Speaker Recognition in the Wild

Gabriele Cerizza
Department of Computer Science
University of Milan
Milan, Italy
gabriele.cerizza@studenti.unimi.it

***Abstract*—Speaker recognition is the task of recognizing a person from their utterances, either to classify or to verify the identity of a given speaker. In this work, we review the main techniques developed to tackle speaker recognition and compare the performance of different neural network models on this task, using a dataset collected under real-world conditions.**

***Index Terms*—speaker recognition, speaker identification, speaker verification, convolutional neural networks, attention**

I. Introduction

Speaker recognition is a challenging task that includes two problems: speaker identification and speaker verification [1]–[3].

Speaker identification is a closed-set problem akin to classification, in which the model has to identify the person to whom a given utterance belongs. The model is trained on utterances from each speaker. Speaker verification aims at validating whether a given utterance belongs to a given person. In academic research, this is treated as an open-set problem in which the model has to determine if two utterances belong to the same person. During evaluation, the model is presented with pairs of utterances belonging to speakers not included in the training set.

Speaker recognition applications encompass security control, online banking, retrieval, forensic tools and human computer interaction systems. A speaker recognition model can be also employed as part of a speaker diarization pipeline [4].

In our research we compared the performance of different neural network models, which allowed us to tackle both speaker identification and verification at the same time. To train and evaluate the models, we used data collected "in the wild", i.e. under real-world conditions.

The remainder of the report is organized as follows. In Section II we discuss related works in the field of speaker recognition. In Section III we illustrate the characteristics of the dataset and the features. In Section IV we describe three neural networks, whose performances are compared and discussed in Section V. Finally, Section VI contains our concluding remarks.

II. Related works

A. Model Architectures

Speaker recognition methods have been widely studied in literature and, over the years, a particular approach became predominant. The idea is to produce fixed-sized feature representations of speech segments, which can then be processed by a classifier or directly compared. With the advent of neural networks, these speaker representations, known as "embeddings", were obtained as the output of a bottleneck layer trained for classification. Embeddings proved to be robust with respect to acoustic conditions and discriminative with respect to speakers, thus yielding accurate results.

To perform speaker recognition with neural networks, an architecture composed of three blocks is traditionally employed [5]. The first block, also called "trunk", receives as input acoustic features such as MFCCs or Mel spectrograms and produces frame-level features. Typical layers of this block include CNNs, Time-Delay Neural Networks (TDNN) [6], mostly implemented as dilated convolution, and Recurrent Neural Networks such as Long Short-Term Memory (LSTM) networks.

The second block is required to aggregate the frame-level features, which have varying length depending on the length of the speech segment, into fixed dimensional utterance-level features. This block usually consists of a fully connected layer stacked on top of a pooling layer. Many different pooling layers have been developed: from simply taking the average over the temporal dimension to sophisticated attention mechanisms [7]. The output of this block is the aforementioned embedding.

Finally, the third block contains the loss function, possibly preceded by a fully connected layer projecting the embedding into a dimension whose size is equal to the number of speakers.

B. Speaker Verification Peculiarities

a) Scoring: After training the models for classification, a further step is necessary to compare utterances. During evaluation, embeddings are extracted for each utterance and then compared pairwise. These comparisons may be carried out with backend systems like Probabilistic Linear Discriminant Analysis (PLDA), possibly trained on a different dataset [8]. However, more recently, embeddings have been compared simply by cosine similarity, achieving gains in performance [7].

b) Score normalization: To produce well calibrated and reliable scores, normalization is often applied. A popular score normalization technique is "Adaptive S-Norm" (AS-Norm) [9].

Let $s(\cdot, \cdot)$ be the score between two embeddings, $\mathcal{E}_e^{\text{top}}$ be the cohort of the N closest embeddings to the enrollment utterance

e and $\mathcal{E}_t^{\text{top}}$ be the cohort of the N closest embeddings to the test utterance t. Furthermore, let

$$S_e(\mathcal{E}_e^{\text{top}}) = \{(s,\varepsilon)\}_{\forall \varepsilon \in \mathcal{E}_e^{\text{top}}}, S_e(\mathcal{E}_t^{\text{top}}) = \{(s,\varepsilon)\}_{\forall \varepsilon \in \mathcal{E}_t^{\text{top}}} . \quad (1)$$

The normalized score is computed as

$$s(e,t) = \frac{1}{2} \cdot \left(\frac{s(e,t) - \mu(S_e(\mathcal{E}_e^{\text{top}}))}{\sigma(S_e(\mathcal{E}_e^{\text{top}}))} + \frac{s(e,t) - \mu(S_e(\mathcal{E}_t^{\text{top}}))}{\sigma(S_e(\mathcal{E}_t^{\text{top}}))} \right) . \quad (2)$$

c) Metrics: Once the similarity score between each pair of embeddings has been computed, we need to evaluate how good the model is in deciding whether the speech segments belong to the same speaker. To this end, various metrics have been proposed. Two notable metrics are Equal Error Rate (EER) and Minimum Detection Cost Function (MinDCF) [10], [11].

EER is defined as the point on the Receiver Operating Characteristic (ROC) curve in which $P_{\text{miss}} = P_{\text{fa}}$, where P_{miss} is the ratio of samples belonging to the same speaker classified as dissimilar and P_{fa} is the ratio of dissimilar samples classified as belonging to the same speaker. EER can be considered as a summary of the ROC curve.

On the other hand, MinDCF is derived from the normalization of the following weighted sum of misses and false-alarm error probabilities for a given decision threshold θ:

$$C_{\text{miss}} \times P_{\text{target}} \times P_{\text{miss}}(\theta) + C_{\text{fa}} \times (1 - P_{\text{target}}) \times P_{\text{fa}}(\theta), \quad (3)$$

where C_{miss} and C_{fa} are respectively the cost of misses and false-alarms and P_{target} is the *a priori* probability of the specific speaker. The costs are usually fixed to 1. We refer to [11] for the mathematical details of the normalization operation.

C. Pooling Layers

a) Temporal Average Pooling (TAP): This is the simplest pooling, which takes the mean of the features along the time domain [7], [12]. In this way, all frames equally contribute to the utterance representation.

b) Self-Attentive Pooling (SAP): Under the assumption that not all frames are equally informative, SAP uses an attention mechanism to learn which weight to assign to each frame. The frames are then multiplied by their respective weights and summed to obtain the utterance-level representation [7]. A SAP layer can act as a Voice Activity Detection (VAD) layer [13].

More formally, let W, b and μ be learnable parameters. Additionally, let $\{x_1, \ldots, x_T\}$ be the time domain features of a given utterance. We first compute the attention weights w_t as

$$h_t = \tanh(W x_t + b) , \quad (4)$$

$$w_t = \frac{\exp(h_t^T \mu)}{\sum_{t=1}^{T} \exp(h_t^T \mu)} . \quad (5)$$

Finally, we take the sum

$$e = \sum_{t=1}^{T} w_t x_t . \quad (6)$$

c) Other: Other notable pooling layers have been proposed. One of them is Self Multi-Head Attention Pooling, which splits the input to the layer into N sequences, applies SAP to each of them and then concatenates the results [14]. Another one is Attentive Statistics Pooling (ASP), which combines SAP with mean and standard deviation statistics [5].

D. Softmax for Speaker Recognition

Neural networks for speaker recognition may be effectively trained with Softmax and Cross-entropy loss, which are traditionally employed for classification problems.

The *caveat* is that Softmax penalizes only classification errors, without enforcing intra-class compactness and inter-class separation [3]. This makes Softmax unsuited to learn discriminative features, which map utterances belonging to the same speaker close to each other and far from utterances belonging to other speakers [15]. To remedy this, angular based losses have been developed.

This class of loss functions is chiefly represented by Angular Additive Margin Softmax (AAM Softmax or ArcFace) [16]. This function introduces an angular margin penalty m, which forces the cosine similarity between the sample and its true class to be m more than the cosine similarity between the sample and wrong classes. This difference is also multiplied by a scale factor s, which prevents the gradient from getting too small [2], [3]. On the downside, it is difficult to train with AAM Softmax and results are sensitive to the parameter values.

Recently, Sub-center AAM Softmax (SC AAM Softmax) has been proposed [17]. This loss function relaxes the intra-class compactness constraint by incorporating K sub-centers for each class and forcing each sample to be close to any of the positive sub-centers. It is expected that one dominant sub-center will contain the majority of "clean" samples belonging to the class, while the hard and noisy samples will gravitate toward the non-dominant sub-centers. As such, SC AAM Softmax is more robust to noise.

III. Data and Features

A. Dataset

For our study, we elected to use the VoxCeleb1 dataset, which is characterized by speech segments collected "in the wild" from YouTube [4], [12], [18]. This dataset is particularly challenging due to the fact that the samples present both extrinsic and intrinsic variations. Extrinsic variations include background noise, chatter, music, laughter, cross-talk and varying room acoustics. Intrinsic variations concern the heterogeneity of age, gender and nationality of the speakers.

The VoxCeleb1 dataset consists of over 150,000 utterances from 1,251 celebrities. However, due to hardware constraints, we were forced to confine our experiments to a randomly

sampled subset. We extracted all the utterances of 100 randomly chosen speakers, while retaining the gender proportions to guarantee the representativeness of the sample. We kept the official splits for the training, validation and test sets that were provided for identification. To evaluate the verification capabilities of the models, we used utterances from 10 further speakers not present in the subset previously described. Some statistics on both the dataset and the subsets are provided in Table I.

TABLE I
VOXCELEB1 DATASET STATISTICS

		Full Dataset	Identif. Set	Verif. Set
Speakers No.		1,251	100	10
Samples No.		153,516	13,042	758
Gender	*Male*	0.55	0.55	0.50
	Female	0.45	0.45	0.50
Nationality[a]	*USA*	0.64	0.59	0.80
	UK	0.17	0.19	0.10
	Canada	0.04	0.02	-
	Australia	0.03	0.04	-
Seconds	*Mean*	8.25	8.20	8.05
	Std	5.31	5.35	5.14

[a]Only the four most frequent nationalities in the entire dataset are listed.

B. Features

As input features, we extracted 80-dimensional log Mel spectrograms with a window length of 25 ms (Hamming window) and a frame-shift of 10 ms, to which we applied cepstral mean normalization at the instance level. MFCCs are ill suited for the VoxCeleb1 dataset, since they degrade with real-world noise and they lack speaker discriminating features like pitch information [18]. VAD is ineffective, since VoxCeleb1 consists mostly of continuous speech [3].

In order to improve generalization, we also applied data augmentation. We performed four types of offline data augmentation. For each training sample we

1) perturbed the waveform by a speed factor of 0.9 or 1.1, randomly chosen;
2) added background noise, randomly chosen from the MUSAN dataset [19], with a signal-to-noise ratio DB between 0 and 15;
3) added babble effect by superimposing a speech track randomly chosen from MUSAN over the waveform, with a signal-to-noise ratio DB between 15 and 20;
4) added reverberation by using the `pedalboard` library from Spotify[1].

As a consequence, we obtained a total of 59,140 training samples. Furthermore, we chained SpecAugment [20] online data augmentation by randomly masking 0 to 5 frames in the time domain and 0 to 10 frequency bands. Validation and test samples were left untouched.

[1]https://github.com/spotify/pedalboard

C. K-Means Clustering

For a better understanding of the complexity of the task, we show in Fig. 1 the distribution of clusters identified by K-Means on the test set, after learning the location of the centers on a training set with a maximum of 50 utterances per speaker. To obtain a bidimensional representation of the data, we used Principal Component Analysis (PCA), likewise trained on a subset of the training set.

IV. MODELS AND TRAINING

A. Models

We compared three different neural network models, which we describe below. The block diagram of the systems is illustrated in Figure 2.

a) ResNet34-SE: The first model is based on the ResNet34 architecture [21], modified to accept log Mel spectrograms as input. Following the approach described in [3], [22], we halved the number of channels in the convolutional layers, removed the max pooling layer, reduced the kernel size of the first convolutional layer and adopted SAP to aggregate features in the time domain. Additionally, we introduced Squeeze-and-Excitation (SE) layers [23], which proved to be good for speaker verification according to [24]. For the loss function, we used SC AAM Softmax.

b) LAS-MHA: Taking inspiration from the Listen, Attend and Spell (LAS) [25] model for sequence-to-sequence speech recognition and from the Multi-Head Attention (MHA) [26] employed in language modeling, we devised a new architecture. We stacked a MHA layer with 8 heads on top of the encoder part of the LAS model, composed of CNN layers followed by bidirectional LSTM layers. A TAP layer was inserted between the CNN and LSTM layers. We used SC AAM Softmax during training.

c) EfficientNetV2: Many state-of-the-art architectures in speaker recognition have been borrowed from the computer vision domain. For this reason, we decided to use EfficientNetV2 [27], a recently developed model that performs well in image classification while being optimized for training speed and parameter efficiency. EfficientNetV2 is characterized by its use of depthwise convolution in place of traditional convolution, which reduces the number of parameters. We used a simple TAP layer. EfficientNetV2 was trained with standard Softmax loss.

B. Training and Scoring

Training was carried out in two steps, according to the Large Margin Fine-Tuning strategy [28]. We first trained the models on random crops of the utterances with a length of 2 seconds (eventually padded), to avoid overfitting. When the models stopped learning, we fine-tuned them with random crops of 4 seconds. We also increased the margin m of SC AAM Softmax from 0.1 to 0.15 and the scale s from 15 to 20. Increasing the margin makes training harder, but leads to more discriminative embeddings. Additionally, we dropped SpecAugment online data augmentation. The models were

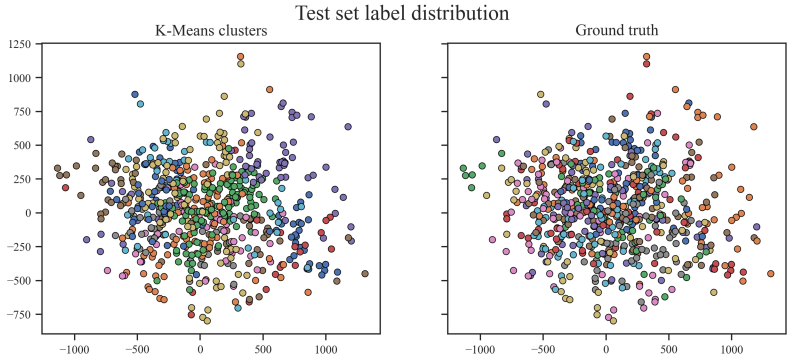

Fig. 1. Comparison between K-Means clusters and ground truth labels on the speaker identification test set.

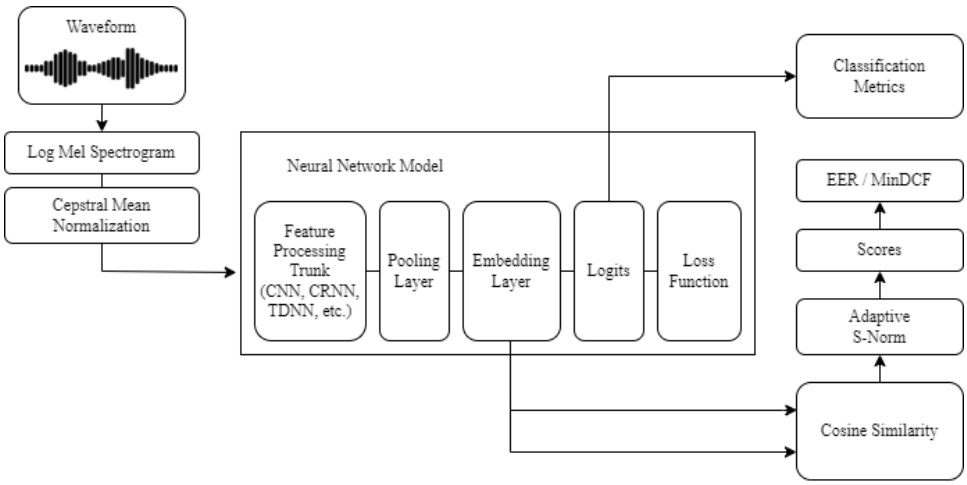

Fig. 2. Block diagram of a generic speaker recognition system.

evaluated against 4 seconds random crops for identification and full length utterances for verification.

We used 256 dimensional embeddings, which were found to be the best for identification in [2]. Adam was chosen as optimizer, with a learning rate of 0.001.

For speaker verification, we compared the embeddings by cosine similarity and then normalized the scores with AS-Norm, using cohorts of size 100.

V. Results

Considering the fact that we used a subset of VoxCeleb1 for both training and test set, a direct comparison with models described in literature is not possible. Besides, researchers often trained their models on the much larger VoxCeleb2 dataset before evaluating them on the VoxCeleb1 test set. For this reason, we added the kind of training set to the results shown in Table II. As a baseline, we used a Support Vector Machine (SVM) with RBF kernel and parameter C equal to 1, trained on the same subset as K-Means (see Section III).

Our models performed better than the baseline, but could not match the current state-of-the-art systems. We argue that literature models (i) were trained on larger datasets, which provide more diversity and variation and thus better generalization, as the models trained on VoxCeleb2 suggest; (ii) were usually trained on 6 seconds long utterances, which was not possible for us; (iii) were aggregated in fusion or ensemble systems with up to 14 models [24].

ResNet34-SE and EfficientNetV2 achieved similar performances on the identification task, but the former was clearly superior on the verification task. This can be ascribed to the more discriminative embeddings obtained with SC AAM Softmax.

VI. Conclusions

This work investigated the field of speaker recognition, illustrating the most popular techniques and comparing different neural network models. Our models, albeit far from the state-of-the-art, showed promising results, which could be improved in future research by way of larger datasets and more computational power.

References

[1] M. M. Kabir, M. F. Mridha, J. Shin, I. Jahan, and A. Q. Ohi, "A survey of speaker recognition: Fundamental theories, recognition methods and opportunities," *IEEE Access*, vol. 9, pp. 79 236–79 263, 2021.

TABLE II
VOXCELEB1 TEST SET RESULTS

Model	Year	Training Set	Top1 Accuracy	Top5 Accuracy	F1 Score	EER(%)	MinDCF[a]
Nagrani et al. [29]	2017	VoxCeleb1	80.50	92.10	-	-	-
Nagrani et al. [29]	2017	VoxCeleb1	-	-	-	7.80	0.710 (0.01)
Cai et al. [7]	2018	VoxCeleb1	89.90	95.70	-	-	-
Cai et al. [7]	2018	VoxCeleb1	-	-	-	5.27	0.439
Cai et al. [7]	2018	VoxCeleb1	-	-	-	4.46	0.577
Okabe et al. [5]	2018	VoxCeleb1	-	-	-	3.85	0.406 (0.01)
Hajibabaei, Dai [2]	2018	VoxCeleb1	**94.60**	**98.10**	-	4.69	0.453 (0.01)
Hajibabaei, Dai [2]	2018	VoxCeleb1	92.80	97.50	-	4.30	0.413 (0.01)
Chung et al. [12]	2019	VoxCeleb1	89.00	96.15	-	5.37	-
Chung et al. [12]	2019	VoxCeleb1	89.00	95.94	-	5.26	-
Hajavi, Etemad [30]	2021	VoxCeleb1	-	-	-	3.14	-
Thienpondt et al. [28]	2021	VoxCeleb2	-	-	-	0.64	0.070 (0.01)
Thienpondt et al. [28]	2021	VoxCeleb2	-	-	-	0.56	0.074 (0.01)
Zhao et al. [31]	2021	VoxCeleb2	-	-	-	**0.52**	0.050 (0.01)
Zhao et al. [31]	2021	VoxCeleb2	-	-	-	0.56	**0.048 (0.01)**
SVM (our baseline)	2022	VoxCeleb1 Subset	13.98	-	11.46	-	-
ResNet34-SE (ours)	2022	VoxCeleb1 Subset	64.21	84.21	59.28	14.50	0.893 (0.01)
LAS-MHA (ours)	2022	VoxCeleb1 Subset	49.32	65.26	45.40	24.47	0.984 (0.01)
EfficientNetV2 (ours)	2022	VoxCeleb1 Subset	67.67	85.11	**64.26**	16.44	0.974 (0.01)

[a]If provided by the Authors, we noted the P_{target} value within parentheses.

[2] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[3] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *Interspeech 2020*, Oct 2020.

[4] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *Interspeech 2018*, Sep 2018.

[5] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256.

[6] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech 2015*, 2015, pp. 3214–3218.

[7] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

[8] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proc. Interspeech 2017*, 2017, pp. 999–1003.

[9] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, and J. Černocký, "Analysis of Score Normalization in Multilingual Speaker Recognition," in *Proc. Interspeech 2017*, 2017, pp. 1567–1571.

[10] N. Brümmer and E. De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[11] "Nist 2018 speaker recognition evaluation plan," https://www.nist.gov/itl/iad/mig/nist-2018-speaker-recognition-evaluation, accessed: 2022-03-07.

[12] J. S. Chung, J. Huh, and S. Mun, "Delving into voxceleb: environment invariant speaker recognition," *arXiv preprint arXiv:1910.11238*, 2019.

[13] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[14] M. India, P. Safari, and J. Hernando, "Self multi-head attention for speaker recognition," *arXiv preprint arXiv:1906.09890*, 2019.

[15] Y. Liu, L. He, and J. Liu, "Large Margin Softmax Loss for Speaker Verification," in *Proc. Interspeech 2019*, 2019, pp. 2873–2877.

[16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[17] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center arcface: Boosting face recognition by large-scale noisy web faces," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 741–757.

[18] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[19] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[20] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] H. S. Heo, B.-J. Lee, J. Huh, and J. S. Chung, "Clova baseline system for the VoxCeleb speaker recognition challenge 2020," *arXiv preprint arXiv:2009.14153*, 2020.

[23] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.

[24] L. Zhang, H. Zhao, Q. Meng, Y. Chen, M. Liu, and L. Xie, "Beijing zkj-npu speaker verification system for voxceleb speaker recognition challenge 2021," *arXiv preprint arXiv:2109.03568*, 2021.

[25] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," *arXiv preprint arXiv:1902.01955*, 2019.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 096–10 106.

[28] 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 *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5814–5818.

[29] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[30] A. Hajavi and A. Etemad, "Siamese capsule network for end-to-end speaker recognition in the wild," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7203–7207.

[31] M. Zhao, Y. Ma, M. Liu, and M. Xu, "The speakin system for voxceleb speaker recognition challange 2021," *arXiv preprint arXiv:2109.01989*, 2021.