020.

[18] Kyu J. Han, Jing Pan, Venkata Krishna Naveen Tadala, Tao Ma, and Daniel Povey, "Multistream CNN for Robust Acoustic Modeling," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6873–6877, 2021.

[19] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang, "CN-CELEB: a challenging Chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[20] Lantian Li, Ruiqi Liu, Jiawen Kang, Yue Fan, Hao Cui, Yunqi Cai, Ravichander Vipperla, Thomas Fang Zheng, and Dong Wang, "CN-Celeb: multi-genre speaker recognition," 2020.

[21] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker Recognition for Multi-speaker Conversations Using X-vectors," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5796–5800, 2019.

[22] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Fred Richardson, Suwon Shon, François Grondin, Réda Dehak, Leibny Paola García-Perera, Daniel Povey, Pedro A. Torres-Carrasquillo, Sanjeev Khudanpur, and Najim Dehak, "State-of-the-Art Speaker Recognition for Telephone and Video Speech: The JHU-MIT Submission for NIST SRE18," in *Proc. Interspeech 2019*, 2019, pp. 1488–1492.

[23] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in *Proc. Interspeech 2018*, 2018, pp. 3743–3747.

[24] Chien-Lin Huang, "Speaker Characterization Using TDNN, TDNN-LSTM, TDNN-LSTM-Attention based Speaker Embeddings for NIST SRE 2019," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2020)*, 2020, pp. 423–427.

[25] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[26] S. Wang, J. Rohdin, O. Plchot, L. Burget, K. Yu, and J. Černocký, "Investigation of specaugment for deep speaker embedding learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7139–7143.

[27] Y. Tang, G. Ding, J. Huang, X. He, and B. Zhou, "Deep speaker embedding learning with multi-level pooling for text-independent speaker verification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6116–6120.

[28] Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, David Snyder, Aswin Shanmugam Subramanian, Jan Trmal, Bar Ben Yair, Christoph Boeddeker, Zhaoheng Ni, Yusuke Fujita, Shota Horiguchi, Naoyuki Kanda, Takuya Yoshioka, and Neville Ryant, "CHiME-6 Challenge: Tackling Multi-speaker Speech Recognition for Unsegmented Recordings," in *Proc. The 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020, pp. 1–7.

[29] D. Povey, A. Ghoshal, Gilles Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, Petr Motlícek, Yanmin Qian, Petr Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The kaldi speech recognition toolkit," 2011.