Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

Advances in Speaker Recognition for Telephone and Audio-Visual Data: the JHU-MIT Submission for NIST SRE19

Jesús Villalba[1], *Daniel Garcia-Romero*[2], *Nanxin Chen*[1], *Gregory Sell*[2], *Jonas Borgstrom*[3], *Alan McCree*[2], *L. Paola García-Perera*[1], *Saurabh Kataria*[1], *Phani Sankar Nidadavolu*[1], *Pedro A. Torres-Carrasquillo*[3], *Najim Dehak*[1]

[1]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
[2]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD, USA
[3]MIT Lincoln Laboratory, Lexington, MA, USA

{jvillalba,dgromero}@jhu.edu, jonas.borgstrom@ll.mit.edu

Abstract

We present a condensed description of the joint effort of JHU-CLSP, JHU-HLTCOE and MIT-LL for NIST SRE19. NIST SRE19 consisted of a Tunisian Arabic Telephone Speech challenge (CTS) and an audio-visual (AV) evaluation based on Internet video content. The audio-visual evaluation included the regular audio condition but also novel visual (face recognition) and multi-modal conditions. For CTS and AV-audio conditions, successful systems were based on x-Vector embeddings with very deep encoder networks, i.e, 2D residual networks (ResNet34) and Factorized TDNN (F-TDNN). For CTS, PLDA back-end domain-adapted using SRE18 eval labeled data provided significant gains w.r.t. NIST SRE18 results. For AV-audio, cosine scoring with x-Vector fine-tuned to full-length recordings outperformed PLDA based systems. In CTS, the best fusion attained EER=2.19% and Cprimary=0.205, which are around 50% and 30% better than SRE18 CTS results respectively. The best single system was HLTCOE wide ResNet with EER=2.68% and Cprimary=0.258. In AV-audio, our primary fusion attained EER=1.48% and Cprimary=0.087, which was just slightly better than the best single system (EER=1.78%, Cprimary=0.101).

For the AV-video condition, our systems were based on pre-trained face detectors–MT-CNN and RetinaFace– and face recognition embeddings–ResNets trained with additive angular margin softmax. We focused on selecting the best strategies to select the enrollment faces and how to cluster and combine the embeddings of the faces of the multiple subjects in the test recording. Our primary fusion attained EER=1.87% and Cprimary=0.052. For the multi-modal condition, we just added the calibrated scores of the individual audio and video systems. Thus, we assumed complete independence between audio and video modalities. The multi-modal fusion provided impressive improvement with EER=0.44% and Cprimary=0.018.

1. Introduction

The National Institute of Standards and Technology (NIST) regularly conducts speaker recognition evaluations (SRE) to assess the state-of-the-art of the technology [1]. These evaluations focus on the speaker detection task, i.e., given one or more enrollment recordings and a test recording, we need to decide whether the enrollment speaker is also present in the test. Along the years, SRE has evolved from telephone speech [2], to far-field microphone [3, 4], to non-English telephone speech [5, 6] and audio from amateur Internet videos [6, 7]. NIST SRE19[1] was similar to SRE18. It consisted of a telephone condition (CTS) challenge with Tunisian Arabic speech recorded in Tunisia through PSTN and VoIP networks. It also included a regular evaluation on amateur Internet videos (VAST) [7]. As novelty, the image part of the videos was also provided, which derived into a video only evaluation (face recognition) and a multi-modal evaluation (fusion of audio and video).

In this paper, we analyze the JHU-MIT submission to NIST SRE18. This is the joint effort of teams at Johns Hopkins CLSP and HLTCOE, and MIT Lincoln Laboratory. All our audio systems were based on some flavour of x-Vector [8] front-end plus PLDA [9] or cosine scoring back-end. For the video part, we used pre-trained face detectors and ArcFace face recognition embeddings [10]. We focused on selecting the best strategies to select the enrollment faces and how to cluster and combine the embeddings of the faces of the multiple subjects in the test recording.

2. Datasets

2.1. CTS condition

As in 2018, NIST SRE19 CTS condition consisted of Tunisian Arabic recorded in Tunisia from the *Call My Net 2* corpus (CMN2). Given that most training data available is English recorded in the US, this condition is most challenging. During the duration of the challenge users were able to submit scores and monitor performance in a ∼600k trials eval set (Progress set). After the challenge deadline, the results on the larger Eval set (∼2M trials) were released.

We used different datasets combinations to train x-vectors:

- CLSP18: This is the same dataset that JHU-CLSP team used in SRE18 [11, 12]. It contained VoxCeleb1+2; NIST SRE04-12 and MIXER 6 telephone; SRE12 and MIXER 6 microphone phonecalls; and Switchboard phase1-3 and cellular1-2. It was augmented 3× by adding noise and/or reverberation. It contained 13k speakers.
- CLSP19: This is CLSP18 where we removed microphone phonecalls and augmented 5×.
- COE: Similar to CLSP19 without SRE12 and MIXER

[1] https://www.nist.gov/itl/iad/mig/nist-2019-speaker-recognition-evaluation

10.21437/Odyssey.2020-39

6 and adding GSM AMR codec augmentation[2] on VoxCeleb data

- MITLL: Similar to CLSP19 without SRE12 but with only $3\times$ augmentation.

Impulse responses were taken from the AIR dataset[3] and noises were taken from the MUSAN corpus[4].

Back-ends were trained with SRE English telephone data and adapted using SRE18 CTS Dev unlabeled and Eval labeled. SRE18 CTS Dev unlabeled was also used for score normalization. SRE18 CTS Dev labeled was used for calibration and performance monitoring.

2.2. Audio-Visual condition

The SRE19 Audio-Visual (AV) condition consisted of internet videos extracted from the VAST corpus. While in SRE18 we only had access to the audio part of the video, in SRE19 we also had access to the image part. This derived into three evaluation conditions: audio only, video only (face recognition) and multimodal. These are amateur videos so a wide range of acoustic conditions may be expected. Also, videos may contain multiple speakers, so diarization is needed to isolate the target speaker. In the enrollment side, ground truth diarization marks were provided for audio; and key frames and face bounding-boxes for video.

x-Vectors and PLDA for the audio condition were trained on Voxceleb1+2 with augmentation, containing 7K speakers. Data from Speakers In The Wild (SITW) and SRE18 VAST Dev was used for x-vector centering and adaptive S-Norm as described in [11, 12]. This year, we also added 10-60 seconds segments extracted from Dihard II evaluation dev and eval sets to the S-Norm cohort. The ground truth RTTMs were used to create ground truth VAD to get the speech of a single speaker per segment. For calibration/fusion and performance monitoring of the audio condition, we used SRE18 VAST Eval set. We considered SRE19 AV Dev too small to be reliable.

For the visual condition, we used Janus [13] Dev core for AS-Norm. No other data was used for training since we used pre-trained embeddings and cosine scoring back-ends. For performance monitoring, we used Janus Eval core and SRE19 AV Dev. SRE19 AV Dev was used to train calibration/fusion. We did not trust on Janus Eval to train the calibration because there was a significant mismatch between Janus and SRE19 Dev score distributions.

3. Audio front-end

All our front-ends were based on some flavour of the x-Vector approach [14, 8].

3.1. Kaldi TDNN x-vectors

Acoustic features for Kaldi x-vectors were 23 dimension and 40 dimension MFCC for CTS and AV conditions respectively. We used short-time mean normalization and removed silence frames. VAD was performed using Kaldi energy VAD for CTS and neural network trained on AVA-Speech [15] for AV.

Kaldi x-Vector implementations were based of TDNN architectures. The MITLL team used extended TDNN (E-TDNN) as defined in [16, 12], which includes 12 layers from input to embedding. The JHU-CLSP team explored several versions of factorized TDNN (F-TDNN) [17] with skip connections. The first version (v1) is the same architecture that we used for SRE18, which is also described in [12]. This included 12 layers from input to embedding; and F-TDNN layers of 1024 dimension and 256 bottleneck dimension. The second version (v2) is the same as v1 but with F-TDNN layers of 2048 dimension and 512 bottleneck dimension. Finally, the third version (v3) is deeper and narrower F-TDNN with 16 layers from input to embedding; and F-TDNN layers of 725 dimension and 180 bottleneck dimension. All networks used mean+stddev pooling and were trained with softmax cross-entropy.

Eventually for CTS condition, we used MITLL E-TDNN (MITLL-1 in Table 2); F-TDNN v1 x-vector trained on CLSP18 setup (CLSP18-1); and F-TDNN v3 trained on CLSP19 setup (CLSP19-2). For AV condition, we used MITLL E-TDNN (MITLL-1 in Table 4); F-TDNN v3 (CLSP-1) and F-TDNN v2 (CLSP-2).

3.2. JHU-HLTCOE PyTorch x-vectors

Acoustic features for HLTCOE x-vectors were short-time mean normalized 64 dimension and 80 dimension Mel-filter-banks for CTS and AV conditions respectively. VAD was performed using Kaldi energy VAD for CTS and neural network trained on SRE+internal data for AV.

Several encoder networks were tested, i.e., E-TDNN, ResNet34, Wide ResNet34 and 1D ResNet34. ResNet34 was modified from its original form [18] by removing the downsampling in the first convolutional layer (set stride=1) and removing maxpooling layer. This network had 64 channels in the first residual block and 512 in the last one. Wide ResNet34 doubled the number of channels to 128 and 1024 respectively. 1D Resnet34 was a version of ResNet34 where 2D convolutions were replaced by 1D convolutions. All networks used mean+stddev pooling. This networks were implemented in PyTorch and trained with SGD using 4 GPUs in parallel.

Eventually for CTS condition we had 5 of these x-vector networks (COE-{1-5}) summarized in Table 2. COE-{1,2} were trained with softmax cross-entropy while the others were trained with additive margin softmax (AM-Softmax) [19]. For AV condition, we just had one ResNet34 x-vector trained with AM-Softmax.

3.3. JHU-CLSP PyTorch ResNet2D x-vectors

Acoustic features for this version were short-time mean normalized 23 dimension and 40 dimension Mel-filter-banks for CTS and AV conditions respectively. VAD was as for Kaldi x-vectors.

For CTS, we used a Thin ResNet50 (16 channels in the first residual block) with *multi-head attention* pooling (64 heads) (CLSP19-2 in Table 2). For AV, we used a Thin ResNet34 with *learnable dictionary encoder* pooling layer (LDE) [20, 12] with 64 clusters (CLSP-3 in Table 4); and a ResNet34 (64 channels in first block) with LDE with 8 clusters (CLSP-4). These networks were trained with additive angular margin softmax (AAM-Softmax) [10] and Adam optimizer.

3.4. Speaker diarization

Since the test utterance poses a multi-speaker scenario, speaker diarization was required. JHU-CLSP used the setup described in [12], which is similar to Kaldi x-vector callhome diarization

[2] http://www.3gpp.org/ftp/Specs/archive/26_series/26.073/26073-800.zip
[3] http://www.openslr.org/resources/28
[4] http://www.openslr.org/resources/17

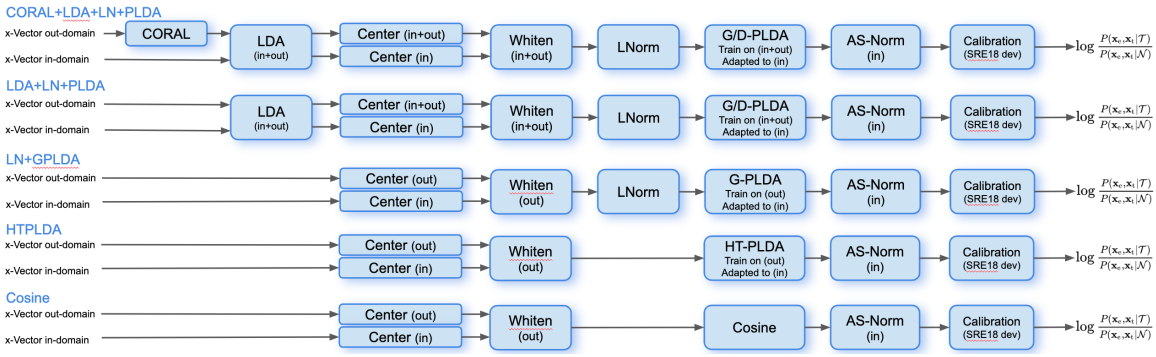

Figure 1: Audio back-end configurations.

recipe[5], based on [21]. We used CLSP-1 F-TDNN embeddings for this. JHU-HLTCOE used the leave-one-out PLDA GMM approach from [22]. This algorithm alternates between updating the speaker models and generating segment speaker posteriors.

4. Audio back-end

4.1. CTS back-end

Figure 1 depicts different back-end configurations used in our submissions. For CTS, JHU-CLSP and MIT-LL used CORAL+LDA+LN+PLDA. CLSP used generative PLDA while MIT-LL used discriminative version. HLTCOE systems used Heavy tailed PLDA [23] with embeddings trained with Softmax cross-entropy. When using AM-Softmax, HLTCOE used LN+GPLDA, since AM-Softmax optimizes length normalized embeddings.

JHU-HLTCOE systems introduced D-Cosine method, which refined the x-vector classification head by using full-length recordings as described in [24]. This method allowed us to just use cosine scoring back-end. For this refinement, we augmented SRE18 CTS Eval (188 speakers) to obtain 100K utterances. We combined this data with SRE04-10 (4K speakers). This resulted in a dataset balanced in the number of utterances between the two sets.

We used SRE18 CTS Eval as in domain adaptation data for Correlation alignment (CORAL) [25], LDA, centering and PLDA; and SRE18 CTS unlabeled for adaptive S-Norm. JHU-CLSP and MITLL also used SRE18 CTS unlabeled for PLDA adaptation using pseudo-speaker labels from clustering. CORAL target covariance and adapted PLDA within/between-class covariances were computed as weighted sum of in and out-of-domain covariances [26].

4.2. AV-back-end

For AV, JHU-CLSP and MIT-LL used LDA+LN+PLDA configuration with generative (CLSP) or discriminative (MIT-LL) PLDA. Meanwhile, JHU-HLTCOE used D-Cosine back-end with x-vector network fine-tuned with full-length VoxCeleb recordings. For in-domain centering, we used SITW (HLTCOE and MIT-LL) or SITW + SRE18 VAST Dev (CLSP). For AS-Norm, we used SITW + SRE18 VAST Dev + Dihard II cohort as described in Section 2.2.

[5] https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization/v2

5. Video front-end

5.1. JHU-CLSP embeddings

For face recognition, we used original InsightFace ArcFace [10] embeddings and RetinaFace [27] face detection implementations[6]. We obtained RetinaFace pretrained model[7]; and four ArcFace models[8] with different sizes: LResNet100 (r100), LResNet50 (r50), LResNet34(r34) and MobileFaceNet (mobile). ArcFace models were trained on MS1M-Arcface dataset. RetinaFace was trained on WiderFace dataset[9].

The face detection procedure was different for enrollment and test:

- Enrollment: We applied RetinaFace detector at the frames given in reference bounding boxes (bbox) ± 2 frames. Then we kept the faces that overlap with the ref. bbox. If we don't detect any faces in the ref. bbox we keep all the faces in the video.
- Test: We detected faces over the full video extracting frames at 1 frame per second.

After face detection, we aligned the faces with the landmarks obtained with Multi-task Cascaded Convolutional Network (MT-CNN) [28] (included with the InsightFace models) and extracted embeddings with the InsightFace model. If the MT-CNN failed to detect the landmarks, we used the landmarks provided by the RetinaFace detector.

5.2. JHU-HLTCOE embeddings

We utilized a third party InsightFace implementation[10]. We used pre-trained ResNet-101 embedding model[11], which was trained on the MS-Celeb-1M dataset (3.8M faces).

Face detection was performed with a MT-CNN [28], also included in the downloaded implementation. The default settings were used for an initial detection pass run on frames extracted every quarter of a second from all videos. Faces with a final confidence score less than 0.95 were subsequently excluded in order to minimize false alarms or low-resolution faces that might corrupt downstream processing.

Enrollment models were created by averaging embeddings from any detected boxes that overlap with the provided enrollment box. Frames were searched for overlapping boxes within a second before or after the enrollment box's frame (a total of 9

[6] https://github.com/deepinsight/insightface
[7] https://github.com/deepinsight/insightface/tree/master/RetinaFace
[8] https://github.com/deepinsight/insightface/wiki/Model-Zoo
[9] http://shuoyang1213.me/WIDERFACE/WiderFace_Results.html
[10] https://github.com/foamliu/InsightFace-v3
[11] https://github.com/foamliu/InsightFace-v3/releases/download/v1.0/BEST_checkpoint.tar

Table 1: JHU-CLSP video back-end enroll/test methods.

Enroll/Test	All	AHC	Self-Att	Self-Att+Enroll-Att
All	be1			
Avg	be2			
Median	be3	be4	be6	be7
AHC		be5		
Self-Att			be9	

frames searched). If no overlapping boxes were found with any of the given enrollment faces, that model was left empty and scores for its trials were added simply as zero after calibration.

Test faces were clustered using AHC to 21 clusters in all videos. The choice of 21 clusters (as well as the confidence threshold and search range for enrollment faces) was determined with experiments on the Janus dataset.

6. Video back-end

6.1. JHU-CLSP back-end

We evaluated different back-end strategies based on cosine scoring. In the enrollment and test side, these back-ends follow one of these strategies:

- All: keep all the face embeddings.
- Avg: average the face embeddings.
- Median: compute the median of the face embeddings.
- AHC: perform agglomerative clustering (AHC) with stopping threshold=0.8.
- Self-Att: Use self-attention [29] procedure to enhance embeddings and keep all of them. This method consisted in, for each embedding $\mathbf{x}_t$ in the video, we computed a new enhanced embedding $\mathbf{y}_t$ as

$$\mathbf{p}_t = \mathrm{softmax}_k(a\cos(\mathbf{x}_t, \mathbf{x}_k)) \quad t = 1, \ldots, T \tag{1}$$

$$\mathbf{y}_t = \sum_{k=1}^{T} p_{tk}\mathbf{x}_k \qquad t = 1, \ldots, T \tag{2}$$

  where we set $a = 2$ empirically. The idea is to improve each embedding by doing a weighted average of the embeddings that are closer to it.

- Enroll-Att: apply attention procedure between enrollment embedding $\mathbf{e}_i$ of video i and test embeddings $\mathbf{x}_j$ to produce a single test embedding $\mathbf{z}_{ij}$ closer to the enrollment, that is

$$\mathbf{q}_{ij} = \mathrm{softmax}_k(b\cos(\mathbf{e}_i, \mathbf{x}_{jk})) \tag{3}$$

$$\mathbf{z}_{ij} = \sum_{k=1}^{T} q_{ijk}\mathbf{x}_{jk} \tag{4}$$

  where we set $b = 7$. In this case, the final test embedding used in each trial depends on the enrollment embedding.

We obtained several back-ends using different enroll/test combinations of the above methods, shown in Table 1. Those methods produce one or more *refined* embeddings in each side. We scored all enrollment vs test *refined* embeddings and computed the maximum.

For each back-end, we had versions with and without adaptive S-Norm. We used the top 1000 element from JANUS Dev as cohort.

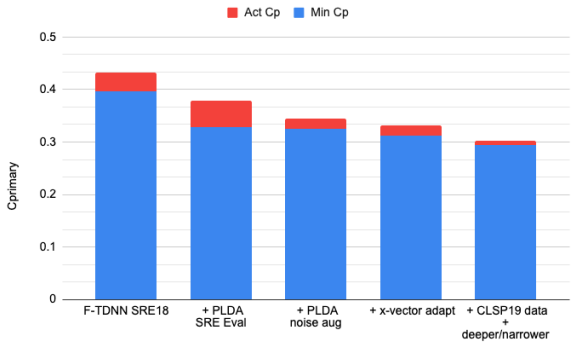

Figure 2: Factorized TDNN results on SRE19 CTS Eval.

6.2. JHU-HLTCOE back-end

For a given model/test pair, the enrollment model was scored against all 21 face clusters using cosine scoring of the embeddings, and the maximum score was kept for the trial.

7. Results CTS

7.1. Single systems

Table 2 presents results for the single systems used in our submissions. Cprimary is the average of DCF at priors 0.01 and 0.005. If we compare the results for SRE18 Dev and SRE19 Eval, we observe that they are not well correlated. We conclude that SRE18 Dev is too small to be reliable (only 25 speakers). We think that a larger development set–including some speakers from the SRE18 Eval–, could provide a better approximation for SRE19 Eval performance. Thus, we mainly focus on the SRE19 results for our analysis.

The best performing system was a wide ResNet34 (COE-5) with GPLDA back-end. However, it was just 5% better than the standard ResNet34 (COE-3). Thin version of ResNet34 (CLSP19-2) performed significantly worse, though setup is not fully comparable. The third best system was the 1D version of ResNet34 (COE-4), while Kaldi deeper version of F-TDNN (CLSP19-1) was in the fourth position. The 12-layer F-TDNN (CLSP18-1) was our best network in previous NIST SRE18 evaluation. Comparing CLSP18-1 to COE-5, we improved x-Vector performance by 23% in this new evaluation. Comparison between COE-2 and COE-3 systems seems to confirm the utility of large margin losses to improve performance.

Comparing COE systems with D-Cosine and PLDA scoring, we observe that the simple PLDA adaptation was more effective than x-vector fine-tuning with in-domain data. D-Cosine produced miscalibration, greatly degrading actual Cprimary.

7.2. F-TDNN analysis

Figure 2 shows how we improved factorized TDNN performance during the challenge. The baseline system consisted of our SRE18 F-TDNN network with PLDA back-end adapted using only SRE18 dev unlabeled set. Adding SRE18 Eval to PLDA adaptation improved by 13% relative. Then, we added noise augmentation to the PLDA data obtaining another 9% improvement, mainly due to calibration correction. Following, we fine-tuned the x-vector network with a balanced amount on SRE telephone and SRE18 Eval data obtaining a 4% improvement. x-Vector adaptation improved by 15% on dev, unfortunately this improvement did not hold in the eval. Finally, we changed the SRE18 x-Vector network by another F-TDNN deeper and narrower (CLSP19-1 in Table 2) and trained on CLSP19 training

Table 2: Results single systems in CTS condition

System	Arch	Layers	Chann./Emb. dimension	Loss	Back-end	SRE18 CTS Dev EER(%)	SRE18 CTS Dev Min Cp	SRE18 CTS Dev Act Cp	SRE19 CTS Eval EER(%)	SRE19 CTS Eval Min Cp	SRE19 CTS Eval Act Cp
COE-1	E-TDNN	11	1024/512	Softmax	HT-PLDA	3.79	0.207	0.223	2.98	0.274	0.319
COE-2	ResNet2D	34	64/512	Softmax	HT-PLDA	4.06	0.222	0.233	3.06	0.284	0.320
COE-3	ResNet2D	34	64/256	AM-Softmax	D-Cosine	4.65	0.229	0.231	3.31	0.273	0.386
					LN+GPLDA	3.44	0.221	0.240	**2.76**	**0.256**	**0.267**
COE-4	ResNet1D	34	1024/256	AM-Softmax	D-Cosine	4.43	**0.204**	**0.207**	3.54	0.302	0.408
					LN+GPLDA	3.41	0.222	0.232	3.01	0.272	0.287
COE-5	ResNet2D	34	128/256	AM-Softmax	D-Cosine	3.99	**0.203**	0.214	3.21	0.263	0.333
					LN+GPLDA	**3.27**	**0.204**	0.224	**2.68**	**0.248**	**0.253**
CLSP18-1	F-TDNN	12	1024/512	Softmax	LDA+LN+GPLDA	4.26	0.228	0.236	3.76	0.312	0.331
CLSP19-1	F-TDNN	16	725/512	Softmax	LDA+LN+GPLDA	3.95	0.255	0.269	3.75	0.295	0.302
CLSP19-2	ResNet2D-MHAtt	50	16/400	AAM-Softmax	LDA+LN+GPLDA	4.03	0.228	0.247	3.88	0.332	0.372
MITLL-1	E-TDNN	12	512/512	Softmax	LDA+LN+DPLDA	4.00	0.256	0.271	3.80	0.319	0.368

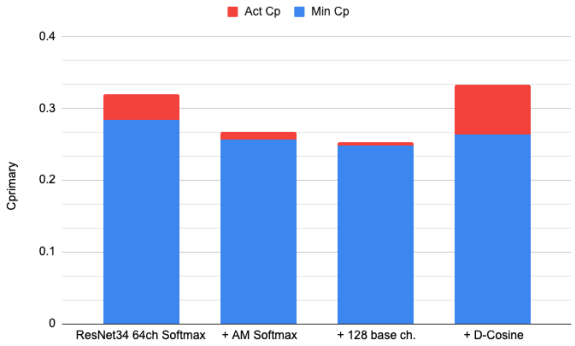

Figure 3: JHU-HLTCOE ResNet34 results on SRE19 CTS Eval.

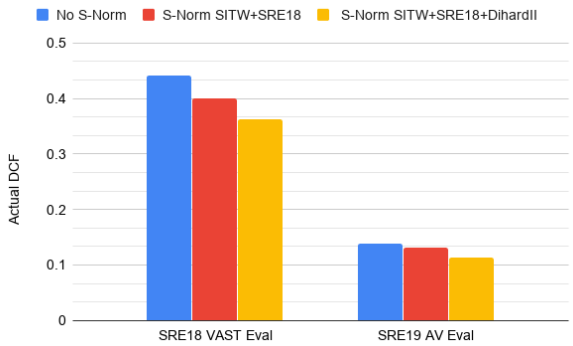

Figure 4: AS-Norm impact on SRE18 VAST and SRE19 AV

setup, which improved by another 9%. The total improvement from the baseline was 30% relative.

7.3. JHU-HLTCOE ResNet34 analysis

Figure 3 presents some analysis on the JHU-HLTCOE team 2D ResNet34. The baseline system consisted on ResNet34 trained with softmax cross-entropy and heavy-tailed PLDA back-end adapted to SRE18 Eval. When training the embedding with AM-softmax, we improved by 17% relative. As AM-softmax loss includes length normalization, we didn't need HT-PLDA an we just used G-PLDA. Then, doubling the ResNet channels, we improved by another 5%. Finally, we tried to fine-tune the network with in-domain data and just use cosine scoring. However, this degraded by 32%.

7.4. Fusion submissions

Table 3 shows the results for our best fusion submissions. Three strategies were explored for the fusion: A) individual system calibration plus score averaging; B) individual system calibration plus manual system weight tuning + post-calibration; and C) trained fusion with large L2 regularization plus post-calibration with low L2 regularization. The small dev set made difficult training a reliable fusion/calibration. For this reason, we resorted to two-step processes. The best JHU-HLTCOE submission included systems COE-{1-5}-PLDA + COE-{3-4}-D-Cosine with fusion A). Meanwhile, JHU-MIT submission included COE-{1-4}-PLDA + COE-4-D-Cosine + CLSP18-1 + CLSP19-{1,2}. JHU-MIT best in progress used fusion B) and best in eval used fusion C). The best fusion improved EER and Cprimary by 18% w.r.t. the best single system.

8. Results Audio-Visual

8.1. Audio

8.1.1. Single systems

Table 4 shows results for single systems in the audio only track of the Audio-Visual evaluation. DCF is measured at target prior 0.05. We used SRE18 VAST Eval as main development set since we decided that SRE19 AV dev was too small to obtain meaningful conclusions. We observe that SRE19 Eval is much easier than SRE18 VAST (SRE19 DCF 3× lower than in SRE18). We speculate that this could be because of bad diarization marks in the enrollment side, enrollment diarization marks were refined for SRE19[12]. JHU-HLTCOE ResNet34 (COE-1) was the best for all metrics and datasets. This x-Vector network was fine-tuned with full-length recordings, which allowed us to use cosine scoring instead of PLDA. However, this system suffered some miscalibration on the SRE19 AV Eval. Because of this miscalibration, the deep/narrow F-TDNN (CLSP-2) and ResNet34 with 8 clusters LDE (CLSP-4) attained actual DCF very close to COE-1. However, COE-1 was significantly better

[12]Communication from LDC at SRE19 workshop

Table 3: *Submission results on CTS condition.*

Submission	SRE18 CTS Dev			SRE19 CTS Progress			SRE19 CTS Eval		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
JHU-HLTCOE Best in Prog./Eval	3.30	0.162	0.181	**2.34**	**0.176**	**0.177**	**2.19**	**0.202**	**0.205**
JHU-MIT Best in Prog.	3.31	**0.139**	**0.163**	2.40	0.186	0.189	2.24	0.213	0.231
JHU-MIT Best in Eval	3.30	0.146	0.167	2.36	0.189	0.194	2.23	0.209	0.219

Table 4: *Single audio systems results on SRE18 VAST Eval and SRE19 AV*

System	Arch	Layers	Chann./Emb. dimension	Loss	Back-end	SRE18 VAST Eval			SRE19 AV Dev			SRE19 AV Eval		
						EER(%)	Min Cp	Act Cp	EER(%)	Min Cp	Act Cp	EER	Min Cp	Act Cp
COE-1	ResNet2D	34	64/256	AM-Softmax	D-Cosine	**8.04**	**0.258**	**0.262**	**4.69**	**0.172**	**0.185**	**1.76**	**0.065**	**0.101**
CLSP-1	F-TDNN	12	2048/512	Softmax	LDA-LN-GPLDA	10.18	0.33	0.339	5.84	0.205	0.239	2.58	0.108	0.117
CLSP-2	F-TDNN	16	725/512	Softmax	LDA-LN-GPLDA	10.35	0.349	0.351	5.87	0.214	0.239	2.55	0.105	0.109
CLSP-3	ResNet2D-LDE64	34	16/400	AAM-Softmax	LDA-LN-GPLDA	10.21	0.358	0.367	6.64	0.25	0.277	3.06	0.118	0.128
CLSP-4	ResNet2D-LDE8	34	64/512	AAM-Softmax	LDA-LN-GPLDA	10.59	0.322	0.329	5.09	0.198	0.201	2.59	0.102	0.106
MITLL-1	E-TDNN	12	512/512	Softmax	LDA-LN-DPLDA	13.04	0.460	0.469	9.21	0.328	0.344	4.89	0.228	0.250

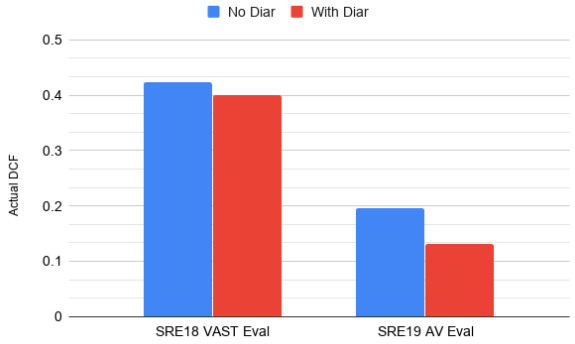

Figure 5: Diarization impact on SRE18 VAST and SRE19 AV

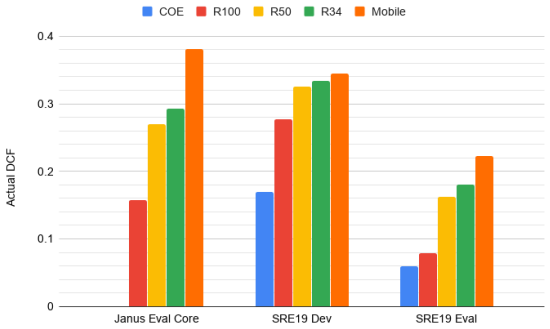

Figure 6: ArcFace embeddings comparison on Janus and SRE19 AV

in EER and minimum DCF. We also tried wide F-TDNN with 40M parameters (CLSP-1), which improved on the SRE18 Eval but did not improve on the SRE19 AV Eval.

8.1.2. S-Norm and diarization

All the systems in our submissions included adaptive S-Norm with cohorts from SITW, SRE18 VAST dev and segments extracted from Dihard II evaluation. Figure 4 shows the impact of S-Norm. For this analysis we used the wide F-TDNN in CLSP-1 system. The Dihard II data helped to introduce a wider variability in the cohort and improved SRE18 by 9% and SRE19 by 13%. The total gain from AS-Norm was around 17% for both SRE18 and SRE19 Eval.

Figure 5 shows the impact of diarization. Diarization only improved by 5% in SRE18 while it improved by 32% in SRE19. We wonder if this difference can also be due to bad diarization marks on the SRE18 enrollment side.

8.1.3. Submissions

Table 5 shows the results for our fusion submissions. We used the same greedy fusion scheme that we used last year [11, 12]. The JHU-MIT primary used systems COE-1 and CLSP-{1,2,3} in Table 4. The JHU-MIT contrastive also added MITLL-1. Though CLSP-4 was also a good system was not included in the fusion since it did not provide gains according to our fusion squeme. JHU-HLTCOE primary submission was the single system COE-1. The fusion improved EER and DCF by 16% and 14% relative respectivey w.r.t. the best single system.

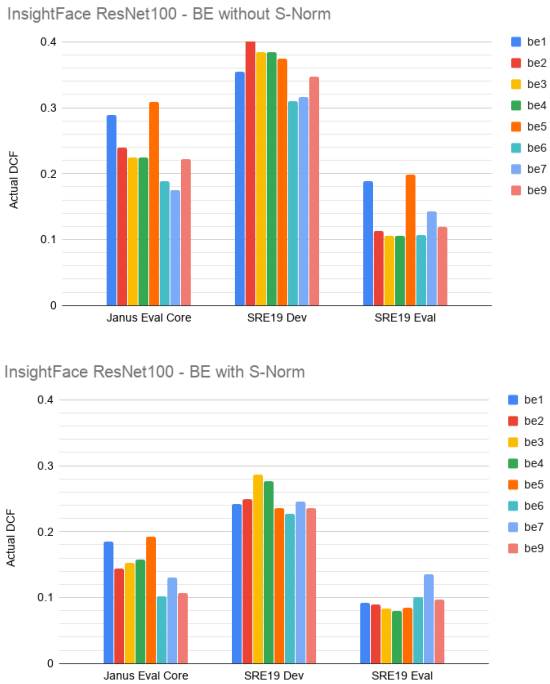

Figure 7: Video back-ends comparison on Janus and SRE19 AV

Table 5: *Submitted audio systems results on SRE18 VAST Eval and SRE19 AV*

System	SRE18 VAST eval			SRE19 AV dev			SRE19 AV eval		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
JHU-MIT Primary	**7.64**	0.247	0.250	**4.50**	0.177	**0.183**	**1.48**	**0.062**	**0.087**
JHU-HLTCOE Primary	8.04	0.258	0.262	4.69	**0.172**	**0.185**	1.76	**0.065**	0.101
JHU-MIT Contrastive	**7.65**	**0.243**	**0.243**	4.51	**0.171**	**0.186**	1.51	**0.065**	**0.090**

Table 6: *Submitted video systems results on SRE19 AV.*

Systems	SRE19 AV Dev			SRE19 AV Eval		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
JHU-MIT Primary	5.12	**0.140**	**0.140**	1.87	**0.051**	**0.052**
JHU-HLTCOE Prim/Single	6.02	0.185	**0.2**	2.21	0.057	0.059
JHU-CLSP Single	9.73	0.219	0.236	4.87	0.096	0.097
JHU-MIT Contrastive	**5.00**	**0.140**	0.149	**1.41**	**0.051**	0.054

Table 7: *Submitted multi-modal systems results on SRE19 AV.*

Systems	SRE19 AV Dev			SRE19 AV Eval		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
JHU-MIT Primary	1.89	0.037	0.054	0.44	0.011	0.018
JHU-HLTCOE Prim/Single	**1.12**	**0.034**	**0.350**	0.44	**0.008**	**0.012**
JHU-MIT Single	2.63	0.056	0.070	0.88	0.030	0.049
JHU-MIT Contrastive	1.75	0.037	0.042	**0.22**	0.010	0.017

8.2. Visual

8.2.1. ArcFace embedding analysis

Figure 6 compares performance of several ArcFace embeddings. COE embedding (blue) used ResNet101 PyTorch third party implementation and back-end with average in enrollment side and AHC in test side. The rest of embeddings were original InsightFace implementation with different ResNet network sizes and back-end with median in enrollment and AHC in test. PyTorch re-implementation clearly outperformed InsightFace version in SRE19. For InsignthFace versions, network size had a significant impact in performance. ResNet100 was 51% better than ResNet50 on SRE19 Eval.

8.2.2. JHU-CLSP back-end analysis

Figure 7 compares the video back-end versions evaluated by the JHU-CLSP team. For this experiment we used InsightFace ResNet100. Averaging all back-ends, score normalization improved DCF by 35%, 33% and 29% relative for Janus, SRE19 Dev and SRE19 Eval respectively. Back-end 4 (enroll-median vs test-AHC, green) was the best for SRE19 Eval but not in the other two datasets. Back-ends with self-attention (be6-cyan, b9-pink) performed well across datasets.

8.2.3. Submissions

Table 6 shows the results of our submissions to the AV-video condition. The JHU-MIT primary was a fusion of JHU-HLTCOE system plus 3 JHU-CLSP systems (R100-be9-SNorm, R100-be4, Mobile-be1). Systems for the fusion were selected using a greedy fusion scheme trained on SRE19 Dev. JHU-HLTCOE primary submission was a single system. JHU-MIT single was ResNet100-be9-SNorm. JHU-MIT contrastive was JHU-HLTCOE system plut 20 JHU-CLSP systems. Primary fusion improved DCF by 11% w.r.t. JHU-HLTCOE system.

8.3. Multi-modal

Table 7 shows the results of our submissions to the multi-modal AV condition. Each submission was obtained by adding the scoring of the corresponding audio and video conditions. Thus, we assumed independence between modalities. The fusion of modalities provided huge gains. For JHU-MIT primary, it improved DCF by 79% w.r.t. audio and 65% w.r.t. video. For JHU-HLTCOE single system, it improved by 88% w.r.t. audio and 83% w.r.t. video. Figure 8 compares primary fusions and single systems for different modalities. We can see that video outperformed audio. Multi-modal fusion improved by 88% w.r.t. single audio system.

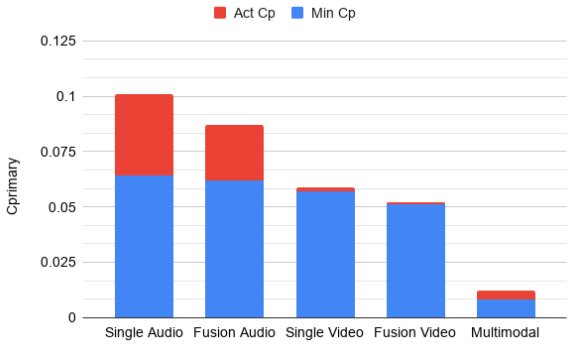

Figure 8: SRE19 AV Eval modality comparison

9. Conclusions

We analyzed the JHU-MIT systems for NIST SRE19. The results confirmed that using deeper network and large amount of training data was required to obtain the best results. 2D residual networks and PyTorch implementations outperformed Kaldi TDNN versions for the first time in a NIST evaluation. We think that PyTorch improved because it offers full parallel Multi-GPU training, which helps to train networks more efficiently and faster. Also, results seem to confirm the utility of large margin losses (AM-softmax, AAM-softmax).

For the CTS condition, we improved w.r.t. SRE18 result. Significant improvement was obtained by using SRE18 Eval labeled data for back-end adaptation. Generative PLDA back-end performed better than discriminative versions due to easy adaptation. We obtained small improvement by in-domain adaptation of x-vector network.

For the Audio-Visual audio only condition, x-vector fine-tuning with full length recordings combined with cosine scoring performed the best avoiding the need for PLDA. In the video only condition, out-of-the-box pre-trained face-detectors/embeddings and simple back-ends were successful. The multi-modal fusion provided huge gains w.r.t. single modality in the range of 80% of improvement.

10. References

[1] G. Doddington, "The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, no. 2-3, pp. 225–254, jun 2000.

[2] M. Przybocki, A. Martin, and A. Le, "NIST Speaker Recognition Evaluations Utilizing the Mixer Corpora - 2004, 2005, 2006," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 1951–1959, sep 2007.

[3] L. Brandschain, D. Graff, C. Cieri, K. Walker, and C. Caruso, "The Mixer 6 Corpus: Resources for Cross-Channel and Text Independent Speaker Recognition," in *LREC10*, Valletta, Malta, may 2010, pp. 2441–2444.

[4] J. Villalba, E. Lleida, A. Ortega, and A. Miguel, "The I3A Speaker Recognition System for NIST SRE12: Post-evaluation Analysis," in *Interspeech 2013*, Lyon, France, aug 2013, pp. 3679 – 3683.

[5] S. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, D. Reynolds, E. Singer, L. Mason, and J. Hernandez-Cordero, "The 2016 NIST Speaker Recognition Evaluation," in *Interspeech 2017*, ISCA, aug 2017, pp. 1353–1357.

[6] S. Sadjadi, C. Greenberg, D. Reynolds, E. Singer, L. Mason, and J. Hernandez-Cordero, "The 2018 NIST speaker recognition evaluation," in *Interspeech 2019*, Graz, Austria, aug 2019, pp. 1483–1487.

[7] J. Tracey and S. Strassel, "VAST : A Corpus of Video Annotation for Speech Technologies Main corpus Subcorpora," in *LREC 2018*, Miyazaky, Japan, may 2018, pp. 4318–4321.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors : Robust DNN Embeddings for Speaker Recognition," in *ICASSP 2018*, Alberta, Canada, apr 2018, pp. 5329–5333.

[9] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Odyssey 2010*, Brno, Czech Republic, jul 2010.

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *CVPR 2019*, 2019.

[11] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin, R. Dehak, L. Garcia-Perera, D. Povey, P. Torres-Carrasquillo, S. Khudanpur, and N. Dehak, "State-of-the-art Speaker Recognition for Telephone and Video Speech: the JHU-MIT Submission for NIST SRE18," in *INTERSPEECH 2019*, Graz, Austria, sep 2019.

[12] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. García-Perera, F. Richardson, R. Dehak, P. Torres-Carrasquillo, and N. Dehak, "State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations," *Computer Speech & Language*, vol. 60, pp. 101026, mar 2020.

[13] G. Sell, K. Duh, D. Snyder, D. Etter, and D. Garcia-Romero, "Audio-Visual Person Recognition in Multimedia Data From the Iarpa Janus Program," in *ICASSP 2018*, apr 2018, vol. 2018-April, pp. 3031–3035.

[14] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *INTERSPEECH 2017*, Stockholm, Sweden, aug 2017, pp. 999–1003.

[15] S. Chaudhuri, J. Roth, D. Ellis, A. Gallagher, L. Kaver, R. Marvin, C. Pantofaru, N. Reale, L. Reid, K. Wilson, and Z. Xi, "AVA-Speech: A Densely Labeled Dataset of Speech Activity in Movies," in *Interspeech 2018*, Hyderabad, India, aug 2018, pp. 1239–1243.

[16] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-Speaker Conversations Using X-Vectors," in *ICASSP 2019*, Brighton, UK, may 2019.

[17] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in *INTERSPEECH 2018*, Hyderabad, India, sep 2018.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," dec 2015.

[19] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large Margin Cosine Loss for Deep Face Recognition," in *CVPR 2018*, jun 2018, pp. 5265–5274.

[20] W. Cai, J. Chen, and M. Li, "Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System," in *Odyssey 2018*, Les Sables d'Olonne, France, jun 2018, pp. 74–81.

[21] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker Diarization Using Deep Neural Network Embeddings," in *ICASSP 2017*, New Orleans, LA, USA, mar 2017, pp. 4930–4934.

[22] A. McCree, G. Sell, and D. Garcia-Romero, "Speaker Diarization Using Leave-One-Out Gaussian PLDA Clustering of DNN Embeddings," in *Interspeech 2019*, Graz, Austria, sep 2019, pp. 381–385.

[23] A. Silnova, N. Brümmer, D. Garcia-Romero, D. Snyder, and L. Burget, "Fast Variational Bayes for Heavy-tailed PLDA Applied to i-vectors and x-vectors," in *Interspeech 2018*, Hyderabad, India, 2018, pp. 72–76.

[24] D. Garcia-Romero, D. Snyder, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "x-Vector DNN Refinement with Full-Length Recordings for Speaker Recognition," in *Interspeech 2019*, Graz, Austria, sep 2019, pp. 1493–1496.

[25] Md J. Alam, G. Bhattacharya, and P. Kenny, "Speaker Verification in Mismatched Conditions with Frustratingly Easy Domain Adaptation," in *Odyssey 2018*, Les Sables d'Olonne, France, jun 2018, pp. 176–180.

[26] D. Garcia-Romero and A. McCree, "SUPERVISED DOMAIN ADAPTATION FOR I-VECTOR BASED SPEAKER RECOGNITION," in *ICASSP 2014*, Florence, Italy, may 2014, pp. 4075–4079.

[27] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage Dense Face Localisation in the Wild," may 2019.

[28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, oct 2016.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *NIPS 2017*, 2017, pp. 5998—-6008.