DeepASV System Description for the first edition of the CN-Celeb Speaker Recognition Challenge 2022

Woo Hyun Kang, Jahangir Alam

Computer Research Institute of Montreal (CRIM)

woohyun.kang,jahangir.alam@crim.ca

Abstract

In this report, we provide description of our experimented systems on the CNCeleb dataset. The CNCeleb dataset provides a difficult set of trial that were collected from multiple genres of speech and consists of real-world adversaries, including noise, overlapped background speakers, cross-channel, and short durational test samples. In order to extract a reliable speaker embedding vector under such harsh environment, we have trained multiple systems with different training strategies and architectures. More specifically, we have experimented with not only the conventional ECAPA-TDNN or ResNet architectures, but also the recently proposed multi-stream hybrid neural network. Furthermore, we have trained the systems with speaker discriminative losses, along with a domain generalization training strategy. Our experimental results show that the hybrid architectures can effectively improve the speaker verification performance in a multi-genre scenario. Moreover, fusing different types of hybrid systems further improved the performance, which indicates that different hybrid architectures can learn complementary speaker-dependent information to each other.

1. Introduction

In recent years, various research were done to build a reliable automatic speaker verification (ASV) system, where the system aims to verify whether the given speech is from the claimed speaker or not. Usually, utterance-level fixed-dimensional vectors (i.e., embedding vectors) are extracted from the enrollment and test speech samples and then fed into a backend scoring algorithm (e.g., cosine similarity, probabilisic linear discriminant analysis) to measure their similarity or likelihood of being spoken by the same speaker. Current trends in ASV is to employ depp learning architectures for extracting embedding vectors and have shown good performance especially when a large amount of speech data is available for training the system [1]. In [1, 2], a speaker recognition model consisting of a time-delay neural network (TDNN)-based frame-level network and a segment-level network was trained and the hidden layer activation of the segment-level network denoted as x-vector, was extracted as the embedding vector. In [3], an ECAPA-TDNN architecture was proposed, which has shown state-of-the-art performance by introducing residual and squeeze-and-excitation (SE) components to the widely used TDNN-based embedding system. In [4, 5], a hybrid neural network (HNN) for speaker embedding extraction was proposed, which not only employs different types of network architectures (i.e., 2D-CNN, TDNN, LSTM) but also exploits the short-durational statistics of the hidden representations for bagging the instantaneous speaker-dependent information.

Despite the success of deep learning-based ASV systems in well-matched conditions, the deep learning-based embedding methods are vulnerable to the performance degradation caused by mismatched conditions [6]. In a realistic scenario, there could be numerous mismatches between the enrolled speech and the test speech, including recording channels, environmental noise, room conditions. Since such non-speaker attributes can introduce different variability to the speech distribution, the ASV performance usually degrades when the training or enrollment speech samples are from a different domain than the test speech samples. Therefore, many recent researches focused on building a reliable ASV system for ”in the wild” speech recordings, where the speech samples are collected from diverse sources (e.g., TV shows, web uploaded videos) [7–10].

Table 1: 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 CNCeleb benchmark [9, 10] provides a standard benchmark for evaluating ASV systems on adverse conditions. More specifically, the CNCeleb benchmark consists of the following challenges:

- The dataset consists of audio from multiple genres of speech (e.g., interview, singing, movie, drama).
- The audio samples includes various real-life adversaries (e.g., noise, background speakers, cross-channel, short durational speech).

In order to solve these problems, we experimented with several deep learning-based ASV systems with different architectures and training strategies. More precisely, we have experimented with not only the widely adopted ECAPA-TDNN or ResNet systems, but also the recently proposed multi-stream HNN and ensembled HNN architectures. Furthermore, to exploit the complementarity of different architectures in terms of ASV, we have performed score-level fusion to obtain the final score.

The rest of this paper is organized as follows: The datasets used for training our systems are described in Section 2. In Section 3, detailed information on the submitted systems are described. Section 4 presents the results of the submitted systems and Section 5 concludes the paper.

2. Dataset

The CNCeleb corpus [9, 10] is comprised of CN-Celeb 1 [9] and CN-Celeb 2 [10] subsets and sampled at 16kHz with 16-bit precision. Statistics of CNCeleb corpus is presented in Table 1 in terms 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 1), which consists of 2,793 speakers with 11 different genres (i.e., advertisement, drama, entertainment, interview, live broadcast, movie, play, recitation, singing, speech, vlog) [11]. For a detailed information about the CNCeleb dataset please see [9, 10]. The evaluation subset of the CN-Celeb 1 [9] dataset is used for reporting results.

For building all our systems for the Task 1 SV i.e, the fixed track of the CN-Celeb Speaker Recognition Challenge 2022 (CNSRC 2022) we only used CNCeleb_train (1 & 2) as presented in Table 1 for training/tuning of various steps of our systems.

3. System description

3.1. Acoustic features

In our submitted systems, we have extracted 2 types of handcrafted acoustic features and used them as input:

- Mel-frequency cepstral coefficient (MFCC): 40-dimensional Mel-frequency cepstral coefficients 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.
- Mel-filterbank spectrogram (MFB): 40-dimensional Mel-spectrograms are extracted using an analysis window of 25 msec with a frame shift of 10 msec.

3.2. 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.

3.2.1. Offline Data Augmentation

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.
- Music: A single music file (without vocals) is randomly selected from MUSAN, trimmed or repeated as necessary to match duration, and added to the original signal (5-15dB SNR).
- Noise: MUSAN noises are added at one second intervals throughout the recording (5-15dB SNR).
- Babble: Three to seven speakers are randomly picked from original training data, summed together, then added to the original signal (13-20dB SNR).

For all the trained systems, we commonly apply above mentioned offline augmentation to the input speech prior to the MFCC extraction process [12]. So, all our developed systems use the augmented CNCeleb_train data (original CNCeleb_train + supplementary data generated over CNCeleb_train).

3.2.2. Online Data Augmentation

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

For the MFCC features, we have applied CepsAugment, which is similar to the SpecAugment strategy. However, instead of masking the MFB features, it is directly applied to the MFCC features.

Table 2: 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 → Output*
1	*TDNN-ReLU*	t-2:t+2	d × T → 512 × T
2	*TDNN-ReLU*	t-2,t,t+2	512 × T → 512 × T
3	*TDNN-ReLU*	t-3,t,t+3	512 × T → 512 × T
4	*Dense-ReLU*	t	512 × T → 512 × T
5	*Dense-ReLU*	t	512 × T → 1500 × T
6	*Pooling (mean + stddev)*		1500 × T → 3000
7	*Dense-ReLU*		3000 → 512
8	*Dense-ReLU*		512 → 512
9	*Softmax*		512 → # speakers

Table 3: 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 → Output*
1	*TDNN-ReLU*	t-2:t+2	d × T → 512 × T
2	*Dense-ReLU*	t	512 × T → 512 × T
3	*TDNN-ReLU*	t-2,t,t+2	512 × T → 512 × T
4	*Dense-ReLU*	t	512 × T → 512 × T
5	*TDNN-ReLU*	t-3,t,t+3	512 × T → 1500 × T
6	*Dense-ReLU*	t	512 × T → 512 × T
7	*TDNN-ReLU*	t-4,t,t+4	512 × T → 512 × T
8	*Dense-ReLU*	t	512 × T → 512 × T
9	*Dense-ReLU*	t	512 × T → 1500 × T
10	*Pooling (mean + stddev)*		1500 × T → 3000
11	*Dense-ReLU*		3000 → 512
12	*Dense-ReLU*		512 → 512
13	*Softmax*		512 → # speakers

3.3. Speaker embedding architecture

In this paper, we adopted the following architectures, which have shown competitive performance in the text-independent speaker verification task:

- ResNetSE34 [13]: also known as the Fast ResNet, which follows the same general structure as the original ResNet with 34 layers (ResNet-34) [14] with squeeze-and-excitation [15], but only uses one-quarter of the channels in each residual block to reduce computational cost.
- ECAPA-TDNN [3]: an architecture that achieved state-of-the-art performance in text-independent speaker recognition. The ECAPA-TDNN uses squeeze-and-excitation as in the SE-ResNet, but also employs channel- and context-dependent statistics pooling and multi-layer aggregation.

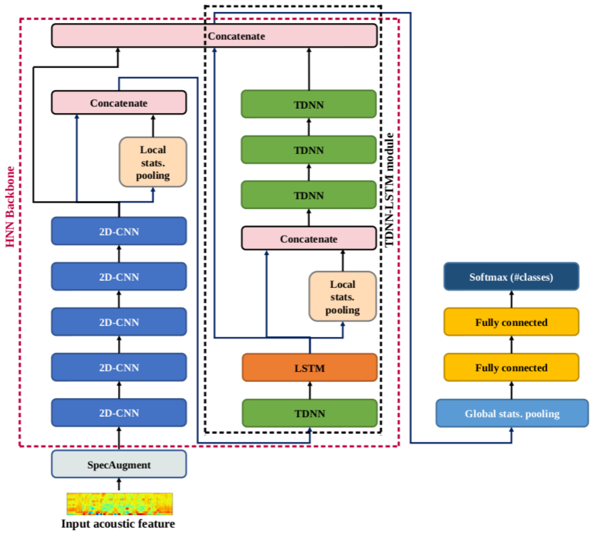

Figure 1: Schematic diagram of the hybrid neural network (HNN) architecture as embeddings extractor for automatic speaker verification task [4, 5]. The HNN backbone architecture is shown inside the red dotted rectangle. Inside the black dotted rectangle the TDNN-LSTM module is presented. Here, the acronyms CNN, TDNN and LSTM stand for Convolutional Neural Network, Time Delay Neural Network and Long Short-Term Memory, respectively [16, 17].

- HNN [4, 5]: a hybrid architecture that employs convolutional neural network (CNN), TDNN, and long-short term memory (LSTM) modules and global-local statistics pooling layers.
- ETDNN-LSTM [16]: ETDNN-LSTM is similar to the standard HNN, but does not employ any 2D-CNN layers. From the ETDNN architecture, the second layer was replaced with the LSTM layer and the local statistics are appended as in the HNN.
- MSHNN [16, 17]: MSHNN follows a similar backbone structure with the HNN, but adopts a multi-stream scheme to capture the speaker information latent in different temporal resolutions.
- ENSEMBLE [16, 17]: ensembled embedding extractor architecture that incorporates different hybrid backbone networks in a parallel manner.

3.3.1. Hybrid neural network (HNN)

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.

3.3.1.1. 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 [18] 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, frame-level representations with information on not only the relation between the local frames, but also the local frequency bins could be obtained.

3.3.1.2. 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 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 [19], where the second TDNN layer of the standard x-vector [20] is replaced with a LSTM layer.

3.3.1.3. Multi-level global-local statistics pooling

In the HNN architecture, a multi-level statistics pooling (MLSP) [21] 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, the HNN extracts the first- and second- order statistics. However, unlike the conventional x-vector, the 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 done within a short durational moving window similarly to the speech activity detection proposed in [22]. 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.

3.3.2. Multi-stream hybrid neural network (MSHNN)

As presented in Figure 2, the MSHNN system follows a similar backbone structure with the standard 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 resolution. 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 [16, 17]. 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 (MLSP) [21] is also employed in the MSHNN framework for

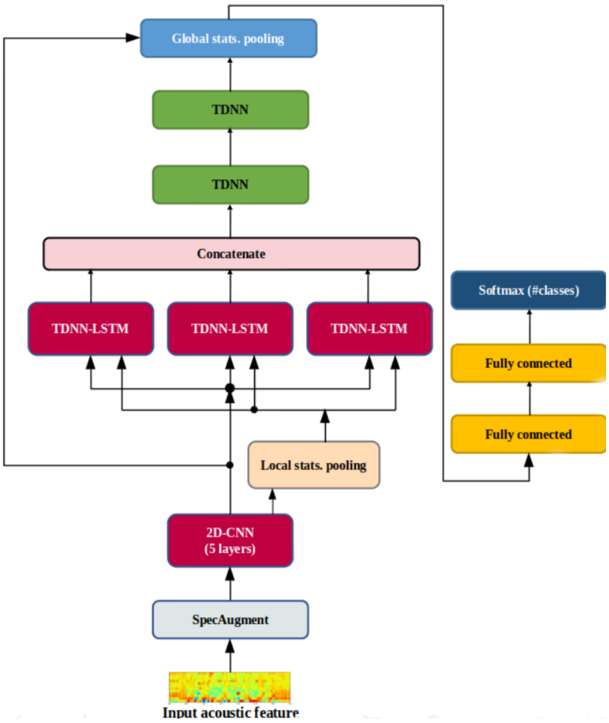

Figure 2: Schematic diagram of the Multi-Stream Hybrid Neural Network (MSHNN) architecture as embeddings extractor for automatic speaker verification task [16, 17].

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.

3.3.3. *Ensembled embedding extractor*

The ensemble embedding extractor architecture, denoted as ENSEMBLE, incorporates multiple hybrid backbone networks in parallel manner. In this paper, we have experimented with 3 different configurations for ENSEMBLE.

Figure 3 depicts the architecture for ENSEMBLE-1, which ensembles the standard HNN and ETDNN-LSTM [17]. The global multi-level statistics pooling is performed from the last layers of the two hybrid backbone architectures 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.

The ENSEMBLE-2 and ENSEMBLE-3 follows the same general framework with ENSEMBLE-1, but ensembles 3 different architectures:

- ENSEMBLE-2: ensembles HNN, ETDNN-LSTM, and ETDNN [16].
- ENSEMBLE-3: same as ENSEMBLE-2 [16] but ensembles HNN, TDNN-LSTM, and TDNN.

The detailed architecture for the TDNN and ETDNN can be found in Table 2 and Table 3, respectively. The general architecture for ENSEMBLE-2 is shown in Figure 4.

3.4. Training objectives

3.4.1. *Softmax-based objectives*

In this paper, we trained the systems using two softmax-based objectives, including the standard softmax cross-entropy and the angular additive margin softmax (AAMSoftmax) objectives [23].

The AAMSoftmax objective is formulated as follows:

$$L_{AAMSoftmax} = -\frac{1}{N}\sum_{i=1}^{N} log(\frac{e^{s(cos(\theta_{y_i,i}+m))}}{K_1}), \quad (1)$$

where $K_1 = e^{s(cos(\theta_{y_i,i}+m))} + \sum_{j=1,j\neq i}^{C} e^{scos\theta_{j,i}}$, N is the batch size, C is the number of classes, y_i corresponds to label index, $\theta_{j,i}$ represents the angle between the column vector of weight matrix W_j and the i-th embedding ω_i, where both W_j and ω_i are normalized. The scale factor s is used to make sure the gradient is not too small during the training and m is a hyperparameter that encourages the similarity of correct classes to be greater than that of incorrect classes by a margin m.

3.4.2. *MIM-DG: Mutual information minimization-based domain generalization*

The MIM-DG regularization strategy [24] aims to extract an embedding ω from the input speech X with maximum speaker-dependent information while suppressing the nuisance information (e.g., genre). To maximize the speaker information within the embedding vector, the embedding network is trained to minimize the cross-entropy-based loss function, such as AAMSoftmax given the speaker embedding and the speaker labels.

3.4.2.1. *Mutual information upper bound and Conditional likelihood estimation via Normalizing Flow*

Given the embedding vector ω and its corresponding genre label c, to minimize $I(\omega; c)$, we aim to estimate and minimize the upper bound of the mutual information via the CLUB formulation [25]. However, in order to achieve this, we need to estimate the conditional likelihood $p(\omega|c)$.

For this purpose, a generative model is employed, more specifically a normalizing flow model called Real NVP (Real-valued Non-Volume Preserving) [26]. Once the RealNVP model is trained, we can estimate the mutual information upper bound as follows:

$$\begin{aligned} L_{nuisance} =& E_{p(\omega,c)}[\log p_\omega(\omega|c)] \\ &- E_{p(\omega)p(c)}[\log p_\omega(\omega|c)], \end{aligned} \quad (2)$$

where $\log p_\omega(\omega|c)$ is the conditional log-likelihood estimated using the conditional RealNVP.

3.4.2.2. *Training strategy*

In the MIM-DG framework [24], the embedding network is trained where the discriminability of the embedding ω in terms of the speaker label y is maximized while the mutual information between ω and the genre label c is minimized. To accomplish

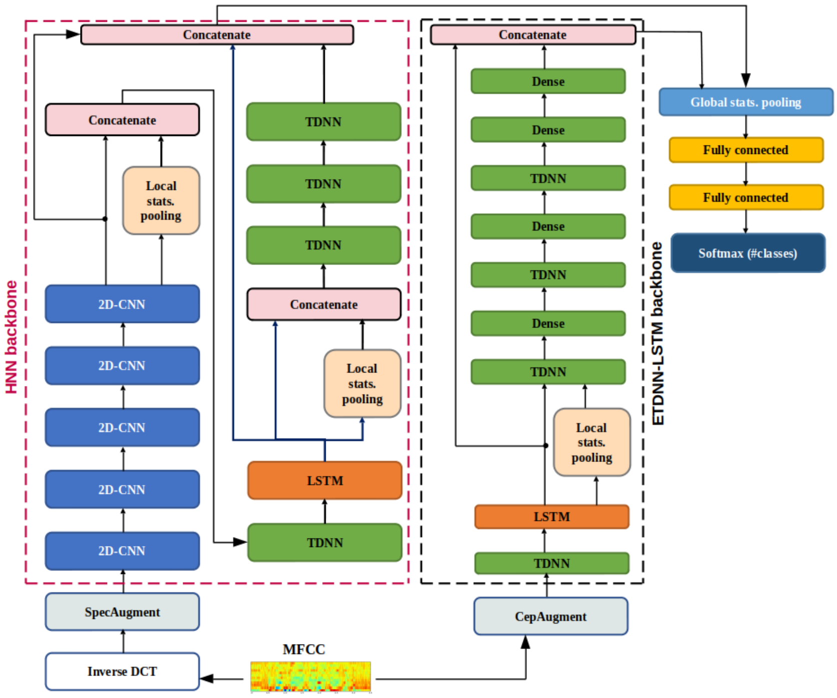

Figure 3: Schematic diagram of the ENSEMBLE-1 system. This embedding extractor employs two hybrid backbone architectures, namely the HNN (hybrid neural network) backbone and the extended TDNN-LSTM (ETDNN-LSTM) backbones, in parallel fashion. In this Figure the HNN backbone module is marked by the red dashed rectangle and inside the black dashed rectangle is the ETDNN-LSTM backbone architecture [17].

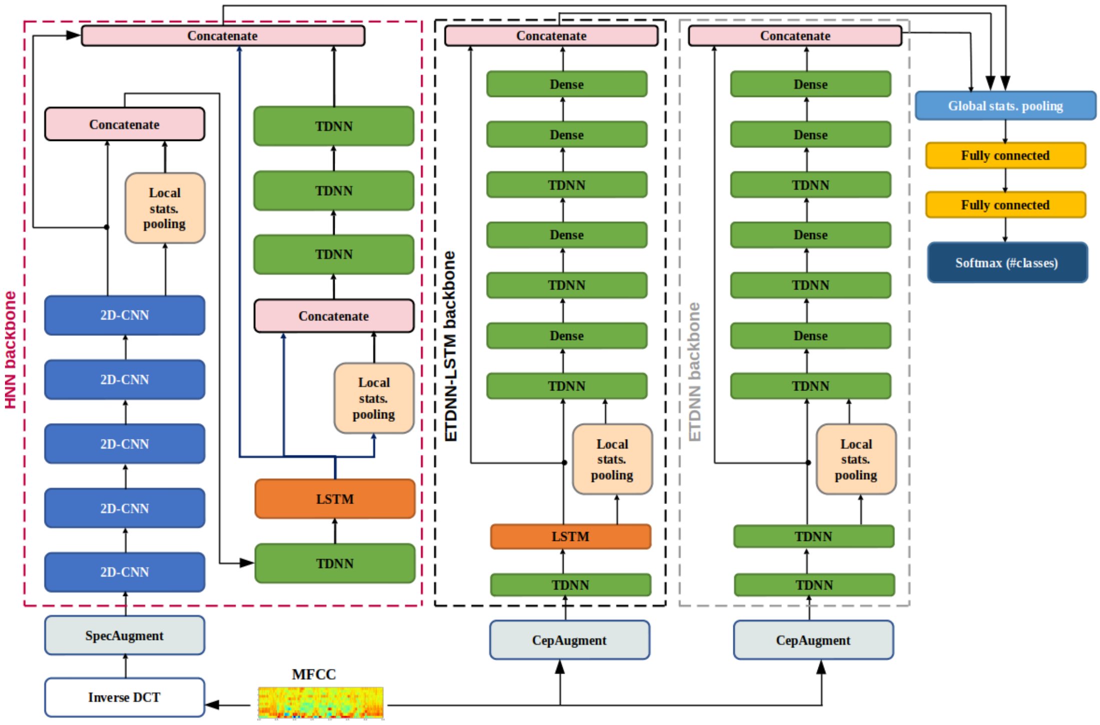

Figure 4: Schematic diagram of the ENSEMBLE-2 system. This embedding extractor employs three hybrid backbone architectures, namely the HNN (hybrid neural network) backbone, extended TDNN (ETDNN) and the ETDNN-LSTM backbones, in parallel fashion. In this Figure the HNN backbone module is marked by the red dashed rectangle and inside the black dashed rectangle is the ETDNN-LSTM backbone architecture [16].

Table 4: The experimental results of the deep embedding systems on the CNCeleb1 evaluation sets in terms of EER and minimum detection cost functions (minDCF)

#	Architecture	Input feature	Training objective	Backend	EER [%]	minDCF
Baseline	TDNN	MFCC	Softmax	PLDA	14.18	0.6407
1	ResNetSE34	MFB	AAM-Softmax	Cosine similarity	11.61	0.6692
2	ResNetSE34	MFB	AAM-Softmax	PLDA	10.71	0.5840
3	ECAPA-TDNN	MFCC	AAM-Softmax	Cosine similarity	11.84	0.4955
4	ECAPA-TDNN	MFCC	AAM-Softmax + MIM-DG	Cosine similarity	11.79	0.4914
5	HNN	MFB	Softmax	LDA (175) + PLDA	8.99	0.4881
6	ENSEMBLE-3	MFCC	Softmax	LDA (130) + PLDA	8.68	0.4903
7	ENSEMBLE-3	MFCC	Softmax	LDA (180) + PLDA	8.96	0.4723
8	ENSEMBLE-2	MFCC	Softmax	LDA (180) + PLDA	8.87	0.4740
9	ENSEMBLE-2	MFCC	Softmax	LDA (220) + PLDA	8.91	0.4694
10	ENSEMBLE-1	MFCC	Softmax	LDA (130) + PLDA	8.69	0.4920
11	ENSEMBLE-1	MFCC	Softmax	LDA (160) + PLDA	8.73	0.4837
12	ETDNN-LSTM	MFCC	Softmax	LDA (190) + PLDA	10.18	0.5208
13	MSHNN	MFB	Softmax	LDA (200) + PLDA	9.05	0.4708
14	ENSEMBLE-3	MFCC	Softmax	LDA (150) + PLDA	8.72	0.4893
15	ENSEMBLE-2	MFCC	Softmax	LDA (220) + PLDA	8.85	0.4688
16	ENSEMBLE-2	MFCC	Softmax	LDA (130) + PLDA	8.68	0.4905
17	HNN	MFB	Softmax	LDA (190) + PLDA	9.08	0.4855
18	ETDNN-LSTM	MFCC	Softmax	LDA (250) + PLDA	10.10	0.5202
19	ENSEMBLE-2	MFCC	Softmax	LDA (250) + PLDA	8.97	0.4706
Score-level fusion of #5, #7, #9, #11, #13, #15, #17, #19					**8.22**	**0.4504**

this, the MIM-DG optimizes the network with the following objective function, which incorporates a speaker discriminant loss $L_{speaker}$ and a mutual information regularization loss Equation 2:

$$L_{MIM-DG} = -L_{speaker} + \beta L_{nuisance}, \quad (3)$$

where β is a predefined coefficient. In our experiment, we used $\beta = 0.001$.

The MIM-DG training is done in a 2-stage process: embedding network update and RealNVP update. In the embedding network update phase, we freeze the RealNVP parameters and estimate the conditional likelihoods to compute $L_{nuisance}$. Then the embedding network and classification network parameters are updated through $L_{MIM-DG} = -L_{speaker} + \beta L_{nuisance}$. In the RealNVP update phase, the embedding network parameters are frozen and the embeddings are extracted. Given the training data and their corresponding embeddings, the RealNVP is updated via likelihood maximization.

4. Results

Table 4 shows the performance of the experimented systems on the CNCeleb1 evaluation set. As shown in the results from System 1 and 2, the probabilistic linear discriminant analysis (PLDA) backend was more effective than the simple cosine similarity scoring on the CNCeleb dataset. This may be attributed to the PLDA's well known strength in domain mismatched condition, as the CNCeleb evaluation set consists of cross-genre trials. Moreover, from System 3 and 4, we could see that the MIM-DG strategy can improve the performance by disentangling the genre information from the speaker embedding vectors.

In terms of the architecture, it could be seen that the hybrid systems (i.e., HNN, ETDNN-LSTM, ENSEMBLE) generally performs much better than the standard ResNetSE34, TDNN and ECAPA-TDNN systems. This tells us that the hybrid architectures can effectively extract the speaker information even from an adverse condition, which may be accredited to the hybrid systems' capability to capture the time-frequency correlation in a long temporal context. While the standard HNN and MSHNN system was able to outperform the baseline system, ensembling multiple hybrid architectures (i.e., ENSEMBLE-1, ENSEMBLE-2, ENSEMBLE-3) further improved the performance. Among the individual systems, the best performance was achieved by the ENSEMBLE-3 system with 130 dimensional linear discriminant analysis (LDA) and PLDA backend (i.e., System 6), which outperformed the standard TDNN baseline with a relative improvement of 38.79% in terms of EER.

We have also applied score-level fusion across different systems. With a simple sum fusion scheme, where the scores from different systems are aggregated via summation, the best performance was achieved by fusing System 5, 7, 9, 11, 13, 15, 17, and 19, which consists of HNN, ENSEMBLE-1, ENSEMBLE-2, ENSEMBLE-3, and MSHNN architectures. The best performing fused score outperformed the best performing individual system (i.e., System 6) with a relative improvement of 5.3% in terms of EER. This indicates that different hybrid architectures can learn complementary speaker-dependent information to each other.

5. Conclusion

In this report, we described our experimented systems on the CNCeleb dataset. The CNCeleb benchmark provides a difficult set of trials where the speech samples are collected from multiple genres of videos and consists of several challenging real-world adversaries. In order to overcome these problems, we experimented with several deep learning-based automatic speaker verification (ASV) systems with different architectures and training strategies, including the recently proposed hybrid architecture-based systems and MIM-DG strategy. Our experimental results showed that the hybrid systems can effectively capture the speaker information even in a cross-genre scenario, and ensembling multiple hybrid systems further improved the

[*] Systems #5 to #19 were designed and developed by J. Alam

performance. Simple score-level fusion of different ensembled hybrid systems showed the best performance, which indicates that different hybrid architectures can learn complementary information relevant to the ASV task.

In our future studies, we will focus on applying a more sophisticated fusion method for exploiting the complementarity between different hybrid systems. Moreover, we will analyze the effect of genre information on the performance of the hybrid ASV systems.

6. 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.

7. References

[1] David Snyder, D. Garcia-Romero, Daniel Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH*, 2017.

[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)*, 2018, pp. 5329–5333.

[3] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*, Helen Meng, Bo Xu, and Thomas Fang Zheng, Eds. 2020, pp. 3830–3834, ISCA.

[4] 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.

[5] 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.

[6] Z. Meng, Y. Zhao, J. Li, and Y. Gong, "Adversarial speaker verification," in *ICASSP*, 2019, pp. 6216–6220.

[7] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[8] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[9] 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.

[10] 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.

[11] 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," 2021.

[12] 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.

[13] 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.

[14] 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.

[15] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[16] Jahangir Alam, Woo Hyun Kang, and Abderrahim Fathan, "Novel Neural Speaker Embedding Extractors for Text-Independent Speaker Verification," in *Interspeech (Submitted)*, 2022.

[17] Jahangir Alam, Woo Hyun Kang, and Abderrahim Fathan, "Hybrid Neural Network-based Deep Embedding Extractors for Text-Independent Speaker Verification," in *Proc. of Odyssey*, 2022.

[18] 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.

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

[20] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[21] 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.

[22] 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.

[23] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Cotsia, and Stefanos P Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[24] Woo Hyun Kang, Jahangir Alam, and Abderrahim Fathan, "Domain generalized speaker embedding learning via mutual information minimization," in *Odyssey*, 2022.

[25] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin, "Club: A contrastive log-ratio upper bound of mutual information," *arXiv preprint arXiv:2006.12013*, 2020.

[26] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, "Density estimation using real NVP," in *ICLR*, 2017.