The Kriston AI System for the VoxCeleb Speaker Recognition Challenge 2022：Track1 &2

Qutang Cai, Guoqiang Hong, Zhijian Ye, Ximin Li, Haizhou Li

Kriston AI Lab

Presented by **Zhijian Ye**

September 22, 2022

CONTENTS

Data preparation and augmentation

Training Data

Track1 & Track2: Only use VoxCeleb2 dev dataset (1,092,009 utterances and 5,994 speakers)

Development Data

- VoxCeleb1-O
- VoxCeleb1-E
- VoxCeleb1-H
- VoxSRC22-dev

Augmentation

- Offline speaker augmentation strategy with 3-fold speed[1] (0.9,1.0,1.1; 17,982 speakers total)

- Online Kaldi-style augmentation: MUSAN noises, music, and babble and reverberation from the Room Impulse Response and Noise Database (RIR)

Features

- Fbank with {96, 104, 112, 120} for track1.

- Raw waveforms for fine-tuning models in track 2.

- No additional voice activity detection (VAD).

---

1. H. Yamamoto, K. A. Lee, K. Okabe, and T. Koshinaka, “Speaker Augmentation and Bandwidth Extension for Deep Speaker Embedding,” in Proc. Interspeech, 2019, pp. 406–410.

Model architectures: track 1

Backbone

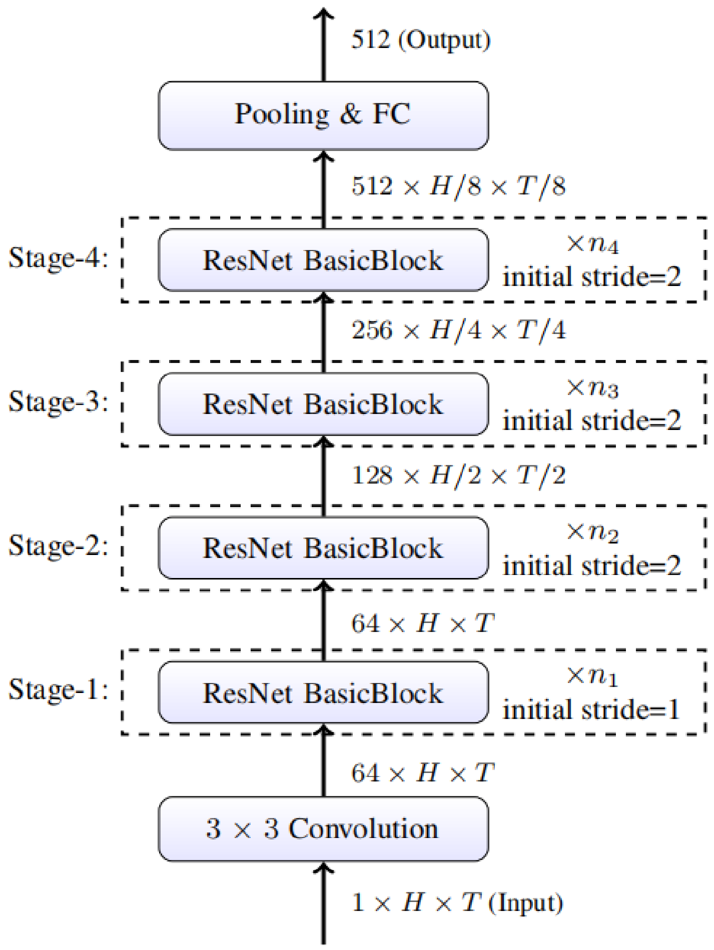


Figure 2: *Base ResNet architecture.*

ResNet variants

Name	Description
M1	Changing input feature dimension
M2	Changing model depths
M3	Changing kernel sizes
M4	Using attention mechanisms [17, 16]
M5	Using other downsampling operations [18]

Table 1: *Strategies for modifying ResNet.*

Name	M1	M2	M3	M4	M5
R1	96	$3 \times 6 \times 20 \times 3$	✗	✓	✗
R2	112	$3 \times 5 \times 14 \times 3$	✗	✓	✗
R3	120	$3 \times 6 \times 14 \times 3$	✗	✓	✗
R4	104	$3 \times 5 \times 16 \times 3$	✗	✓	✓
R5	104	$3 \times 4 \times 16 \times 3$	9	✓	✓
R6	96	$3 \times 5 \times 16 \times 3$	9	✓	✓

Table 2: *ResNet variants for Track 1.*

- We modified the ResNet architecture with one or more of the strategies listed in Table 1
- We only applied M3 and M4 to the first two stages of the backbone due to memory limits
- For M4, we used channel-wise and frequency-wise squeeze-excitation in to the residual connection, simultaneously. It's worth mentioning that we additionally introduced bias items to the input which also depend on the input like the weights items
- For M5, we altered the downsampling operation at the beginning of each stage from a 2-stride 2×2 convolution with a 2 × 2 average pooling operation.

Pooling layer

SMHA and SMHAS

- We propose a **s**huffled **m**ulti-**h**ead **a**ttention (SMHA) pooling method.

  $$\mathrm{SMHA}(x) = \mathrm{MHA}\left(\mathrm{CAT}(x, \mathrm{SHUFFLE}(x))\right)$$

  Where SHUFFLE is channel shuffle[2], CAT is the concatenation operation

- We also propose a variant of SMHA which name is **s**huffled **m**ulti-**h**ead **a**ttention with **s**tatistics (SMHAS) , where each head's statistics vector (its mean and standard deviation) is used.

- SMHAS was used for the ResNet Variants

- All the head numbers were fixed to 8

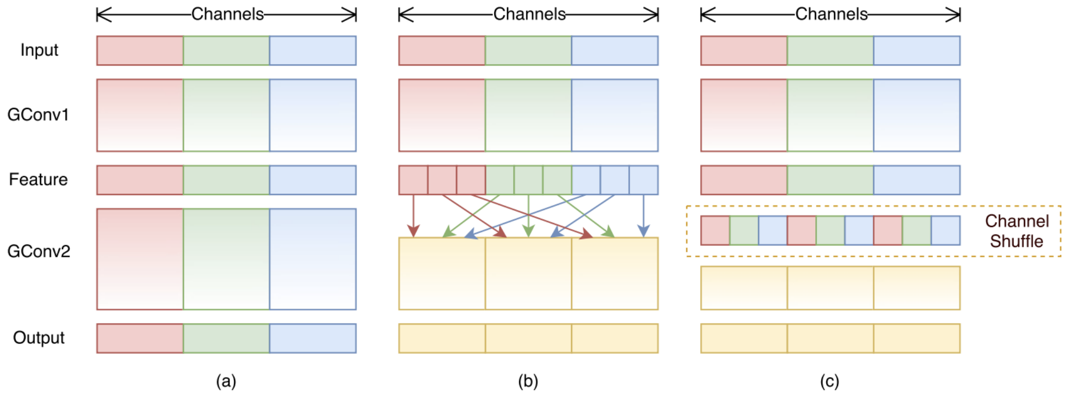


Figure 1. Channel shuffle with two stacked group convolutions. GConv stands for group convolution. a) two stacked convolution layers with the same number of groups. Each output channel only relates to the input channels within the group. No cross talk; b) input and output channels are fully related when GConv2 takes data from different groups after GConv1; c) an equivalent implementation to b) using channel shuffle.

Channel shuffle[2]

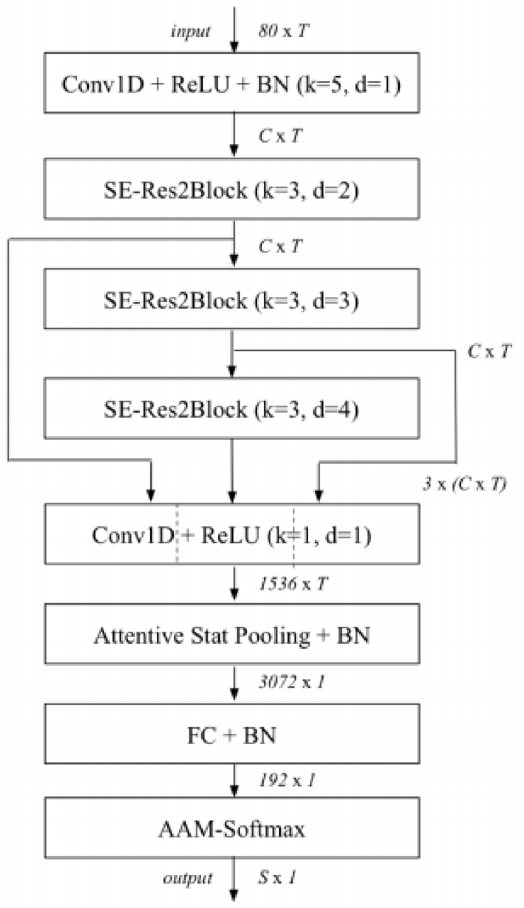
2. X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in IEEE/CVF CVPR, 2018, pp. 6848–6856.

Model architectures: track 2

- The models for Track 2 consisted of the models for Track 1
- Three fine-tuned pre-trained models
- The downstream model was ECAPA-TDNN[3]

Name	Upstream model	Pooling layer
P1	WavLM-L	SMHA
P2	XLSR-300M	STATS
P3	XLSR-1B	STATS

Table 3: *Fine-tuned pretrained models.*



Network topology of the ECAPA-TDNN[3]

3. B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA_x0002_TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in Proc. Interspeech, 2020, pp. 3830–3834.

Training procedure: track 1

Two-stage training procedure

stage-1:

- Use short utterances (2 or 2.24s)
- AM-Softmax with subcenters and inter-topK penalties[4] (subcenter number=3, margin=0.2, scale=35, inter-topK neighbor size=5, and inter-topK penalty=0.06.

stage-2 (LMF[4,5]):

- removing the speaker augmentation
- long utterances (6s)
- AAM-Softmax with subcenters (subcenter number=3, margin=0.5, scale=35)

Other settings:

- 3,000 iterations/epoch
- Batch sizes: 384 (stage 1) and 128 (stage 2)
- Optimizer: AdamW
- Lr_scheduler: ReduceLROnPlateau
- Start learning rates: 3×10^{-4}(stage 1) and 4×10^{-5} (stage 2)

---

4. M. Zhao, Y. Ma, M. Liu, and M. Xu, "The speakin system for voxceleb speaker recognition challange 2021," 2021. [Online]. Available: https://arxiv.org/abs/2109.01989
5. J. Thienpondt, B. Desplanques, and K. Demuynck, "The IDLab VoxSRC-20 submission: Large margin fine-tuning and quality aware score calibration in DNN based speaker verification," in Proc. ICASSP, 2021.

Training procedure: track 2

For P1 and P2

stage-1:

step1: Freezing the upstream models, train the downstream models, with a start learning rate of 3×10^{-4}.

step2: Unfreezing the upstream models and freezing the downstream models, train the upstream models, with a start learning rate of 4×10^{-5}.

step3: Unfreezing the whole model parameters, train the entire models, with a start learning rate of 4×10^{-5}.

stage-2 (LMF):

we trained the entire models with a start learning rate of 2×10^{-5}.

Training procedure: track 2

For P3: Due to the hardware memory limits, we trained only its self attention weights and the downstream model, alternatively.

stage-1:

step1: Freezing the upstream model, train the downstream model, with a start learning rate of 3×10^{-4}.

step2: Train the self attention weights (in the upstream model) and the downstream model alternatively for two cycles:

- **step2.1** Freezing the model parameters except the self attention parts, train the self attention weights with a start learning rate of 4×10^{-5}.
- **step2.2** Freezing the upstream model, train the downstream model with a start learning rate of 3×10^{-4}.

stage-2 (LMF):

The training steps in Stage-2 were also carried out similarly, training the self attention weights and the downstream model alternatively, except that the start learning rates were all set to 2×10^{-5}.

Scoring procedure

- Cosine similarity score was used
- AS-Norm: top 300 imposter scores were used
- QMF[4,5]: cosine score, as-norm score, duration
  VoxCeleb1-H trials was used for calibration
- Fusion: linear weighted combination where weights were picked manually
  Track1: R1—R6 were set to 1
  Track2: 1s for R1--R6
  1s for R1--R6 and P1--P2, 2 for P3

---

4. M. Zhao, Y. Ma, M. Liu, and M. Xu, “The speakin system for voxceleb speaker recognition challange 2021,” 2021. [Online]. Available: https://arxiv.org/abs/2109.01989
5. J. Thienpondt, B. Desplanques, and K. Demuynck, “The IDLab VoxSRC-20 submission: Large margin fine-tuning and quality_x0002_aware score calibration in DNN based speaker verification,” in Proc. ICASSP, 2021.

Experimental results

Table 4: *Single system evaluation results.*

System	#Params	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H		VoxSRC22-dev		VoxSRC22-test	
		EER(%)	$DCF_{0.05}$	EER(%)	$DCF_{0.05}$	EER(%)	$DCF_{0.05}$	EER(%)	$DCF_{0.05}$	EER(%)	$DCF_{0.05}$
R1	51.9M	0.3510	0.0220	0.6077	0.0321	0.9866	0.0545	1.5691	0.1110	1.812	0.1122
R2	46.7M	0.3776	0.0244	0.5860	0.0318	0.9131	0.0521	1.5350	0.1109	1.812	0.1104
R3	48.1M	0.3616	0.0241	0.6205	0.0333	0.9687	0.0560	1.5556	0.1123	-	-
R4	47.9M	0.3457	0.0299	0.5739	0.0312	0.9031	0.0511	1.5186	0.1070	-	-
R5	47.9M	0.3829	0.0271	0.5788	0.0321	0.8944	0.0499	1.5002	0.1071	-	-
R6	47.1M	0.3297	0.0272	0.5771	0.0315	0.9012	0.0512	1.5099	0.1072	-	-
P1	336M	0.3615	0.0327	0.4705	0.0278	0.9578	0.0582	1.4591	0.1000	-	-
P2	337M	0.5797	0.0523	0.4977	0.0296	0.9045	0.0539	1.4140	0.0899	1.648	0.1150
P3	986M	0.5159	0.0434	0.4525	0.0286	0.8759	0.0542	1.4163	0.0962	1.572	0.102
Fusion											
track1	R1-R6	0.2393	0.0209	0.4974	0.0266	0.8160	0.0452	1.3598	0.0977	1.401	0.090
track2	R1-P3	0.2021	0.0153	0.3481	0.0286	0.6262	0.0354	1.0468	0.0760	1.119	0.072

References

1. H. Yamamoto, K. A. Lee, K. Okabe, and T. Koshinaka, “Speaker Augmentation and Bandwidth Extension for Deep Speaker Embedding,” in Proc. Interspeech, 2019, pp. 406–410.
2. X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in IEEE/CVF CVPR, 2018, pp. 6848–6856.
3. B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA_x0002_TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification,” in Proc. Interspeech, 2020, pp. 3830–3834.
4. M. Zhao, Y. Ma, M. Liu, and M. Xu, “The speakin system for voxceleb speaker recognition challange 2021,” 2021. [Online]. Available: https://arxiv.org/abs/2109.01989
5. J. Thienpondt, B. Desplanques, and K. Demuynck, “The IDLab VoxSRC-20 submission: Large margin fine-tuning and quality aware score calibration in DNN based speaker verification,” in Proc. ICASSP, 2021.

THANK YOU