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

Advances in Cross-Lingual and Cross-Source Audio-Visual Speaker Recognition: The JHU-MIT System for NIST SRE21

Jesús Villalba[1,2,3], *Bengt J. Borgstrom*[4], *Saurabh Kataria*[1,2,3], *Magdalena Rybicka*[1,5], *Carlos D. Castillo*[3], *Jaejin Cho*[1,3], *L. Paola García-Perera*[1,2], *Pedro A. Torres-Carrasquillo*[4], *Najim Dehak*[1,2,3]

[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]Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, USA
[4]MIT Lincoln Laboratory, Lexington, MA, USA
[5]AGH University of Science and Technology, Institute of Electronics, Krakow, Poland
{jvillalba,skatari1,mrybick1,carlosdc,jcho52,lgarci27,ndehak3}@jhu.edu,
{jonas.borgstrom,ptorres}@ll.mit.edu,

Abstract

We present a condensed description of the joint effort of JHU-CLSP/HLTCOE, MIT-LL and AGH for NIST SRE21. NIST SRE21 consisted of speaker detection over multilingual conversational telephone speech (CTS) and audio from video (AfV). Besides the regular audio track, the evaluation also contains visual (face recognition) and multi-modal tracks. This evaluation exposes new challenges, including cross-source–i.e., CTS vs. AfV– and cross-language trials. Each speaker can speak two or three languages among English, Mandarin and Cantonese. For the audio track, we evaluated embeddings based on Res2Net and ECAPA-TDNN, where the former performed the best. We used PLDA based back-ends trained on previous SRE and VoxCeleb and adapted to a subset of Mandarin/Cantonese speakers. Some novel contributions of this submission are: the use of neural bandwidth extension (BWE) to reduce the mismatch between the AFV and CTS conditions; and invariant representation learning (IRL) to make the embeddings from a given speaker invariant to language. Res2Net with neural BWE was the best monolithic system.

We used a pre-trained RetinaFace face detector and ArcFace embeddings for the visual track, following our NIST SRE19 work. We also included a new system using a deep pyramid single shot face detector and face embeddings trained on Crystal loss and probabilistic triplet loss, which performed the best. The number of face embeddings in the test video was reduced by agglomerative clustering or weighting the embedding based on the face detection confidence. Cosine scoring was used to compare embeddings. For the multi-modal track, we just added the calibrated likelihood ratios of the audio and visual conditions, assuming independence between modalities. The multi-modal fusion improved Cprimary by 72% w.r.t. audio.

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 and test speakers are the same. Over the years, SRE has evolved from telephone speech [2], to far-field microphone [3, 4], to non-English telephone speech [5, 6, 7], and multi-modal evaluations on internet videos [6, 7]. NIST SRE21[1] consisted of a multi-modal/multi-language/multi-source evaluation including conversational telephone speech (CTS) and audio from videos (AfV). NIST SRE21 included cross-source–i.e., enrollment on CTS and test on AfV–, and cross-language trials, which are novel challenges w.r.t. previous evaluations. As in SRE19 [7], SRE21 included audio, visual (face recognition) and multi-modal tracks. Unlike SRE19, face enrollment consisted of a single picture of the target speaker; and the test videos contained a single subject, so speaker/face diarization was not required.

In this paper, we analyze the JHU-MIT submission to NIST SRE21. This is the joint effort of teams at Johns Hopkins CLSP/HLTCOE, MIT Lincoln Laboratory and AGH. We built on the knowledge acquired building speaker and face recognition systems for previous evaluations [8, 9, 10]. For audio, we had pipelines working at 8 kHz–downsampling AfV data from 16 to 8 kHz; and 16 kHz–upsampling CTS data. We upsampled using either a linear low-pass filter or neural bandwidth extension. We used a novel neural upsampler in time-domain based on TasNet [11] and conditional generative adversarial networks. This system can predict the missing information in the upper band (4-8 kHz) and reduce the mismatch between CTS and AfV data. In terms of embeddings, we moved from

[1] https://www.nist.gov/itl/iad/mig/nist-2021-speaker-recognition-evaluation-sre21

This material is based upon work supported by the Department of Defense under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Defense.

Magdalena Rybicka was supported by the Foundation for Polish Science under grant number First TEAM/2017-3/23 (POIR.04.04.00-00-3FC4/17-00) which is co-financed by the European Union.

10.21437/Odyssey.2022-30

the F-TDNN and ResNets, used in [10], to Res2Net [12] and ECAPA-TDNN [13], which has shown superior performance in AfV data. We tried to make embeddings robust to language mismatch by using invariant representation learning (IRL) [14]. For the video track, we used pre-trained face detectors and ArcFace embeddings [15], as we did for SRE19 [10]. We also added new embeddings using single-shot detectors [16] and embeddings trained on a combination of crystal and probabilistic triplet losses [17], which performed the best on the evaluation set.

2. Datasets

2.1. Train Audio Fixed

The audio track proposed fixed and open training conditions. This paper does not discuss our open submission because it did not provide any interesting insights. The fixed condition included the following datasets:

- **NIST SRE-CTS Superset Train:** Large-scale telephony dataset compiling SRE 1996-2012 [18]. We removed 99 Mandarin(CMN)/Cantonese(YUE) speakers to use them in our development sets. In total, it contained 593517 utterances from 6769 speakers.
- **NIST SRE16 Dev:** It contains 668 recordings from 10 Mandarin speakers and 659 recordings from 10 Cebuano speakers [5].
- **NIST SRE16 Eval 60%:** It contains 60% of the speakers in the NIST SRE16 Eval. It comprises 3299 recordings from 60 Cantonese speakers and 2904 recordings from 61 Tagalog speakers.
- **VoxCeleb 1+2:** It contains 7365 speakers from audio from video [19]. The original distribution of VoxCeleb split each video into multiple short excerpts. We concatenated all excerpts from the same video into one file. This makes the dataset more appropriate for PLDA training and helps balance each video's weight in the embedding training. After concatenation, we obtain 173088 recordings.

We also considered two subsets containing Mandarin and Cantonese speakers. These subsets also kept the recordings from these speakers in other languages, i.e., English.

- **SRE-CHN**: CMN/YUE speakers from the SRE Superset and SRE16. These were selected using the provided language labels.
- **Vox-CHN**: CMN/YUE speakers from VoxCeleb. We trained a language identifier on the SRE Superset to compute language labels for VoxCeleb.

For x-vector training, we augmented speech on the fly with noise and reverberation. Impulse responses were obtained from the Aachen impulse response database (AIR)[2]. Noises were from the MUSAN corpus[3]. We used the same SNR levels as in the Kaldi recipes. For training the back-ends, we did not use any data augmentation.

2.2. Train Visual

Pre-trained RetinaFace detectors were trained on WideFace[4]. ArcFace embeddings were trained on the MS-Celeb-1M [20] (3.8M faces over 85k subjects). For the RG1 embeddings, we used Universe face dataset, a combination of curated MS-Celeb-1M, UMDFaces [21] and UMDFacesVideos datasets (5.8M faces over 58k subjects).

2.3. Development datasets

We prepared three development datasets for monitoring performance, calibration, and fusion¿

- **NIST SRE16 Eval YUE40%:** It contains 40% of the speakers (40) in the NIST SRE16 Eval set. We used the same trial list as in the original SRE16 but kept only the trials involving those 40 speakers.
- **NIST SRE-CTS Superset dev**: This set contains 99 CMN/YUE speakers from the NIST SRE CTS superset. This set is balanced with 25/25 male/female CMN speakers and 25/25 male/female YUE speakers. It contains 10349 enrollment segments, 4312 test segments, and 22M trials. We covered all enrollment conditions (1, 3 segments) and all possible language pairings.
- **NIST SRE21 Dev**: This is the SRE21 development set provided by the organization. It contains 20 speakers with 193k audio trials and 38.9k audio-visual trials.

SRE-CTS superset dev and SRE21 Audio dev were used for individual audio system calibration. SRE21 dev was used to train the final audio fusion. NIST SRE21 Visual Dev was used to train face recognition calibration and fusion.

3. Audio embeddings

3.1. Acoustic features and VAD

The acoustic features were 80 and 64 log-Mel-filter banks for 16 kHz and 8 kHz systems respectively. Features were short-time mean normalized with a 3 seconds window. Silence frames were removed using Kaldi energy VAD.

3.2. Architectures

All the x-vector architectures follow the x-vector scheme [22, 23]. In essence, the embedding network consists of an encoder that extracts frame-level discriminant embeddings, a pooling mechanism, and a classification head. We tried several encoder architectures and used either statistics pooling (mean+stddev) [22] or channel-wise attentive statistics pooling [13]. The network is trained to minimize the categorical cross-entropy loss of the predicted speaker posteriors. We used additive angular margin softmax loss [15] in all our networks. Following, we describe the encoder architectures.

3.2.1. Res2Net50

This encoder is based on the original ResNet50 architecture proposed in [24]. ResNet50 has an input stem layer followed by 16 Bottleneck residual blocks like the one in Figure 1(left). These blocks are based on 2D convolutions. This architecture downsamples the feature maps $\times 3$ with a stride of 2 ($8\times$ total downsampling), at the same time that duplicates the number of channels in the convolutions. Res2Net [12] replaces the standard bottleneck blocks with Res2Net blocks in Figure 1(right). Res2Net divides the channels in the bottleneck layer into several groups–this is known as the scale parameter. Each group (except the first one, which is just copied in the output) passes through a 3×3 convolution and is added to the input of the convolution of the next group. Hence, each

[2]http://www.openslr.org/resources/28
[3]http://www.openslr.org/resources/17
[4]http://shuoyang1213.me/WIDERFACE/WiderFace_Results.html

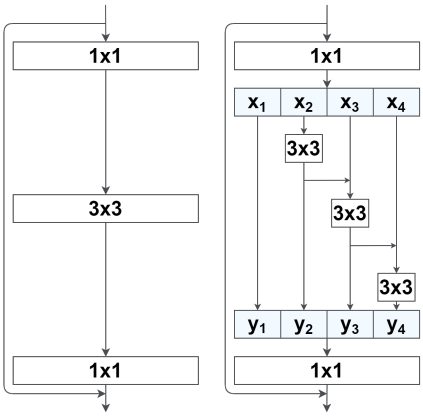

Figure 1: ResNet50 standard bottleneck blocks (left) and Res2Net50 bottleneck block (right)

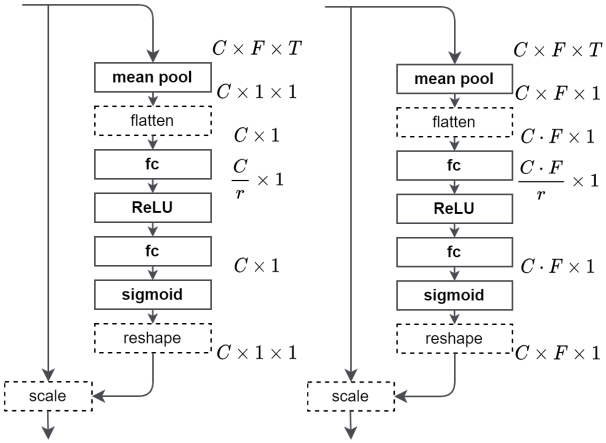

Figure 2: Standard squeeze-excitation (SE) (left) and temporal squeeze-excitation (TSE) (right)

group observes a different receptive field. We set the scale to 8, and the number of channels in each group (width) was set to 26 for the first Res2Net block, following [12]. Each time the network downsamples the feature maps, we duplicate the width of the Res2Net groups. The output of this network is a four-dimensional tensor $(B, C, F/8, T/8)$, where B is batch-size, C is the number of channels, F is the number of Mel filters, and T is time. Channel and frequency dimensions are flattened to $(B, C \times F/8, T/8)$ before passing the features to the pooling layer [25]. This architecture has been used for speaker recognition in [26, 27].

3.2.2. TSE-Res2Net50

This encoder adds squeeze-excitation (SE) [28] blocks to Res2Net50. The original SE in 2D ResNets, in Figure 2 (left), performs a pooling operation in both time and frequency dimensions (spatial dimensions in image). Then, it applies a scaling to the feature maps, which is channel-dependent but the same for all the frequency dimensions. We observed that standard SE does not provide significant gains for speaker recognition. In [29, 27], we proposed temporal squeeze-excitation (TSE), depicted in Figure 2 (right). TSE pools only in the temporal axis and applies a different scaling for each channel and frequency dimension. Our TSE-Res2Net used scale=4 and width=26.

3.2.3. ECAPA-TDNN

ECAPA-TDNN [13] can be regarded as a TSE-Res2Net with 1D convolutions. Following [30], we used a network with 4 Res2Net blocks with 2048 channels each. The number of channels is kept fixed along the network. This network does not downsample the feature maps. Instead, it uses dilated convolutions to increase the receptive field of the network. Additionally, the output of the Res2Net blocks is concatenated at the encoder output and projected with a kernel=1 convolution.

3.3. Training procedure

All networks were first trained on 4 second chunks using an effective batch-size of 512. The actual batch size depended on the GPU memory and network size, and gradient accumulation was used to achieve the desired effective batch size. The learning rate was set to 0.02 and kept constant for 40k model updates. After that, it was divided by two every 10k steps until convergence. Later, the networks were fine-tuned using cyclic cosine learning rate scheduling on longer utterances (10-15 second chunks except for Res2Net50-scale=8, which uses 10 second chunks).

3.4. JHU-Fixed

Here, we summarize the networks included in our fusions. Unless indicated otherwise, they were trained on all the fixed condition data, used channel-wise attentive statistics pooling, and SoX for linear upsampling. However, the networks with Vox suffix were just trained on VoxCeleb to be used on AfV-AfV trials, and used statistics pooling.

- **8 kHz Networks:** Res2Net50.
- **16 kHz Networks:** TSE-Res2Net50, TSE-Res2Net50-IRL–it fine-tunes the previous TSE-ResNet50 with invariant representation learning (IRL) as described in Section 6–, Res2Net50-Neural-BWE–it was fine-tuned on data upsampled by neural bandwidth extension as described in Section 5–, Res2Net50-Vox, TSE-Res2Net50-Vox, ECAPA-TDNN.

3.5. MITLL-Fixed

The MITLL-v1 used an 8 kHz Res2Net50 network with statistics pooling. It was trained on the VoxCeleb 1+2 and SRE-CTS data sets, and the training included a fine-tuning stage using longer duration segments.

4. Language Identification

Since VoxCeleb and SRE21 eval do not have language labels, we built a language identification system. The system was based on a ThinResNet34 neural network ($4\times$ fewer channels than standard ResNet34). The network was trained on the SRE CTS superset. Then, we used the network logits to label VoxCeleb and SRE21 eval. For SRE21, we restricted the decision to English, Mandarin, and Cantonese. For VoxCeleb, we allowed any language. This system was trained on our 16 kHz setup. However, we restricted the Mel filter banks to a maximum frequency of 3.9 kHz. Therefore, we just considered the lower band to perform the classification.

5. Bandwidth Extension

We developed a bandwidth extension (BWE) model to upsample the narrow-band signals of training and test sets. Similar to

our previous work [31], the goal is to reduce mismatch among narrow-band and wide-band signals. Our BWE model was a Conditional Generative Adversarial Network (cGAN), consisting of a generator and a discriminator network. The generator follows the ConvTasNet [11] architecture, which processes the signals in the time domain. It consists of an encoder, separation, and decoder stage. The encoder stage consists of a 1-D convolutional block. The separation stage estimates a mask, which is multiplied by the encoded representations. Finally, a 1-D convolutional block transforms the masked representations back to the time-domain in the decoding stage. We used the open-source ConvTasNet[5] with a network depth of eight, and one layer stack. We used the discriminator of Parallel WaveGAN (PWG) [32][6]. This is a 10-layer deep CNN based on 1-D convolutions and LeakyReLU activations. We change the number of channels to 80.

The cGAN is a supervised learning model trained on (wide-band, narrow-band) pairs from VoxCeleb. To create its narrow-band counterpart, we removed the upper-band information by downsampling and subsequent linear upsampling. The cGAN loss is

$$\mathcal{L}_{\text{cGAN}}(x_n, x_w) = \mathcal{L}_{\text{adv}}(x_n, x_w) + \lambda_{\text{sup}}\mathcal{L}_{\text{sup}}(x_n, x_w) \quad (1)$$

$$\mathcal{L}_{\text{adv}}(x_n, x_w) = \max_{\mathcal{G}} \min_{\mathcal{D}} \mathbb{E}[(1 - \mathcal{D}(x_w))^2] \quad (2)$$

$$+ \mathbb{E}[(\mathcal{D}(\mathcal{G}(x_n)))^2] \quad (3)$$

$$\mathcal{L}_{\text{sup}}(x_n, x_w) = ||\mathcal{G}(x_n) - x_w||_1 \quad (4)$$

where λ_{sup}=0.01, x_n is the simulated narrow-band sample, x_w is the corresponding wide-band sample, $\mathcal{G}$ is generator network, $\mathcal{D}$ is discriminator network, $\mathcal{L}_{\text{adv}}$ is the Least-Squares GAN (LSGAN) [33] based adversarial loss, and $\mathcal{L}_{\text{sup}}$ is the L1 supervision loss. The GAN was trained using alternative gradient descent, where the generator is updated twice for each discriminator update. We used Adam optimizer with batch-size of 16 and 3 seconds audio segments.

Neural BWE improved act. C_{primary} in CTS-CTS trials from 0.235 to 0.209, and CTS-AfV cross-source trials from 0.507 to 0.450, with 11% relative improvement.

6. Language Invariant Embeddings

To reduce the detrimental effect of the cross-language trials, we incorporated the invariant representation learning method [14] to enforce language invariant embeddings. During the training process, each utterance x was paired with randomly chosen recording x' belonging to the same speaker, but containing a different language. The final loss ($\mathcal{L}_{\text{IRL}}$) was calculated as a weighted sum of the classification losses $\mathcal{L}$ of both samples and the L_2 distance between their cross-language speaker embeddings (denoted as $\mathcal{L}_{\text{d}}$),

$$\mathcal{L}_{\text{IRL}}(x, x') = \alpha\mathcal{L}(x) + \beta\mathcal{L}(x') + \gamma\mathcal{L}_{\text{d}}(x, x') \; . \quad (5)$$

where we set $\alpha = \beta = 0.06$ and $\gamma = 0.1$. This approach was to fine-tune the TSE-Res2Net50 on the SRE-CHN dataset. The network was fine-tuned using the cosine learning rate schedule described above for nine epochs.

This technique improved Act. C_{primary} for most language pairs on SRE21 dev: ENG-ENG (32%), CMN-CMN (16%), ENG-CMN (3%). However, the improvement was not translated to SRE21 eval.

[5]https://github.com/naplab/Conv-TasNet

[6]https://github.com/kan-bayashi/ParallelWaveGAN

7. Audio Back-ends

7.1. JHU-v2

The JHU-v2 back-end pipeline consisted of condition-dependent centering; global PCA dimensionality reduction; Whitening; length normalization and PLDA adapted to the Mandarin/Cantonese speakers in SRE (SRE-CHN) and Vox-Celeb (Vox-CHN). Following, we describe the steps in more detail.

First, we computed separate means and covariances for CTS ($\boldsymbol{\mu}_{\text{CTS}}, \mathbf{S}_{\text{CTS}}$) and AfV ($\boldsymbol{\mu}_{\text{AfV}}, \mathbf{S}_{\text{AfV}}$). Then, we adapted these means and covariances for each language (ENG, CMN, YUE) in SRE-CHN and Vox-CHN, obtaining six pairs $\{\boldsymbol{\mu}_{a-b}, \mathbf{S}_{a-b} | a \in \{\text{AfV}, \text{CTS}\}, b \in \{\text{ENG}, \text{CMN}, \text{YUE}\}\}$. The adapted means were used to center SRE-CHN, Vox-CHN, the dev sets, and the SRE21 eval set. The rest of the training data was centered using $\boldsymbol{\mu}_{\text{CTS}}$ and $\boldsymbol{\mu}_{\text{AfV}}$.

Second, we computed a linear projection, which jointly does PCA and whitening. The total covariance to train the projection was computed as the average of the adapted covariances,

$$\mathbf{S} = \frac{1}{6} \sum_{a \in \{\text{AfV}, \text{CTS}\}} \sum_{b \in \{\text{ENG}, \text{CMN}, \text{YUE}\}} \mathbf{S}_{a-b} \; . \quad (6)$$

In this manner, all conditions had the same weight in the PCA computation. This projection was applied to all the centered data. We observed that an aggressive PCA dimension reduction was beneficial for the SRE21 dev sets. For the networks trained on all the Fixed-condition data, PCA dimension was selected to keep 50% of the data variance. For networks trained on Voxceleb , we kept 85% of the variance.

Third, we applied length normalization on the projected and whitened data.

Fourth, we trained an SPLDA model in all the training data and adapted it to the combined SRE-CHN and Vox-CHN data. Adaptation weights were set to 0.75 for the speaker covariance, and 0.5 for the channel covariance.

Finally, we observed that Adaptive S-Norm improved C_{primary} on the SRE21 dev set. We used the top 5000 cohort segments from SRE-CHN and Vox-CHN. However, AS-Norm hurt the performance on the evaluation data.

7.2. JHU-v3

The JHU-v3 back-end uses the same embedding pre-processing as JHU-v2. However, the PLDA model was source-dependent. First, we trained an SPLDA on all the training data. Then, we adapted different PLDAs to the CTS (SRE) or AfV (VoxCeleb) conditions. Finally, we adapted the CTS PLDA to SRE-CHN; and the AfV PLDA to Vox-CHN data. Thus, we obtained different PLDA models for CTS-CTS and AfV-AfV trials. The PLDA for cross-source trials (CTS-AfV) was computed by averaging the speaker and channel covariances of the CTS and AFV PLDAs. We did not use AS-Norm with this back-end.

JHU-v3 improved Act. C_{primary} by 12% w.r.t. JHU-v2 when combined with embeddings trained on VoxCeleb. Restricting to AfV-AfV trials, it improved by 49%. It did not improve for embeddings trained on all the data.

7.3. MITLL-v1

The MITLL-v1 system used source-dependent back-ends to address the variety of audio sources included in the evaluation. Specifically, it trained separate PLDA scoring pipelines for each possible trial combination: CTS, AfV, and cross-source. Each

Table 1: *Ablation results for condition dependent calibration on SRE21 Audio*

Calibration	SRE 21 Visual dev		SRE21 Visual eval	
Condition	Min Cp	Act Cp	Min Cp	Act Cp
Indep.	0.339	0.365	0.396	0.402
Dep.	**0.309**	**0.323**	0.357	0.361
Dep. No-Lang.	0.333	0.343	**0.350**	**0.353**
Dep. No-NEnr.	0.315	0.332	0.362	0.364
Dep. No-Source	0.320	0.338	0.409	0.412

pipeline included LDA dimension reduction to 100, followed by global centering and whitening, and length normalization. A simplified PLDA (S-PLDA) model with speaker dimension 75 was then used to generate verification scores. The SRE-CHN set was used for the CTS and Cross conditions, and the VoxCeleb 1+2 and SRE-CHN sets were used for the AfV condition.

8. Audio Calibration and Fusion

8.1. JHU single system calibration

JHU trained a condition-dependent calibration where scores s were converted into log-likelihood ratios as

$$\text{LLR} = as + b + \mathbf{w}_l^{\text{T}}\mathbf{l} + \mathbf{w}_c^{\text{T}}\mathbf{c} + \mathbf{w}_e^{\text{T}}\mathbf{e} \tag{7}$$

where a and b are condition independent scaling and bias; $\mathbf{l}$, $\mathbf{c}$ and $\mathbf{e}$ are 1-hot vectors that indicate the language, source and number of enrollment segments conditions; and $\mathbf{w}_l$, $\mathbf{w}_c$ and $\mathbf{w}_e$ are trainable weights. Conditioning vectors are defined as

$$\mathbf{l} = \begin{bmatrix} \text{ENG}-\text{ENG} \\ \text{ENG}-\text{CMN} \\ \text{ENG}-\text{YUE} \\ \text{CMN}-\text{CMN} \\ \text{CMN}-\text{YUE} \\ \text{YUE}-\text{YUE} \end{bmatrix} \mathbf{c} = \begin{bmatrix} \text{CTS}-\text{CTS} \\ \text{CTS}-\text{AfV} \\ \text{AfV}-\text{AfV} \end{bmatrix} \quad \mathbf{e} = \begin{bmatrix} \text{1side} \\ \text{3sides} \end{bmatrix} . \tag{8}$$

We assumed trial symmetry, e.g., YUE-ENG is the same as ENG-YUE. This method is equivalent to having a condition independent scaling and condition dependent bias for each trial. We trained one calibration on SRE-Superset+SRE21-audio-dev (prior) and another on SRE21-audio-dev. The final calibration parameters were 0.9 $\times$ SRE21 plus 0.1 $\times$ the prior parameters.

Table 1 shows an ablation study of the conditions impacting the calibration. For SRE21 Dev, the best result was conditioning on the three factors (language, num. of enroll. segments, audio source). For SRE21 Eval, cond. dependent calibration improved by 10% w.r.t. cond. independent. However, removing the language conditioning was 2% better than using all the factors. Conditioning on the enrollment-test sources was the most critical factor. If we remove it, the result was worse than the condition independent result.

8.2. MITLL single system calibration

The MITLL-v1 system conditioned score calibration on the audio source and number of enrollment cuts (i.e., 1c vs. 3c). For the CTS back-end pipeline, separate logistic regression calibration mappings were trained for the 1c and 3c subsets of the CTS trials from the SRE21-Dev set. Similarly, for the Cross back-end, different calibrators were trained for the 1c and 3c subsets of the Cross trials from the SRE21-Dev set. For the AfV back-end, only a 1c mapping was trained for the AfV trials of the SRE21-Dev set.

8.3. Fusion

We trained a different fusion for each source condition (CTS-CTS, CTS-AfV and AfV-AfV) on SRE21 audio dev. We used a greedy fusion scheme to select the best fusion combination as last year [8]. We trained the fusion at $P_{\mathcal{T}} = 0.1$ to have enough false alarm errors. Finally, we re-calibrated the fused scores at $P_{\mathcal{T}} = 0.05$ condition-independent.

9. Audio Single Systems

Table 2 summarizes the results for the single systems that were part of our fusions. We observe that systems with score-normalization obtained lower Act. Cp compared to non-S-Norm systems on the development sets. However, S-Norm damaged performance on SRE21 Eval. We also note that the system with IRL performed the best on SRE21 dev, but the improvement did not translate to SRE21 eval, obtaining similar performance as the system without IRL. This shows that SRE21 dev is too small to make informed decisions about which system is the best. Overall, the systems at 16 kHz performed better than the systems at 8 kHz. Res2Net using neural upsampling was the best system on the eval, with 15% overall improvement w.r.t. the equivalent system using linear upsampling. We also observed that Res2Net performed better than ECAPA-TDNN. Systems trained just on VoxCeleb only performed well for AfV-AfV trials, so they did not perform well overall. However, they were used in the source-dependent fusions, as shown in Table 3.

10. Audio Submissions

Table 3 summarizes the systems that were included in our fixed condition fusions. For the primary, we chose the best fusion of four systems on SRE21 dev, without any restrictions. For the contrastive, we chose the best fusion of four systems without AS-Norm. Thus, we wanted to evaluate how significant was the effect of AS-Norm on the eval. As observed in the single systems, the contrastive without AS-Norm performed 6% better in terms of Act. C_{primary} than the primary. For our best single, we chose a different system for each source condition. This system was 3.6% better than our best monolithic system (Res2Net50 with neural BWE). The primary fusion improved Act. C_{primary} by 13% w.r.t. the best single.

11. Visual Embeddings

11.1. Pretrained InsightFace

To extract embeddings for face recognition, we used original MX-Net[7] and PyTorch[8] implementation of InsightFace's ArcFace [15] embeddings and RetinaFace [34] face detector. For enrollment, we detected a single face on the enrollment picture. For test, we detected faces over the video at a rate of 1 frame per second. Then, we aligned the faces with the facial landmarks obtained and extracted the embeddings.

[7] https://github.com/deepinsight/insightface
[8] https://github.com/foamliu/InsightFace-v3

Table 2: *Audio systems results on SRE16 YUE40%, SRE-Superset dev, and SRE21 Audio*

System			SRE16 YUE40%			SRE-Superset dev			SRE21 Audio dev			SRE21 Audio eval		
Embed.	BE	S-Norm	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
Single Fixed 8 kHz														
Res2Net50	MIT-v1	N	2.511	0.190	0.245	1.260	0.066	0.082	6.05	0.386	0.394	6.46	0.368	0.372
Res2Net50	JHU-v2	Y	**1.131**	0.129	0.147	1.250	0.062	0.063	3.84	0.279	0.283	4.37	**0.338**	0.423
		N	1.181	0.126	**0.134**	1.174	0.057	0.069	**3.72**	0.322	0.328	**4.21**	0.357	0.377
Single Fixed 16 kHz Linear Upsampling														
TSE-Res2Net50	JHU-v2	Y	1.259	0.144	0.152	1.180	0.057	0.078	5.32	0.293	0.316	5.24	0.357	0.372
		N	1.397	0.147	0.189	1.157	0.061	0.102	4.81	0.309	0.323	4.83	0.357	0.361
TSE-Res2Net50-IRL	JHU-v2	Y	1.204	0.147	0.168	1.248	0.062	0.078	4.17	**0.245**	**0.256**	5.09	0.357	0.366
		N	1.140	**0.107**	0.181	1.270	0.064	0.095	3.93	0.311	0.326	4.81	0.358	0.360
Res2Net50-Vox	JHU-v3	N	3.911	0.290	0.318	3.301	0.179	0.223	8.95	0.436	0.451	7.33	0.475	0.478
TSE-Res2Net50-Vox	JHU-v3	N	4.114	0.287	0.351	3.481	0.190	0.258	9.04	0.462	0.473	8.34	0.501	0.507
ECAPA-TDNN	JHU-v2	Y	1.497	0.129	0.168	1.340	0.071	0.088	6.24	0.364	0.386	6.03	0.382	0.413
		N	1.565	0.134	0.236	1.371	0.076	0.106	5.86	0.395	0.409	5.71	0.395	0.400
Single Fixed 16 kHz Neural Upsampling														
Res2Net50	JHU-v2	Y	1.299	0.113	0.115	1.142	**0.050**	**0.062**	4.77	0.328	0.340	4.67	**0.337**	0.391
		N	1.264	0.121	0.158	**1.119**	0.052	0.073	4.27	0.323	0.333	4.45	**0.338**	**0.353**
Submissions Fixed														
Primary									**3.20**	**0.201**	**0.206**	3.91	**0.261**	0.297
Contrastive									3.37	0.242	0.247	**3.76**	0.271	**0.278**
Single									4.00	0.242	0.249	4.14	0.322	0.340

Table 3: *Audio Submissions Summary Fixed-Condition*

Single Systems			Primary			Contrastive			Single		
Embed.	BE	S-Norm	CTS-CTS	CTS-AFV	AFV-AFV	CTS-CTS	CTS-AFV	AFV-AFV	CTS-CTS	CTS-AFV	AFV-AFV
8 kHz											
MIT	MIT-v1	N	✓	✓		✓	✓				
Res2Net50	JHU-v2	Y		✓							
16 kHz Linear Upsampling											
TSE-Res2Net50-IRL	JHU-v2	Y	✓	✓						✓	
		N					✓				
Res2Net50-Vox	JHU-v3	N			✓			✓			✓
TSE-Res2Net50-Vox	JHU-v3	N	✓		✓	✓	✓	✓			
ECAPA-TDNN	JHU-v2	N			✓	✓		✓			
16 kHz Neural Upsampling											
Res2Net50	JHU-v2	Y	✓	✓					✓		
		N			✓	✓	✓	✓			

11.2. RG1 embeddings

At evaluation time, we ran our SSD-based face detector [16] frame by frame and generated a face detection score and a face-box for each face. For each detected face, we identified 21 key points along with yaw, roll and pitch [17]. If there were multiple faces in one frame, we selected the one with the highest face detection score. We used the key points to align the face using a similarity transformation. For each aligned face, we computed the RG1 feature vector [16]. To compute these vectors, we extract 512-dimensional embeddings from a ResNet101 trained on Crystal loss. Then, a linear projection trained with probabilistic triplet loss reduces RG1 vectors to 128 dimension, making them more discriminant.

12. Visual Back-ends

12.1. AHC+Cosine Back-end

In this back-end, the embeddings obtained from the test video were clustered using agglomerative clustering (AHC) using cosine scoring with a stopping threshold=0.7. In this manner, we expect to get different clusters for different face positions. Finally, we score the enrollment embedding against all the test cluster centers and take the maximum score. We used adaptive AS-Norm using the top 1000 embeddings from JANUS dev cohort.

12.2. Weighted-Avg+Cosine back-end

We pooled the test vectors by a weighted sum depending the face detection score. The weight was $\min(0.5 \cdot \log(x/(1-x)), 7)$ where x is the face detection score. Then, we computed the dot product between the pooled test vector and the enrollment representation. For videos without faces, we set the calibrated LLR to zero.

13. Visual Calibration and Fusion

Visual systems were calibrated on SRE21 visual dev using linear logistic regression. However, given that there were only 1-2 false alarms at the target operating point, we did the following. We assumed Gaussian distributed scores and computed the mean and variance of target and non-target scores. Then, we generated enough scores from the Gaussian distributions to train the calibration. For fusion, we did the same as for calibration but using full covariance Gaussians to consider the correlation between systems.

14. Visual Single Systems and Submissions

Table 4 shows the results for SRE21 visual. The primary submission was the fusion of the three systems; and the contrastive was the fusion of MX-Net and PyTorch ArcFace embeddings. It was difficult to predict the best single given the small dev set.

Table 4: *Visual systems results on SRE21 Visual*

System	SRE 21 Visual dev			SRE21 Visual eval		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
Single systems						
MX-Net InsightFace	4.55	0.049	**0.065**	5.34	0.149	0.171
PyTorch InsightFace	2.40	0.050	0.111	2.51	0.091	0.107
RG1	**2.32**	**0.047**	0.136	**2.02**	**0.066**	**0.081**
Submissions						
Primary	**2.15**	0.052	0.071	**2.24**	**0.077**	0.179
Contrastive	2.66	**0.046**	**0.055**	2.86	0.095	0.178
Single	2.40	0.050	0.111	2.51	0.091	**0.107**

Table 5: *Audio-Visual systems results on SRE21 Audio-Visual*

System	SRE 21 AV dev			SRE21 AV eval		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
Primary	**0.50**	**0.010**	0.017	**0.62**	**0.033**	0.082
Contrastive	0.57	0.012	**0.015**	0.69	0.036	**0.065**
Single	0.70	0.015	0.019	0.73	0.047	0.086

The system with the best dev Act Cp was significantly worse than others on the eval. Eventually, the best system was the RG1 embeddings. Fusions did not help since they were difficult to calibrate.

15. Audio-Visual Submissions

Assuming well-calibrated log-likelihood ratios and independence between audio and visual modalities, the log-likelihood ratio of the audio-visual fusion is the sum of the audio and visual scores. The primary and contrastive fused the corresponding audio submissions and the visual primary. The single submission was the fusion of the audio and visual single systems. Table 5 shows the results. We found that Act. C_{primary} on SRE21 eval improved by 75% from single audio to single AV; by 72% from primary audio to AV; and by 23% from visual to AV. The calibration problem in the visual condition–given the few dev trials– limited further improvement.

16. Conclusions

We analyzed the JHU-MIT systems for the SRE21. Regarding the audio track, we found that Res2Net embeddings performed better than ECAPA-TDNN. Res2Net combined with neural upsampling performed the best overall. Invariant representation learning improved specific language pairs, but it did not improve the average metric on the SRE21 Eval. Adapting backends to CMN/YUE speakers helped. S-Norm improved the dev set but degraded the eval set. Regarding the visual track, the best embeddings were based on a combination of Crystal and probabilistic triplet loss. It was challenging to get benefits from fusion due to the limited number of dev. trials available to train fusion/calibration. Since audio and video are nearly independent modalities, multi-modal fusion improves about 70% w.r.t. audio. Recipes for some JHU systems are publicly available in the Hyperion toolkit for audio[9] and visual[10].

17. References

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

[2] Mark Przybocki, Alvin F Martin, and A. N. 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] Linda Brandschain, David Graff, Christopher Cieri, Kevin Walker, and Chris Caruso, "The Mixer 6 Corpus: Resources for Cross-Channel and Text Independent Speaker Recognition," in *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC10*, Valletta, Malta, may 2010, pp. 2441–2444.

[4] Craig S. Greenberg, Vincent M. Stanford, Alvin F. Martin, Meghana Yadagiri, George R. Doddington, John J. Godfrey, and Jaime Hernandez-Cordero, "The 2012 NIST speaker recognition evaluation," in *Interspeech 2013*, ISCA, aug 2013, pp. 1971–1975, ISCA.

[5] Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig Greenberg, Douglas Reynolds, Elliot Singer, Lisa Mason, and Jaime Hernandez-Cordero, "The 2016 NIST Speaker Recognition Evaluation," in *Interspeech 2017*, ISCA, aug 2017, pp. 1353–1357, ISCA.

[6] Seyed Omid Sadjadi, Craig S. Greenberg, Douglas A. Reynolds, Elliot Singer, Lisa Mason, and Jaime Hernandez-Cordero, "The 2018 NIST speaker recognition evaluation," in *Interspeech 2019*, Graz, Austria, aug 2019, pp. 1483–1487.

[7] Seyed Omid Sadjadi, Craig Greenberg, Elliot Singer, Douglas A. Reynolds, Lisa Mason, and Jaime Hernandez-cordero, "The 2019 NIST Speaker Recognition Evaluation CTS Challenge," in *Proceedings of Odyssey 2020- The Speaker and Language Recognition Workshop*, Tokyo, Japan, 2020.

[8] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Fred Richardson, Suwon Shon, Francois Grondin, Reda Dehak, Leibny Paola Garcia-Perera, Daniel Povey, Pedro 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 *Proceedings of the 20th Annual Conference of the International Speech Communication Association, INTERSPEECH 2019*, Graz, Austria, sep 2019.

[9] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Leibny Paola García-Perera, Fred Richardson, Réda Dehak, Pedro A. Torres-Carrasquillo, and Najim Dehak, "State-of-the-art Speaker Recognition with Neural Network Embeddings in NIST SRE18 and Speakers In The Wild Evaluations," *Computer Speech & Language*, p. 101026, oct 2019.

[10] Jesús Villalba, Daniel Garcia-Romero, Nanxin Chen, Gregory Sell, Jonas Borgstrom, Alan McCree, Leibny Paola Garcia Perera, Saurabh Kataria, Phani Sankar Nidadavolu, Pedro Torres-Carrasquiilo, and Najim Dehak, "Advances in Speaker Recognition for Telephone and Audio-Visual Data: the JHU-MIT Submission for NIST SRE19," in *Odyssey 2020 The Speaker and Language Recognition Workshop*, Tokyo, Japan, nov 2020, pp. 273–280, ISCA.

[9] https://github.com/hyperion-ml/hyperion/tree/master/egs/sre21-av-a

[10] https://github.com/hyperion-ml/hyperion/tree/master/egs/sre21-av-v

[11] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[12] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr, "Res2Net: A New Multi-Scale Backbone Architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, feb 2021.

[13] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech2020*, 2020, pp. 1–5.

[14] Davis Liang, Zhiheng Huang, and Zachary C Lipton, "Learning noise-invariant representations for robust speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 56–63.

[15] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[16] Rajeev Ranjan, Ankan Bansal, Jingxiao Zheng, Hongyu Xu, Joshua Gleason, Boyu Lu, Anirudh Nanduri, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa, "A fast and accurate system for face detection, identification, and verification," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2019.

[17] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, May 2017, pp. 17–24.

[18] Seyed Omid Sadjadi, "NIST SRE CTS Superset: A large-scale dataset for telephony speaker recognition," aug 2021.

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

[20] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European conference on computer vision*. Springer, 2016, pp. 87–102.

[21] Ankan Bansal, Anirudh Nanduri, Carlos D. Castillo, Rajeev Ranjan, and Rama Chellappa, "UMDFaces: An annotated face dataset for training deep networks," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. oct 2017, pp. 464–473, IEEE.

[22] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017*, Stockholm, Sweden, aug 2017, pp. 999–1003, ISCA.

[23] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-Vectors : Robust DNN Embeddings for Speaker Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, Alberta, Canada, apr 2018, pp. 5329–5333, IEEE.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," dec 2015.

[25] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matejka, and Oldrich Plchot, "But system description to voxceleb speaker recognition challenge 2019," in *VoxSRC Challenge workshop*, 2019.

[26] Tianyan Zhou, Yong Zhao, and Jian Wu, "Resnext and res2net structure for speaker verification," *arXiv preprint arXiv:2007.02480*, 2020.

[27] Magdalena Rybicka, Jesús Villalba, Piotr Żelasko, Najim Dehak, and Konrad Kowalczyk, "Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition," in *Interspeech 2021*, Brno, Czech Republic, aug 2021, pp. 496–500, ISCA.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[29] Saurabh Kataria, Phani Sankar Nidadavolu, Jesús Villalba, Nanxin Chen, Leibny Paola Garcia-Perera, and Najim Dehak, "Feature Enhancement with Deep Feature Losses for Speaker Verification," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, may 2020, pp. 7584–7588, IEEE.

[30] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5814–5818.

[31] Saurabh Kataria, Jesús Villalba, Piotr Żelasko, Laureano Moro-Velázquez, and Najim Dehak, "Deep feature cyclegans: Speaker identity preserving non-parallel microphone-telephone domain adaptation for speaker verification," *arXiv preprint arXiv:2104.01433*, 2021.

[32] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," 2020.

[33] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[34] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.