ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10448221

TB-RESNET: BRIDGING THE GAP FROM TDNN TO RESNET IN AUTOMATIC SPEAKER VERIFICATION WITH TEMPORAL-BOTTLENECK ENHANCEMENT

Sunmook Choi[⋆]*, Sanghyeok Chung*[⋆]*, Seungeun Lee*[⋆]*, Soyul Han*[†]*, Taein Kang*[†]*, Jaejin Seo*[†]*, Il-Youp Kwak*[†]*, Seungsang Oh*[⋆]

[⋆] Korea University, Department of Mathematics, Republic of Korea
[†] Chung-Ang University, Department of Statistics and Data Science, Republic of Korea

ABSTRACT

This paper focuses on the transition of automatic speaker verification systems from time delay neural networks (TDNN) to ResNet-based networks. TDNN-based systems use a statistics pooling layer to aggregate temporal information which is suitable for two-dimensional tensors. Even though ResNet-based models produce three-dimensional tensors, they continue to incorporate the statistics pooling layer. However, the reduction in spatial dimensions in ResNet due to convolution operations, including the temporal axis, raises concerns about temporal information loss and its compatibility with statistics pooling. To address this, we introduce Temporal-Bottleneck ResNet (TB-ResNet), a ResNet-based system that can utilize the nature of statistics pooling more effectively by capturing and retaining frame-level contexts through a temporal bottleneck configuration in its building blocks. The performance of TB-ResNets outperforms the original ResNet counterparts on VoxCeleb1, achieving a significant reduction in both the equal error rate and the minimum detection cost function.

***Index Terms*—** automatic speaker verification, TDNN, ResNet, statistics pooling, deep learning

1. INTRODUCTION

Automatic speaker verification (ASV) is the task of verifying the authenticity of a given test speech signal's speaker identity against an enrolled speaker. In recent times, the architecture of ASV systems has predominantly pivoted toward the domain of deep neural networks. Among these, d-vector systems integrate fully-connected layers with maxout activations [1], while x-vector systems adopt the time delay neural network (TDNN) architecture [2, 3]. It is worth mentioning that the underlying operation of TDNN is conceptually equivalent to that of one-dimensional (1D) convolution, generating intermediate layer outputs having two-dimensional (2D) tensor shape. Meanwhile, TDNN-based models utilize a statistics pooling layer with numerous studies incorporating attention mechanisms [4, 5]. This layer is used as a means of aggregating frame-level features into temporal statistics such as mean and standard deviation. A significant advancement in this direction is ECAPA-TDNN [6], which leverages 1D squeeze-excitation res2blocks and multi-layer feature aggregation to overcome limitations associated with x-vector systems.

However, current research reveals a prominent shift towards models incorporating 2D convolutional layers in ASV systems, especially based on ResNet architectures [7]. In the VoxCeleb Speaker Recognition Challenge 2022 [8], leading teams in both Tracks 1 and 2 employed various ResNet variants [9, 10]. Similar to TDNN-based models, ResNet-based models often retain the utilization of the temporal pooling technique despite the distinct three-dimensional (3D) nature of intermediate layer outputs. However, the customary reduction in spatial dimensions, including the temporal axis, within ResNet-based models raises concerns regarding a potential loss of temporal information, thereby casting doubt on its compatibility with statistics pooling methods.

In light of these observations, this paper introduces *Temporal-Bottleneck ResNet* (TB-ResNet), a novel ResNet-based system that allows us to take advantage of statistics pooling. Our proposed model is strategically designed to capture and retain frame-level contexts by employing a temporal bottleneck configuration within its building blocks, and these blocks can be naturally inserted into existing ResNet architecture. Eventually, the model enriches temporal information resulting in more meaningful statistics through the statistics pooling layer. Our novel TB-ResNet shows a remarkable enhancement in both the equal error rate and the minimum detection cost function compared to the original ResNet counterparts on VoxCeleb1 test datasets.

2. BACKGROUND

We first describe a fundamental component in TDNN-based ASV sytems: a statistics pooling method. TDNN-based models usually find frame-level features by retaining the num-

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (RS-2023-00208284) and Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00033, 50%, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity).

ber of frames, and the features are aggregated by statistics pooling across the time axis. Prior to aggregation, attention mechanisms are typically used to assign attention weights to individual frames, enhancing the quality of speaker embeddings. We illustrate the channel-dependent attentive statistics pooling proposed in the paper [6]. Next, we describe the formulation of residual blocks, which serve as the fundamental constituents of the ResNet architecture [7]. In Section 3, we adapt these residual blocks to suit our proposed model, aiming to amplify the effectiveness of statistics pooling.

2.1. Attentive statistics pooling (ASP)

For a given input speech feature sequence $\{\boldsymbol{x}_{1:T}\}$, a TDNN-based encoder generates the sequence of frame-level features $\{\boldsymbol{h}_{1:T}\} \subseteq \mathbb{R}^C$ of the same length. Before aggregating the frames, the attention score vectors $\boldsymbol{s}_t$ is computed as follows:

$$\boldsymbol{s}_t = \boldsymbol{W}_2 \,\mathrm{ReLU}(\boldsymbol{W}_1 \boldsymbol{h}_t + \boldsymbol{b}_1) + \boldsymbol{b}_2, \quad t = 1, \ldots, T. \tag{1}$$

The learnable parameters $\boldsymbol{W}_1 \in \mathbb{R}^{d\times C}$ and $\boldsymbol{b}_1 \in \mathbb{R}^{d\times 1}$ are employed for linear projection into a reduced d-dimensional feature space (in our experiment, $d = \mathrm{int}(C/8)$). The projected vector undergoes ReLU before being projected back into the original C-dimensional feature space using parameters $\boldsymbol{W}_2 \in \mathbb{R}^{C\times d}$ and $\boldsymbol{b}_2 \in \mathbb{R}^{C\times 1}$. Subsequently, attention weights $\{\alpha_{t,c}\}_{t=1}^{T}$ for each channel c are obtained by normalizing the scores $\{\boldsymbol{s}_t\}_{t=1}^{T} \subseteq \mathbb{R}^{C\times 1}$ across the time dimension:

$$\alpha_{t,c} = \frac{\exp(s_{t,c})}{\sum_{i=1}^{T} \exp(s_{i,c})}, \quad c = 1, \ldots, C. \tag{2}$$

The attention weights play a role in describing the contribution of both individual frames and channels towards the creation of speaker embeddings. In the absence of attention, the weights are uniformly set to $\frac{1}{T}$. Finally, $\{\boldsymbol{h}_{1:T}\}$ is aggregated into the weighted statistics μ_c and σ_c for each channel c:

$$\mu_c = \sum_{t=1}^{T} \alpha_{t,c} h_{t,c}, \qquad \sigma_c = \sqrt{\sum_{t=1}^{T} \alpha_{t,c} h_{t,c}^2 - \mu_c^2} \tag{3}$$

Afterwards, the concatenated vector $[\,\boldsymbol{\mu}\,;\,\boldsymbol{\sigma}\,] \in \mathbb{R}^{2C}$ is passed through additional layers to yield the speaker embeddings.

2.2. Residual block (ResBlock)

The architecture of ResNet18 and ResNet34 is formed by ResBlocks, each of which is formulated as follows:

$$y = \mathrm{ReLU}(F(x) + I(x)) \tag{4}$$

where x and y are the input and the output of a ResBlock, respectively. The function F is a composite of a 3×3 convolution with stride (s, s), batch normalization (BN) [11], ReLU [12], another 3×3 convolution with stride $(1, 1)$, and BN. The skip connection I depends on the dimension of $F(x)$; I is the identity if the dimension of x matches with that of $F(x)$, or otherwise, I is a 1×1 convolution followed by BN, which adjusts the dimension of x to align with that of $F(x)$. To compare with our proposed block, notice that the size of feature maps changes from (F, T, C_1) into $(\frac{F}{s}, \frac{T}{s}, C_2)$ by a ResBlock.

3. TEMPORAL-BOTTLENECK RESBLOCK

ResBlocks diminish spatial dimensions, particularly the temporal dimension in the context of ASV models, through the utilization of stride operations (*e.g.*, letting $s = 2$). Consequently, ResNet-based models often result in potential loss of temporal information, raising concerns regarding their compatibility with ASP methods that aggregate temporal features.

To ensure the appropriate utilization of statistics pooling and to maximize its effectiveness, we claim that it would be efficient to capture and retain temporal information by not reducing the corresponding dimension through the series of ResBlocks. With this motivation, we propose an innovative variant, referred to as *Temporal-Bottleneck Residual Block* (TB-ResBlock), which integrates a transposed convolution that enriches the temporal information. It is formulated as:

$$y = \mathrm{ReLU}(G_2(G_1(x)) + I(x)) \tag{5}$$

Here, the function G_1 employs a 3×3 convolution with stride $(s, 2)$ followed by BN and ReLU. Conversely, G_2 entails a 3×3 transposed convolution with stride $(1, 2)$ followed by BN. A transposed convolution reverses the standard convolution by dimensions, making it a prevalent choice in image upsampling models. The function I remains consistent with its definition in Section 2.2, relying on the dimension of $G_2(G_1(x))$.

An essential observation in the TB-ResBlock pertains to the change of feature map dimensions. The function G_1 alters the size from (F, T, C_1) to $(\frac{F}{s}, \frac{T}{2}, C_2)$, while G_2 subsequently restores the temporal dimension through a transposed convolution, resulting in a tensor size of $(\frac{F}{s}, T, C_2)$. The central objective of this block is to prevent the reduction of the temporal dimension through the series of TB-ResBlocks, thereby facilitating effective aggregation in ASP.

Furthermore, TB-ResBlock brings about another advantageous outcome akin to the effect observed in bottleneck blocks in deeper ResNet architectures [7]. These conventional bottleneck blocks not only diminish the channel dimension for parameter regulation but also operate as feature extractors by compelling the network to compress feature maps, enhancing the capture of essential features. Likewise, our TB-ResBlock reduces and subsequently recovers the number of temporal frames, thereby enabling the exploration of more valuable temporal information. It is noteworthy that our design does not lead to a reduction in parameter count as it does in conventional bottleneck designs, because the spatial dimension of the feature map does not influence the parameter count within convolutional layers.

4. MODEL DESCRIPTION

This section provides a comprehensive account of our novel ResNet variant which employs TB-ResBlocks. Prior to our model, two baseline models are described, namely ResNet with global average pooling and ResNet with ASP. An overview of all network architectures is presented in Table 1.

4.1. ResNet

The architecture of the original ResNet is delineated in Table 1 by replacing Box (A / B / C) with Box A. This ResNet model, incorporating global average pooling (GAP), adheres closely to the model proposed by [7], with minor deviations in the kernel size of the conv1 layer and the final linear layer. Block(c, s, n) in the table represents a group of n consecutive ResBlocks, each structured according to Eq. 4. The initial convolution of the first ResBlock within the group applies a stride of (s, s), while all subsequent convolutions adopt a stride of $(1, 1)$, with a consistent deployment of c filters. Instead of GAP, the second ResNet model embraces the ASP layer, defined in Section 2.1, as described in Table 1 through Box B. Due to the 3D tensor outputs from ResBlocks, the tensor is flattened prior to the application of statistics pooling. ResNet18 and ResNet34 models are described through the tuple (n_2, n_3, n_4, n_5), each of which corresponds to $(2, 2, 2, 2)$ and $(3, 4, 6, 3)$ respectively.

4.2. Temporal-Bottleneck ResNet

The architecture of our proposed TB-ResNet is presented in Table 1, denoted by Box C. The pivotal attribute of this model lies in its preservation of the number of temporal frames throughout the series of TB-ResBlocks. By enhancing temporal information, the model enables the aggregation of well-preserved temporal features across the time axis within the ASP layer. Preceding the ASP layer, an additional depthwise 5×1 convolutional layer dw_conv6 exists to squeeze the frequency dimension. TB-Block(c, s, n) in Table 1 employs n TB-ResBlocks, each of which is defined by Eq. 5. The first TB-ResBlock adopts a convolution with stride $(s, 2)$, followed by a transposed convolution with stride $(1, 2)$. Subsequent n–1 TB-ResBlocks employ $(1, 2)$-strided convolution and $(1, 2)$-strided transposed convolution.

5. EXPERIMENTAL SETUP

5.1. Input feature for training

The input speech is obtained by randomly cutting a 2-second long segment from each utterance. Subsequently it is converted into 80-dimensional log-mel spectrograms using a sampling rate of 16kHz, FFT size of 512, window width of 25ms, and window shift of 10ms. For data augmentation, noise or reverberations are randomly applied to original speeches using MUSAN and Room Impulse Response and Noise database [13, 14]. Additionally, SpecAugment [15] is applied to the extracted spectrograms.

Table 1. Three ResNet architectures

Layer Name	Layer Details	Output Size
input	-	$(80, T, 1)$
conv1	5×5, BN, ReLU	$(80, T, 64)$
maxpool	3×3 window, stride 2	$(40, T/2, 64)$
conv2_x	Block(64, 1, n_2)	$(40, T/2, 64)$
Box (A / B / C)		
linear	speaker embedding	192
Box A: ResNet with GAP		
conv3_x	Block(128, 2, n_3)	$(20, T/4, 128)$
conv4_x	Block(256, 2, n_4)	$(10, T/8, 256)$
conv5_x	Block(512, 2, n_5)	$(5, T/16, 512)$
GAP	-	$(1, 1, 512)$
Box B: ResNet with ASP		
conv3_x	Block(128, 2, n_3)	$(20, T/4, 128)$
conv4_x	Block(256, 2, n_4)	$(10, T/8, 256)$
conv5_x	Block(512, 2, n_5)	$(5, T/16, 512)$
flatten	except for time axis	$(T/16, 5\times512)$
ASP	channel-dependent	5120
Box C: TB-ResNet		
conv3_x	TB-Block(128, 2, n_3)	$(20, T/2, 128)$
conv4_x	TB-Block(256, 2, n_4)	$(10, T/2, 256)$
conv5_x	TB-Block(512, 2, n_5)	$(5, T/2, 512)$
dw_conv6	5×1, BN, ReLU	$(1, T/2, 512)$
ASP	channel-dependent	1024

5.2. Training details

The VoxCeleb2 development set [16], containing 5,994 speakers and 1,092,009 utterances, serves as the training dataset for our research. Training is carried out with a batch size of 128 over 80 epochs, utilizing Adam [17] optimizer with weight decay of 2×10^{-5}. The learning rate is decayed exponentially from 10^{-3} at a rate of 0.97 per epoch. AAM-softmax [18] is employed as the loss function with a margin of 0.2 and a scale of 30, facilitating an enlarged margin between distinct speaker classes. All convolution layers in the models are initialized by Kaiming initialization [19].

5.3. Evaluation details

Model performance is evaluated on the VoxCeleb1 test set [19], comprising 1,251 speakers and 153,516 utterances. The test set contains three trial pair lists: VoxCeleb1-O,

Table 2. Model performance

Model	# Params	VoxCeleb1-O		VoxCeleb1-H		VoxCeleb1-E	
		EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
ResNet18-GAP	11.27M	2.03	0.1410	3.73	0.2300	2.08	0.1410
ResNet18-ASP	13.80M	1.62	0.1109	3.02	0.1842	1.64	0.1100
TB-ResNet18	11.44M	**1.36**	**0.0870**	**2.47**	**0.1497**	**1.39**	**0.0895**
ResNet34-GAP	21.38M	1.60	0.1080	3.20	0.1940	1.73	0.1190
ResNet34-ASP	23.91M	1.35	0.0847	2.69	0.1629	1.43	0.0966
TB-ResNet34	21.55M	**1.13**	**0.0687**	**2.20**	**0.1298**	**1.21**	**0.0779**

VoxCeleb1-H, and VoxCeleb1-E. They encompass 37,611, 550,894, and 579,818 trial pairs, representing 40, 1,190, and 1,251 speakers, respectively.

For each signal, two types of inputs are utilized: the full-length signal and five randomly sampled 3-second long segments. For a given trial pair, cosine similarity s_1 is computed between two 192-dimensional speaker embeddings derived from the full-length signals. Additionally, twenty-five cosine similarities between speaker embeddings of the 3-second segments are computed, with the average of these similarities denoted as s_2. The final score for the pair is $0.5(s_1 + s_2)$. Model performance is evaluated by the equal error rate (EER) and the minimum detection cost function (minDCF or C_{Det}) [20]. The minDCF of a model θ is defined as follows:

$$\begin{aligned} C_{Det}(\theta) &= C_{Miss} \times P_{Target} \times P_{Miss}(\theta) \\ &+ C_{FalseAlarm} \times (1 - P_{Target}) \times P_{FalseAlarm}(\theta). \end{aligned} \tag{6}$$

where $C_{Miss} = C_{FalseAlarm} = 1$ and $P_{Target} = 0.05$.

5.4. Experimental results

All experiments are conducted three times and the average results are presented in Table 2. ResNet18-GAP and ResNet34-GAP denote the models employing GAP, while ResNet18-ASP and ResNet34-ASP correspond to ASP. The substantial enhancement in model performance shows the necessity of ASP in ResNet-based models. Moreover, TB-ResNet18 and TB-ResNet34 exhibit even greater improvement, confirming the importance of enriched frame-level features via our novel temporal bottleneck figure in order to take advantage of ASP.

5.5. Ablation studies

The first investigation is about controlling the number of retained temporal frames prior to ASP. At first, ResBlocks are used to reduce the number of temporal frames, and subsequently, TB-ResBlocks are used to maintain the time dimension unchanged. For example, if the model intended to retain $T/8$ temporal frames, it reduces the time dimension up to the block conv4_x using ResBlocks, and then it keeps the time dimension unchanged by using a TB-ResBlock in conv5_x.

Table 3. Two ablation studies

Model	# Frames	VoxCeleb1-O	
		EER (%)	minDCF
TB-ResNet18	$T/16$	1.68	0.1100
	$T/8$	1.57	0.1050
	$T/4$	1.45	0.0920
	$T/2$	**1.36**	**0.0870**
Bilinear	$T/2$	1.89	0.1155
TB-ResNet34	$T/16$	1.35	0.0850
	$T/8$	1.23	0.0820
	$T/4$	1.27	0.0780
	$T/2$	**1.13**	**0.0687**
Bilinear	$T/2$	1.45	0.0885

Table 3 shows that models tend to achieve better performance when retaining more temporal frames prior to ASP.

The second investigation aims to validate the significance of transposed convolution in TB-ResBlocks. As an alternative to transposed convolution, bilinear interpolation was employed to restore the temporal dimension. Notably, Table 3 illustrates a marked degradation in performance upon the application of bilinear interpolation.

6. CONCLUSION

In this study, we proposed Temporal-Bottleneck ResNet, a novel speaker verification system incorporating a unique temporal bottleneck configuration within its building blocks. It is implemented using transposed convolutions to effectively capture and retain frame-level contexts. The design enhances the meaningful aggregation of temporal information through the ASP layer, ultimately resulting in more informative statistics. Notably, our proposed TB-ResNet showed a substantially increased performance compared to ResNet models employing GAP and ASP, as demonstrated on the VoxCeleb1 test set. These findings underscore the efficacy of our innovative time-enriched design. Importantly, the integration of this novel framework into various speaker verification models based on 2D convolutions is both seamless and practicable.

7. REFERENCES

[1] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

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

[3] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[4] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256.

[5] Miquel India, Pooyan Safari, and Javier Hernando, "Self Multi-Head Attention for Speaker Recognition," in *Proc. Interspeech 2019*, 2019, pp. 4305–4309.

[6] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, jun 2016, pp. 770–778, IEEE Computer Society.

[8] Jaesung Huh, Andrew Brown, Jee weon Jung, Joon Son Chung, Arsha Nagrani, Daniel Garcia-Romero, and Andrew Zisserman, "Voxsrc 2022: The fourth voxceleb speaker recognition challenge," 2023, arXiv:2302.10248.

[9] Qutang Cai, Guoqiang Hong, Zhijian Ye, Ximin Li, and Haizhou Li, "The kriston ai system for the voxceleb speaker recognition challenge 2022," 2022, arXiv:2209.11433.

[10] Zhengyang Chen, Bing Han, Xu Xiang, Houjun Huang, Bei Liu, and Yanmin Qian, "Sjtu-aispeech system for voxceleb speaker recognition challenge 2022," 2022, arXiv:2209.09076.

[11] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, Francis Bach and David Blei, Eds., Lille, France, 07–09 Jul 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456, PMLR.

[12] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Madison, WI, USA, 2010, ICML'10, p. 807–814, Omnipress.

[13] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[14] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[15] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le, "Specaugment: A simple augmentation method for automatic speech recognition," in *Proc. Interspeech 2019*, Graz, 2019, pp. 2613–2617, ISCA.

[16] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[17] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.

[18] 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.

[19] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[20] Seyed Omid Sadjadi, Craig Greenberg, Elliot Singer, Douglas Reynolds, Lisa Mason, and Jaime Hernandez-Cordero, "The 2018 NIST Speaker Recognition Evaluation," in *Proc. Interspeech 2019*, 2019, pp. 1483–1487.