APSIPA Transactions on Signal and Information Processing, 2023, 12, e10

Original Paper

Speaker-Specific Articulatory Feature Extraction Based on Knowledge Distillation for Speaker Recognition

Qian-Bei Hong[1], Chung-Hsien Wu[1,2*] and Hsin-Min Wang[1]

[1] *Graduate Program of Multimedia Systems and Intelligent Computing, National Cheng Kung University and Academia Sinica, Taiwan*

[2] *Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan*

ABSTRACT

This paper proposes a novel speaker-specific articulatory feature (AF) extraction model based on knowledge distillation (KD) for speaker recognition. First, an AF extractor is trained as a teacher model for extracting the AF profiles of the input speaker dataset. Next, a KD-based speaker embedding extraction method is proposed to distill the speaker-specific information from the AF profiles in the teacher model to a student model based on multi-task learning, in which the lower layers not only capture the speaker characteristics from acoustic features, but also learn the speaker-specific features from the AF profiles for robust speaker representation. Finally, speaker embeddings are extracted from the high-level layer, and the obtained speaker embeddings are further used to train a probabilistic linear discriminant analysis (PLDA) model for speaker recognition. In the experiments, speaker embedding models were trained using the VoxCeleb2 dataset and the AF extractor was trained based on the LibriSpeech dataset, and the performance was evaluated using the VoxCeleb1 dataset. The experiments showed that the proposed KD-based models outperformed the baseline models

*Corresponding author: Chung-Hsien Wu, chunghsienwu@gmail.com.

Received 29 July 2022; Revised 04 October 2022
ISSN 2048-7703; DOI 10.1561/116.00000150

without KD. Furthermore, feature concatenation of multimodal results can further improve the performance.

Keywords: Speaker recognition, articulatory feature, knowledge distillation

1 Introduction

Speaker recognition has been an important task for artificial intelligence applications [1, 3, 20] for years. To represent the short-term spectral characteristics of speakers, traditional methods for speaker recognition typically used Gaussian mixture models (GMM-UBM) and i-vector [9, 10, 17]. Villalba *et al.* [43] presented a framework based on the variational autoencoder paradigm to deal with latent variables between i-vectors. Chin *et al.* [7] combined i-vector and sparse representation classifier for speaker identification. Kinnunen *et al.* [18] adopted i-vector as the basic unit for voice conversion. In recent years, deep neural networks (DNN) have been widely used to generate features for speaker recognition [5]. In these studies, DNN was applied to directly capture speaker characteristics and produce speaker embedding as a speaker representation [2, 23, 42, 45, 46]. Recently, most speaker verification (SV) systems were based on x-vector features [37, 39], and the architecture consisted of frame-level and segment-level feature transformations. The frame-level feature transformation was based on time delay neural network (TDNN) structure [44]. It has been proven that using TDNN, speech characteristics extracted from multi-frame signals with shift-invariance were more efficient than those from single-frame signals [36]. The segment-level feature transformation applied statistics pooling to aggregate variable-length features to obtain a fixed-dimensional vector. In [15], a statistics pooling time delay neural network was proposed to improve the x-vector learning ability by capturing more robust speaker characteristics. Nowadays, many studies were focused on improving speaker recognition performance. Tang *et al.* [41] integrated TDNN and long short-term memory (LSTM) to capture speaker information at different levels. Zhu *et al.* [51] proposed a self-attention mechanism for DNN-based embedding and computed the embedding as a weighted average of the speaker's frame-level features. However, speech attributes, such as articulatory features (AFs), which can be used to characterize speaker-specific information, are rarely considered in speaker recognition.

AF is an important representation of phonological properties during speech production. More precisely, AFs are abstract classes for describing the movements or positions of different articulators during speech production [6]. AFs have been successfully used as features in speech recognition in recent years [19, 26, 27, 31, 40, 48], and commonly used methods combine acoustic features

and AFs to improve speech recognition performance. For the speaker recognition task [13, 14], Shen *et al.* [34] utilized decision tree-based phone cluster models to cluster the speech segments with speaker characteristics for speaker diarization. Li *et al.* [22] proposed a feature-level fusion and a score-level fusion approach by combining acoustic and AF information for SV, and the authors indicated that concatenating AF with acoustic features can improve the performance dramatically, but access to the AF is impractical for real world applications. Thus, using an acoustic-to-articulatory inversion technique can deal with this issue [22]. Siniscalchi *et al.* [35] showed the relationship between phone and attribute, and computed a score by articulatory feature detectors to describe the activation level of the specified speech phonetic features that the current frame exhibits. Wu *et al.* [47] integrated senones with an AF posterior probability vector to model a wide range of acoustic-phonetic phenomena in a language for code-switching event detection.

For feature fusion, most of the studies combining various vectors are based on the concatenation of different features [16, 33]. Using an AF extractor is a direct way to generate the AF features of the training speaker dataset for feature fusion. However, the AF features obtained from individual modules may cause performance drops due to representation specificity [49]. Therefore, this study proposes a speaker-specific AF extraction model via knowledge distillation (KD) to capture the speaker characteristics from acoustic features and then learn the speaker-specific information from AFs to distinguish speakers. By applying KD rather than feature concatenation, speaker embedding model learning based on acoustic and AF features simultaneously cannot only achieve a better fusion performance, but also reduce the impact of representation specificity.

In speaker recognition, the state-of-the-art (SOTA) method for speaker discrimination is based on the probabilistic linear discriminant analysis (PLDA) backend [4], by comparing the speaker embedding of the input utterance with the embedding features of the speakers. McCree *et al.* [25] extended the PLDA model to include segment duration as well as to distinguish between session and channel variability. Rohdin *et al.* [32] developed an end-to-end speaker recognition system that is initialized to mimic an i-vector with a PLDA baseline. Snyder *et al.* [38] used a diarization system based on PLDA as a front-end for speaker recognition.

In this paper, we propose a speaker-specific AF extraction technique based on KD for speaker recognition. First, an articulatory feature (AF) extractor is trained as a teacher model to extract the AF profiles, in which each AF profile can be seen as the speaker-specific information. Next, a student model as low-level layers of a speaker embedding model is trained based on multi-task learning using KD to learn more robust speaker embedding. Finally, speaker embeddings are extracted from the high-level layer, and the obtained speaker embeddings are further used to train a PLDA model for scoring.

This paper is organized as follows. An improved structure of frame-level transformation of x-vector for speaker embedding extraction is described in Section 2. The relationship of phones and AFs for AF extractor training is presented in Section 3. A KD model based on acoustic and AF features for speaker embedding extraction is presented in Section 4. Section 5 introduces the dataset and details the experimental results of different comparisons. Finally, conclusions are given in Section 6.

2 Speaker Embedding Extraction

In this section, the structure proposed in [15] is used to improve the x-vector representation, which is regarded as the SOTA feature representation for speaker recognition. As the TDNN layer focuses on local feature extraction, the high-level features extracted through non-linear transformation in the preceding layers may lose some subtle information using low-level features. Therefore, this study integrates TDNN with statistics pooling to exploit the potential ability of the network by considering the variation in temporal context.

2.1 Standard X-Vector Representation

Researchers have been working to improve the performance of speaker recognition [39] using the standard x-vector. For x-vector feature extraction, a speaker discriminative network is trained with a large amount of speech data from speakers for enrollment and evaluation. Table 1 shows the x-vector architecture. Let $X_l = [x_{l,1}\, x_{l,2} \ldots x_{l,T}]^{\mathrm{T}}$ represent an input sequential vector with T frames at the l-th layer. Suppose that $s_{l,t} = flatten(X_{l,t})$, where $X_{l,t} = [x_{l,t-\tau}\, x_{l,t}\, x_{l,t+\tau}]^{\mathrm{T}}$ is the spliced output of X_l at frames $\{t-\tau, t, t+\tau\}$ with a dilation factor τ. The frame-level transformation of the TDNN can be written as

$$x_{l+1,t} = \alpha\left(W_{l+1}^{\mathrm{T}} s_{l,t} + b_{l+1}\right) \tag{1}$$

where $W_{l+1} \in \mathbb{R}^{d_l^s \times d_{l+1}^x}$ is the weight matrix of size $d_l^s \times d_{l+1}^x$, d_l^s is the length of $s_{l,t}$, d_{l+1}^x is the length of $x_{l+1,t}$, b_{l+1} is the bias vector at the $(l+1)$-th layer and $\alpha(\cdot)$ is the activation function.

After the transformation of L frame-level layers, statistics pooling is performed by aggregating all output vectors of the last frame-level layer (i.e., L-th layer) to form a fixed-dimensional vector as

$$\begin{aligned} z &= stat\left([x_{L,1}\, x_{L,2}\, \ldots\, x_{L,T}]\right) \\ &= \begin{bmatrix} mean([x_{L,1}\, x_{L,2}\, \ldots\, x_{L,T}]) \\ std([x_{L,1}\, x_{L,2}\, \ldots\, x_{L,T}]) \end{bmatrix} \end{aligned} \tag{2}$$

Table 1: X-vector system architecture.

Layer	Layer context	Total context	Input dim.	Output dim.
Frame1	$\{t-2, t-1, t, t+1, t+2\}$	5	200	512
Frame2	$\{t-2, t, t+2\}$	9	1536	512
Frame3	$\{t-3, t, t+3\}$	15	1536	512
Frame4	$\{t-4, t, t+4\}$	23	1536	512
Frame5	$\{t\}$	23	512	512
Frame6	$\{t\}$	23	512	1500
Stats pooling	$[0, T)$	T	$1500T$	3000
Segment7	$\{0\}$	T	3000	512
Segment8	$\{0\}$	T	512	512
Softmax	$\{0\}$	T	512	N

where $mean(\bullet)$ is the mean function and $std(\bullet)$ is the standard deviation function.

By pooling the frame-level features to the segment-level features, the discriminative probabilities of speakers can be predicted by I dense layers.

$$h_i = \alpha\left(W_i^{\mathrm{T}} h_{i-1} + b_i\right), \quad i > 0 \tag{3}$$

$$\hat{Y} = softmax\left(W_I^{\mathrm{T}} h_{I-1} + b_I\right) \tag{4}$$

where $h_0 = z$ and the predicted probabilities of N speakers $\hat{Y} = \{\hat{y}_n \in \mathbb{R} : 0 \leq \hat{y}_n \leq 1 \text{ and } \sum_n \hat{y}_n = 1\}$ is determined by a softmax function. In this study, a cross-entropy loss L_{SE} for speaker recognition task is defined as follows.

$$\mathcal{L}_{SE} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{n=1}^{N} y_{i,n} log\left(\hat{y}_{i,n}\right) \tag{5}$$

where B is the batch size and $y_{i,n}$ represents the label of the n-th speaker of the i-th sample in the training process. After model training is completed, x-vector embedding can be obtained from the output of the penultimate dense layer.

2.2 Frame-Level Statistics Pooling TDNN

As the TDNN focuses on local feature extraction, segment-level feature extraction through statistics pooling and non-linear transformation may lose some subtle information using frame-level features. In order to further improve the information representation in TDNN, a feature combination method for each

time-delay layer proposed in [15] (named stats-vector) is adopted to integrate the TDNN with statistics pooling to exploit the potential ability of the network by considering the variation in temporal context. As shown in Equation (1), the TDNN output is obtained by context input transformation. To further consider the variation in the input features, we directly combine $s_{l,t}$ and the statistics pooling result of $X_{l,t}$ to form a new input feature vector, which is then fed into the transformation layer.

$$\begin{aligned}\hat{x}_{l+1,t} &= \alpha\left(W_{l+1}^{\mathrm{T}}\begin{bmatrix} s_{l,t} \\ stat(X_{l,t}^{\mathrm{T}}) \end{bmatrix} + b_{l+1}\right) \\ &= \alpha\left(W_{l+1}^{\mathrm{T}}\begin{bmatrix} x_{l,t-\tau} \\ x_{l,t} \\ x_{l,t+\tau} \\ mean([x_{l,t-\tau}\, x_{l,t}\, x_{l,t+\tau}]) \\ std([x_{l,t-\tau}\, x_{l,t}\, x_{l,t+\tau}]) \end{bmatrix} + b_{l+1}\right)\end{aligned} \tag{6}$$

During the stats-vector training, assuming that the input sequential vectors of $X_{l,t}^{\mathrm{T}}$ are equal at the l-th layer ($x_{l,t-\tau} = x_{l,t} = x_{l,t+\tau}$), the output of $mean(\bullet)$ is the same as the current time vector $x_{l,t}$, and $std(\bullet)$ produce a zero vector, thus, $\hat{x}_{l+1,t}$ is an approximation of $x_{l+1,t}$ as a result of the assumptions on stationarity. Otherwise, the variation of the input sequential vectors of $X_{l,t}^{\mathrm{T}}$ will produce different local means and standard deviations, which provide more helpful features for model training.

3 Articulatory Feature Extraction

This study builds an AF extractor to predict the probabilities of the speech attributes present in the speech signal and helps the speaker embedding extraction model to extract more representative speaker features.

3.1 Speech Attributes of Articulatory Features

AFs can be distinguished based on pronunciation places and manners by speaker voices [21]. As shown in Table 2, 20 attributes, defined in this study, are used to train an AF extractor.

According to linguistics, a word is composed of several phonemes to represent the changes in pronunciation (place or manner of articulation). Thus, the labels of speech attributes in signals can be defined by phones. Table 3 shows the relationship of phonetic symbols and speech attributes in English [50]. The phonetic symbols are defined by the CMU dictionary, which is based on the ARPAbet symbol set developed for speech recognition tasks, and could be downloaded at http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Table 2: The speech attributes used in the study.

Categories	Attributes
Manner	Approximant, Fricative, Nasal, Stop, Vocalic
Place	Anterior, Back, Continuant, Coronal, Dental, High, Labial, Low, Mid, Retroflex, Round, Tense, Velar, Voiced
Silence	Silence

3.2 Articulatory Features Extractor

In this study, a DNN for acoustic-to-articulatory inversion is constructed for AF recognition. This study constructs a multilayer perceptron (MLP)-based model and a TDNN-based model to explore the effects of multi-frame integration at different layers and evaluate the performance of AF recognition, as shown in Figure 1. As each speech phone may correspond to one or several attributes, in this study, the Kaldi automatic speech recognition (ASR) toolkit [29] is used to align the phone positions of the speech signals in the GMM-HMM-based acoustic model training procedure. According to the alignment information, every segment of the training speech signals can be labeled exactly with the attributes which the phone corresponds to. As shown in Figure 2, the AF extractor training is based on the acoustic features of a segment of $\mathcal{T}$ frames. And the acoustic features are extracted from the raw signals according to the duration of each phone labeled in the GMM-HMM-based acoustic model training procedure. Acoustic features $\mathcal{S}$ can be labeled with 20 binary values to indicate which speech attribute is present in the acoustic features.

$$\mathcal{A}_j = \begin{cases} 1, & \mathcal{S} \text{ has the } j\text{th attribute;} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Through the non-linear hidden layer transformation, the AF extractor maps acoustic features into the posterior probabilities of all speech attributes. In this study, the silence attribute SIL was ignored for AF extractor training, because silent signals will be removed in speaker recognition to reduce the interference of non-verbal signals. AF extractor training is different from traditional classifier training. As the acoustic features may contain more than one attribute and the model is trained with a distributed (not one-hot) representation, the prediction score is estimated based on a sigmoid function to scale the value to lie between 0 and 1. The loss function is the mean squared error $\mathcal{L}_{AF}$ defined as follows.

$$\mathcal{L}_{AF} = \frac{1}{BC} \sum_{i=1}^{B} \sum_{j=1}^{C} \left(\mathcal{A}_{i,j} - \hat{\mathcal{A}}_{i,j} \right)^2 \tag{8}$$

Table 3: The relationship of phones and speech attributes.

	Vowels																Consonants																				-
	AA	IY	AY	OW	UW	Y	AE	AH	AO	AW	EY	EH	ER	OY	IH	UH	B	CH	D	DH	G	HH	JH	SH	ZH	TH	V	W	Z	F	K	L	M	N	NG	P	R	S	T	SIL
Appr		•				•																						•			•				•				
High		•			•	•									•	•		•					•	•	•														
Mid				•				•	•			•																											
Low	•		•				•			•	•			•																									
Retr													•																							•			
Back	•		•	•	•			•	•	•	•			•		•					•									•								
Cont	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•					•	•	•	•	•	•	•	•				•	•		
Round				•	•	•			•	•				•		•											•	•								•			
Tense	•		•	•			•		•	•	•			•	•			•				•		•	•	•				•	•				•	•	•	
Voic	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•		•				•	•	•		•	•	•	•	•	•		
Voca	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•												•			•				•			
Fric																		•		•		•	•	•	•	•	•		•	•					•	•		
Nasal																																•	•	•				
Stop																	•		•		•										•				•			•
Coro																			•										•		•		•			•	•	
Labi																	•										•	•		•		•			•			
Dent																				•						•					•		•				•	
Velar																					•									•				•				
Ante																	•		•	•						•	•	•	•	•		•	•	•		•	•	
Sil																																						•

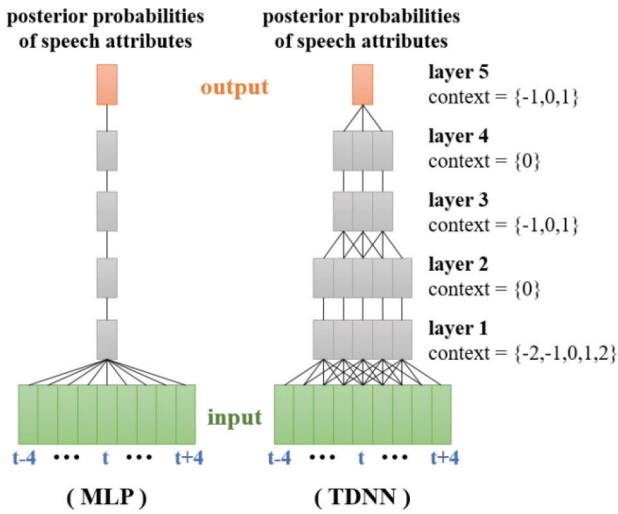

Figure 1: The MLP-based and TDNN-based models were trained for AF recognition. (The output layer of example covers a total temporal context of 9 frames).

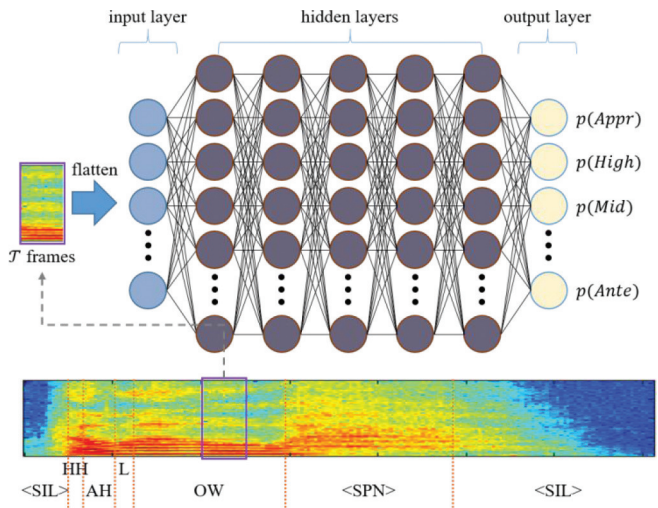

Figure 2: MLP-based AF extractor is used to predict the posterior probabilities of speech attributes. (The input example is "hello" that is constructed by phone sequence of [HH, AH, L, OW]).

where B is the batch size, C is the number of attributes, $\mathcal{A}_{i,j}$ represents the target label of the j-th attribute of the i-th sample, and $\hat{\mathcal{A}}_{i,j}$ is the actual DNN output for the j-th attribute given the i-th sample.

4 Knowledge Distillation for Speaker Recognition

As described in [42], speaker representation is derived by averaging all corresponding d-vectors for the utterances of a specific speaker. Even if speaker embedding is extracted from the same speaker, the speaker embedding is slightly different from each other. Therefore, making the model learn more robust representation to precisely recognize speakers is still a challenge.

As speech pronunciation can reflect the speaker's speaking styles and habits, in which not only different speakers have different pronunciations, but also the same speaker has different presentations. In this study, considering that speech signals are constructed by several different phones, a knowledge distillation model is proposed to characterize speaker-specific information from AF profiles. The speaker embedding model not only captures the speaker characteristics from acoustic features, but also learns the speaker-specific features from the AF profiles to form a more robust speaker representation.

4.1 Multi-Task Learning for Shared Layers

Inspired by [24], speaker embedding extracted with phonetic information by multi-task learning can effectively improve the performance of speaker recognition. In [24], the frame-level shared layers of the x-vector are shared with the ASR network, in which the shared layers learn more informative features by classifying the phonetic features in frame-level transformation. Assuming there are four parameters $\{\theta_s, \theta_a, \theta_f, \theta_l\}$ in the model, where θ_s denotes the frame-level shared layers, θ_a denotes the remaining parameters of the ASR network at frame level, θ_f denotes the remaining parameters of the x-vector at frame level, and θ_l denotes the segment-level parameters of the x-vector. These four parameters will be updated simultaneously at the training stage by gradient backpropagation.

In [24], a training strategy is proposed to deal with the following situation. That is, when the training speaker dataset does not have phonetic labels, it is desirable to use another dataset containing phonetic labels to train the ASR network at different mini-batches. For example, given two specific domain datasets, the speaker dataset only contains speaker labels, while the phonetic dataset only contains phonetic labels. The two datasets are thus merged into different mini-batches. In the training process, when the speaker samples are fed into the model, $\{\theta_s, \theta_f, \theta_l\}$ are updated, while when the phonetic samples are fed into the model, $\{\theta_s, \theta_a\}$ are updated.

4.2 Knowledge Distillation for AF Profiles

As mentioned in Section 4.1, multi-task learning does not train speaker classifier and phonetic label classifier from the same samples as the training speaker

dataset does not have phonetic labels, which will reduce the dependency of the model on these two tasks. Therefore, this study proposes a KD-based speaker embedding extraction model based on multi-task learning to eliminate the problem. In addition, as phonetic information only corresponds to a specific speech sound (one-hot representation) and the AF profiles can be seen as the activation levels of places and manners, we replace the phonetic information with AF profiles as the features for robust speaker representation. Figure 3 shows the proposed KD architecture for speaker embedding extraction, which consists of four procedures based on multi-task learning as described in the followings.

1. An AF extractor is trained as a teacher model for extracting the AF profiles of the input speaker dataset.

2. The predicted AF profiles are used as the soft targets for shared layers learning. The shared layers are transferred to the student model to capture the speaker-specific AF by minimizing a loss function in which the target is the distribution of attribute probabilities predicted by the teacher model.

3. At the same time, the shared layers are used for speaker embedding model training based on multi-task learning.

4. After model training is completed, the speaker embedding is obtained from the output of the penultimate dense layer.

In AF extraction, the AF profiles obtained from different samples are utilized to learn the speaker-specific information. First, according to the context of F frames at the frame-level shared layer S, each sample (T frames) is fed into the model and is divided into $T - F + 1$ frame-level data for shared layers training. Second, an AF extractor is used to obtain the AF profiles of these frame-level data as soft targets, which can represent the different activation levels of speech attributes of speakers. Finally, the output of the shared layer S is fed to a dense layer with a sigmoid function to obtain the AF probabilities. The loss function is defined as follows.

$$\tilde{\mathcal{L}}_{AF} = \frac{1}{(T-F+1)BC} \sum_{i=1}^{(T-F+1)B} \sum_{j=1}^{C} \left(p_i(attr_j) - \tilde{p}_i(attr_j)\right)^2 \tag{9}$$

where $p_i(attr_j)$ represents a soft target for the j-th attribute of the i-th data obtained from the AF extractor, and $\tilde{p}_i(attr_j)$ represents the corresponding AF probability determined from the shared layers.

In speaker embedding extraction, the training loss function is the same as in Equation (5). The loss is back-propagated to update the parameters.

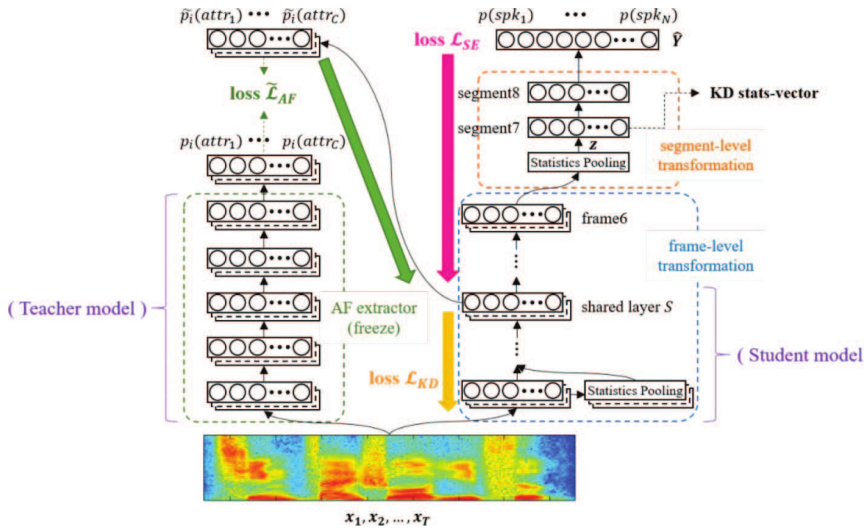

Figure 3: The proposed knowledge distillation architecture for speaker feature extraction. A speaker representation is extracted by considering acoustic features and AF profiles, the network of shared layers is not only to capture the speaker characteristics from acoustic features, but also to learn the speaker-specific information from AF profiles.

Assuming the proposed model is composed of four parameters $\{\theta_s, \theta_{af}, \theta_f, \theta_l\}$, where θ_s denotes the frame-level shared layers, θ_{af} denotes the parameters of the last dense layer of the AF predictor, θ_f denotes the remaining frame-level parameters and θ_l denotes the segment-level parameters. In the training process, θ_f and θ_l are updated based on $\mathcal{L}_{SE}$ and θ_{af} is updated based on $\tilde{\mathcal{L}}_{AF}$. As the shared layers are used for AF extraction and speaker embedding extraction simultaneously, θ_s is updated by the total loss $\mathcal{L}_{KD}$ defined as follows.

$$\mathcal{L}_{KD} = \mathcal{L}_{SE} + \tilde{\mathcal{L}}_{AF} \tag{10}$$

During the shared layer training, the network not only captures the speaker characteristics from acoustic features, but also learns the speaker-specific information of the AFs by KD.

4.3 PLDA Scoring

Speaker recognition model is generally trained by a large number of speakers to learn the rich phonetic characteristics of speakers, and then the trained model is used to further extract speaker embedding. The speaker embedding is extracted from the output of the high-level layer of the trained model for speaker discrimination.

PLDA is utilized to compute the likelihood ratios during the test. In standard PLDA [30], speaker embedding $\phi_{n,r}$ representing sample r from speaker n is given by

$$\phi_{n,r} = \mu + Vy_n + Ux_{n,r} + \epsilon_{n,r} \tag{11}$$

where μ is the global mean, Vy_n represents the between-speaker variation, $Ux_{n,r}$ is the within-speaker variation and $\epsilon_{n,r}$ represents the residual noise. The PLDA parameters $\Theta = \{\mu, V, U, \Sigma\}$ are updated by the expectation-maximization (EM) algorithm. After that, assuming there are two speaker embeddings ϕ_1 and ϕ_2 for similarity comparison, the score between the two speaker embeddings is calculated as

$$score = log \frac{p\left(\phi_1, \phi_2 \mid H_s\right)}{p\left(\phi_1 \mid H_d\right) p\left(\phi_2 \mid H_d\right)} \tag{12}$$

where H_s is the hypothesis for the same speaker and H_d is the hypothesis for different speakers. The details of the PLDA scoring could be found in [11].

5 Experimental Results

5.1 Datasets

In this study, two datasets were used for model training and testing, including VoxCeleb dataset and LibriSpeech dataset. The VoxCeleb dataset was used to evaluate the performance of the proposed mechanism on the speaker recognition task. The VoxCeleb dataset consisted of two versions, VoxCeleb1 [28] and VoxCeleb2 [8], which were released for commercial and research purposes for speaker recognition. The LibriSpeech dataset was used to evaluate the performance of the AF extractor.

- **Training data of the embedding extraction model:** The baseline system and the proposed system were trained on VoxCeleb2 dataset without any data augmentation. The VoxCeleb2 dataset provided two subsets for evaluation: DEV and TEST sets, which contained over 1 million utterances recorded from 6,112 speakers, extracted from YouTube video-sharing platform. The DEV set contained 1,092,009 utterances from 5,994 speakers and had no overlap with the speakers in the VoxCeleb1 dataset. In the experiments, the DEV set was used to train the speaker embedding models and the PLDA models.
- **Training data of AF extractor:** LibriSpeech is a corpus of approximately 1,000 h read English speech, and this corpus was released for automatic speech recognition (ASR) task. The LibriSpeech consists of

460 h "clean" speech and 500 h "other" speech. The clean speech (from 1,172 speakers) was used to train the GMM-HMM-based acoustic models using the Kaldi ASR toolkit, and the phone alignment information was obtained during the training process. After that, an AF extractor was trained with the phone alignment information corresponding to the 20 speech attributes.

- **Testing data:** The VoxCeleb1 dataset contained 153,516 utterances from 1,251 speakers, which was also obtained from YouTube videos. In this study, VoxCeleb1 dataset was used to evaluate the speaker recognition performance.

5.2 Experimental Setup

In the experiments, the audio files in the VoxCeleb dataset were labeled with speaker identities and the silence interval was removed using energy-based voice activity detection. The input features were 40-dimentional Mel-frequency cepstral coefficients (MFCCs), and the spectrogram was extracted based on a 25 ms window with a stride of 10 ms.

- **AF extractor:** Five different models were trained for AF extraction evaluation, including mlp-1f, mlp-9f, mlp-15f, tdnn-9f and tdnn-15f (the first term denotes model name; the second term denotes the number of frames for input). Totally, five hidden layers, each with 512 nodes, were used for non-linear transformation, parametric ReLU (PReLU) was used as the activation function, and the final output layer used a sigmoid function to scale the output values to lie between 0 and 1 as the predicted probabilities of 20 speech attributes. For MLP model, the input was a one-dimensional vector, which directly concatenated all frame-level features as input. For example, mlp-9f concatenated sequential features of 9 frames as input vector. For TDNN model, the frame-level features (two-dimensional matrix) were fed to the model, and each layer received input from the temporal outputs of the previous layer. The main difference between tdnn-9f and tdnn-15f was: tdnn-9f architecture as shown in Figure 1, in which the input of the 2nd-layer and the 4th-layer do not consider the temporal context, but tdnn-15f architecture considered the temporal context as input at the 2nd-layer and the 4th-layer, and the input was the spliced output of the previous layer at frames $\{t-1, t, t+1\}$.
- **Embedding extraction model:** The baseline and the proposed KD models were built using the same architecture. In the training stage, in order to reduce the training cost, the inputs of mini-batches were fixed to three-second long spectral features; each input feature was the MFCC

spectrogram extracted from a 25 ms window with a stride of 10 ms to obtain a sequence of spectral features. For the baseline x-vector as shown in Table 1, there were 512 output nodes in frame1 to frame5 layers and 1,500 output nodes in frame6 layer. For stats-vector model, if the time-delay layer considered temporal context, the subsequence of the output vectors from previous layer were concatenated with the statistics pooling results in the same context to form a new input feature vector, e.g., frame2, frame3 and frame4 layers. For KD, the shared layers further considered the AF recognition loss to learn the speaker-specific information from AF profiles. After that, the statistics pooling produced 3,000 output nodes that were twice the length of the input nodes (consisting of 1,500 nodes for mean operation and 1,500 nodes for standard deviation operation). Segment7 and segment8 layers also consisted of 512 output nodes, and the final softmax layer consisted of 5,994 output probabilities of the training speakers. Rectified linear unit (ReLU) activation function and batch normalization (BN) were applied to each transformation layer for non-linear mapping. For the testing stage, the entire speech signals were fed into the model to extract speaker embeddings. Because the models were trained by segment-based data and the speaker embeddings were extracted by utterance-based data, there is a mismatch between training and testing.

5.3 Analysis of AF Extractor

Because different AFs in a speech pronunciation have rapid change and different durations, five AF extraction models with different input durations were trained to evaluate the performance of AF recognition. In the spectrogram, a single frame is the shortest duration, and most AF durations are shorter than 15 frames. This study analyzed the AF extraction for a duration less than or equal to 15 frames, in which the mlp-1f is the AFs extracted from a single frame (i.e., 25 ms); mlp-9f and tdnn-9f are the AFs extracted from 9 frames (i.e., 105 ms); and mlp-15f and tdnn-15f are the AFs extracted from 15 frames (i.e., 165 ms). As the AF label of LibriSpeech clean set was complete, we further divided the data into a training set and a testing set for AF recognition. In the training set, 921 speakers containing 104,014 recordings were selected randomly for model training. In the testing set, the remaining 251 speakers containing 28,939 recordings were selected for evaluation. As shown in Table 4, this experiment evaluated the performance on multiple attributes prediction and single attribute prediction. The multiple attributes prediction means that the accuracies were calculated based on the correctness of 20-dimensional rounded AF profiles and AF labels; the single attribute prediction means that the accuracies were calculated based on the correctness of each element of the rounded AF profiles and AF labels. Furthermore, the predicted result of each

Table 4: Accuracy comparison of AF recognition on LibriSpeech dataset.

Models	Accuracy (%)	
	Multiple attributes	Single attribute
mlp-1f	51.57	91.74
mlp-9f	**76.70**	**96.46**
tdnn-9f	76.41	96.44
mlp-15f	74.77	96.13
tdnn-15f	75.15	96.23

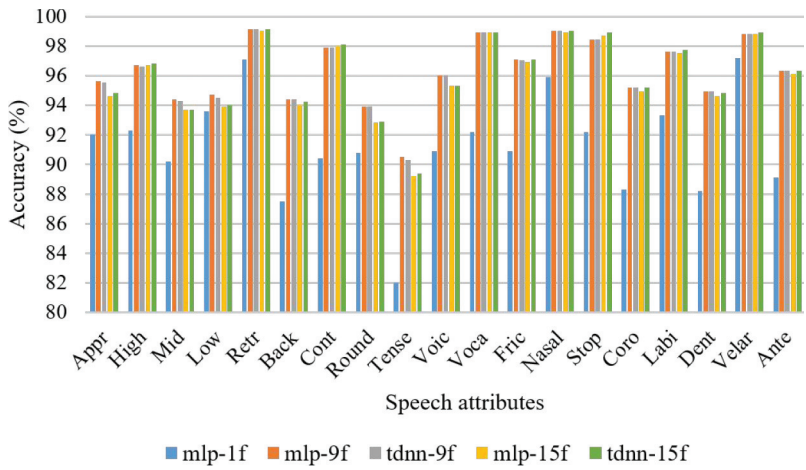

Figure 4: The predicted result of 19 speech attributes (the silence interval was removed) on LibriSpeech dataset. The model with multi-frame input achieved the performance significantly better than model with single-frame input.

speech attribute is shown in Figure 4. Obviously, when the AF extractor was trained by single-frame features, the MLP model achieved the worst accuracy for AF recognition due to insufficient information in the input. The MLP model with an input of 9 frames achieved the best accuracies of 76.70% and 96.46% on multiple attributes prediction and single attribute prediction, respectively. Therefore, using multi-frame features as input can provide more information than single-frame features and achieve the best performance.

Moreover, we further evaluated the performance on phone prediction from the AF profile to ensure the dependency between phone and AF profile. According to Table 3, each AF profile was mapped to a phone label by the Euclidean distance between the predicted AF profile and the phones with one-hot representation of speech attributes. As shown in Figure 5, mlp-1f still

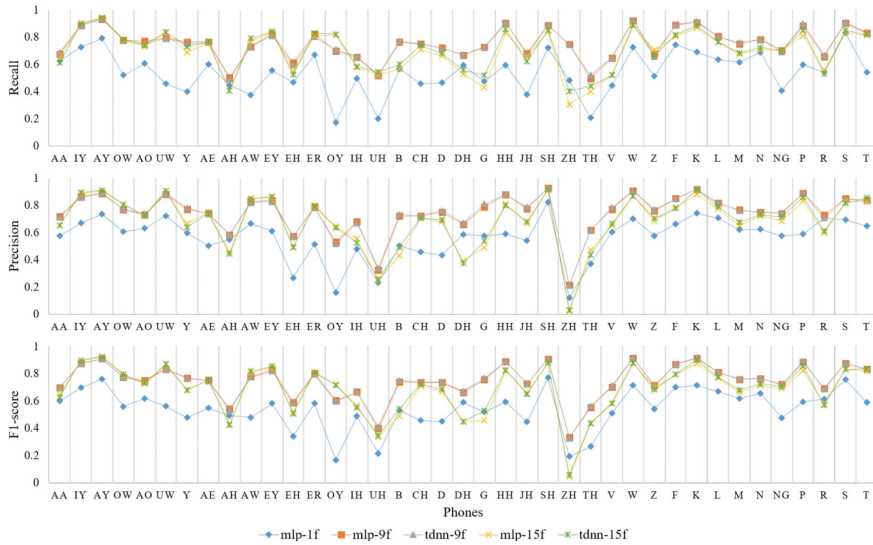

Figure 5: The phone prediction results of AF profiles determined by the Euclidean distance. The upper, middle and lower curves show results of Recall, Precision and F1-score respectively.

performed the worst in phone predictions. It is worth noting that the models with the input of 9 frames and 15 frames on vowel prediction achieved similar performance. But in consonant prediction, the performance of models with an input of 9 frames was better than the models with an input of 15 frames. The reason may be that consonants have short duration and rapid change in pronunciation, and the input with 15 frames has the feature differences at the beginning and the end of a phone. On the other hand, the experiments showed that the performances did not have much difference between the MLP and TDNN models, thus, the mlp-9f (best performance) was selected for the next experiments.

5.4 Speaker Discrimination on AF Profiles

According to the results in [12], using soft targets to train the models can provide more information than hard targets and improve the performance of predictions. As the AF can be used to characterize speaker-specific information, this experiment aimed to prove whether the AF profiles are easier to distinguish speakers than hard targets.

First, the AF extractor was used to obtain the corresponding AF profiles in the VoxCeleb2 and VoxCeleb1 datasets for training and testing, respectively. Next, the x-vector and stats-vector models were trained for speaker embedding

Table 5: Results of SV on AF profiles.

Models	Input types	EER (%)
x-vector	AF hard	21.82
	AF soft	15.50
stats-vector	AF hard	21.63
	AF soft	**15.47**

extraction. In the training stage, each input was fixed to a sequential AF profiles of 3 s. In the testing stage, each corresponding sequential audio AF profile was used to extract speaker embedding. Finally, the VoxCeleb1 (cleaned) list was used to evaluate the performance by PLDA scoring. In the following results, "AF hard" denotes that the input data were obtained by rounding the AF profiles to binary values. "AF soft" denotes that the input data were the AF profiles represented by decimal values. In the SV task, Table 5 shows the equal error rate (EER) for SV. For x-vector and stats-vector, the AF soft method performed better than the AF hard method by 29% and 28% in EER, respectively. Therefore, this experiment showed that AF profiles can be used to characterize speaker-specific information, and using AF soft as input can provide more helpful information to improve the performance of speaker recognition.

5.5 Knowledge Distillation for Speaker Recognition

In this experiment, distilling the information into different layers was analyzed in frame-level transformation. We used the VoxCeleb1 speaker identification task of 1,251 speakers (see iden_split.txt on the VoxCeleb1 webpage) for audio enrollment and evaluation. In the enrollment, the speaker needed to enroll in the system and the corresponding speaker model was constructed. Through enrollment utterances, the obtained speaker-specific embeddings were averaged to form an average embedding as the speaker model. As shown in Table 6, compared to the baseline x-vector and stats-vector systems without KD, the KD with 3 shared layers for x-vector and stats-vector achieved the best accuracies of 63.68% and 62.84%, respectively. Furthermore, we found that when KD was used at higher layers, the performance of KD-based models was worse than the baseline. Conversely, when KD was used at lower layers, the performance of KD-based models was better than the baseline. Thus, distilling the knowledge to a long temporal context (high layers) may cause the learning confusion and degrade the performance, but using KD at lower layers reduces the confusion and achieves the best performance. Combining multimodal results to improve prediction performance has been widely used in recent years. As KD was used in a speaker embedding model, this experiment

Table 6: Accuracy comparison of speaker identification on AF distillation using cosine similarity.

Models	KD shared layers	Accuracy (%)
x-vector	–	62.85
	6	61.65
	5	61.57
	4	62.27
	3	**63.68**
	2	62.94
stats-vector	–	61.28
	6	60.05
	5	61.58
	4	62.36
	3	**62.84**
	2	61.75
x-vector + stats-vector	–	62.60
	3	**63.93**

focused on investigating whether combining the results of multiple KD-based models can further improve the performance. For feature concatenation, we directly concatenated the speaker embeddings obtained from different models to form a combined embedding. According to the results in Table 6, the KD-based x-vector with 3 shared layers and the KD-based stats-vector with 3 shared layers were selected and compared to the method without KD. We can see that in feature concatenation, our proposed KD model achieved the performance better than the method without KD. Therefore, this experiment showed our proposed KD model performed better than the baseline x-vector, and was suitable to be used in multimodality for performance improvement.

6 Conclusion

In this paper, a knowledge distillation model was proposed to distill speaker-specific information to make the model learn more robust speaker representation. First, as AF can be used to characterize speaker-specific information, an AF extractor is proposed to obtain AF profiles of the speech signal. We find that the predicted AF profiles can be used as input features for speaker recognition, and the AF profiles can be seen as the speaker-specific features to provide more helpful information. Next, the knowledge distillation method was used to distill the speaker-specific information of the AF profiles from the teacher model to the student model. The experimental results showed that

our proposed knowledge distillation model achieved the best performance in speaker recognition, and using the feature concatenation of multimodal results can further improve the performance.

In the future, several problems, listed in the followings, should be considered for further improvement.

- As the AF extractor was trained by the LibriSpeech dataset and tested by the VoxCeleb1 dataset, there is a mismatch between training and testing which will degrade the performance.
- In this paper, model training does not apply data augmentation to improve performance.

Therefore, in the future we will explore the learning ability of deep neural networks to deal with the mismatch problems. And combining another mechanism, such as attention, to obtain more robust speaker embedding and increase the degree of aggregation in clustering.

Financial Support

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract No. MOST 111-2221-E-006-150-MY3.

Biographies

Qian-Bei Hong received the B.S. and M.S. degrees in electrical engineering from Southern Taiwan University of Science and Technology, Tainan, Taiwan, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree in the Graduate Program of Multimedia Systems and Intelligent Computing, National Cheng Kung University and Academia Sinica, Tainan, Taiwan. His research interests include multimedia signal processing, deep learning, and speaker recognition.

Chung-Hsien Wu received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1987 and 1991, respectively. Since 1991, he has been with the Department of Computer Science and Information Engineering, NCKU. He became the Chair Professor in 2017. He served as the Deputy Dean of the College of Electrical Engineering and Computer Science, NCKU, from 2009 to 2015. He also worked at Computer Science and Artificial Intelligence Laboratory of Massachusetts Institute of Technology (MIT), Cambridge, MA,

USA, in summer 2003, as a Visiting Scientist. He was the Associate Editor of IEEE Transactions on Audio, Speech and Language Processing (2010–2014), IEEE Transactions on Affective Computing (2010–2014), ACM Transactions on Asian and Low-Resource Language Information Processing, and APSIPA Transactions on Signal and Information Processing (2014∼2020). He was the APSIPA BoG Member in 2019∼2021. He received 2018 APSIPA Sadaoki Furui Prize Paper Award in 2018, and the Outstanding Research Award of Ministry of Science and Technology, Taiwan, in 2010 and 2016. His research interests include deep learning, affective computing, speech recognition/synthesis, and spoken language processing.

Hsin-Min Wang received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1989 and 1995, respectively. In October 1995, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where he is currently a Research Fellow. He also holds a joint appointment as a Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. His major research interests include spoken language processing, natural language processing, multimedia information retrieval, machine learning and pattern recognition. He was an Associate Editor for IEEE/ACM Transactions on Audio, Speech and Language Processing from 2016 to 2020. He is currently on the Editorial Board Member of APSIPA Transactions on Signal and Information Processing. He was the General Co-Chair of ISCSLP2016 and ISCSLP2018 and a Technical Co-Chair of ISCSLP2010, O-COCOSDA2011, APSIPAASC2013, ISMIR2014, and ASRU2019. He was the recipient of the Chinese Institute of Engineers Technical Paper Award in 1995, and the ACM Multimedia Grand Challenge First Prize in 2012. He was an APSIPA Distinguished Lecturer for 2014–2015. He is a Member of the International Speech Communication Association and ACM.

References

[1] S. R. Avutu, D. Bhatia, and B. V. Reddy, "Voice Control Module for Low Cost Local-Map Navigation Based Intelligent Wheelchair," in *Proceedings of the IEEE International Conference on Advanced Computing (IACC)*, 2017, 609–13.

[2] G. Bhattacharya, J. Alam, and P. Kenny, "Deep Speaker Embeddings for Short-Duration Speaker Verification," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2017, 1517–21.

[3] I. Bisio, C. Garibotto, A. Grattarola, F. Lavagetto, and A. Sciarrone, "Smart and Robust Speaker Recognition for Context-aware In-Vehicle Applications," *IEEE Transactions on Vehicular Technology*, 67(9), 2018, 8808–21.

[4] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brummer, "Discriminatively Trained Probabilistic Linear Discriminant Analysis for Speaker Verification," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, 4832–5.

[5] J. Chang and D. Wang, "Robust Speaker Recognition Based on DNN/I-Vectors and Speech Separation," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, 5415–9.

[6] C. P. Chen, Y. C. Huang, C. H. Wu, and K. D. Lee, "Polyglot Speech Synthesis Based on Cross-lingual Frame Selection Using Auditory and Articulatory Features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10), 2014, 1558–70.

[7] Y. H. Chin, J. C. Wang, C. L. Huang, K. Y. Wang, and C. H. Wu, "Speaker Identification Using Discriminative Features and Sparse Representation," *IEEE Transactions on Information Forensics and Security*, 12(8), 2017, 1979–87.

[8] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep Speaker Recognition," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2018.

[9] S. Cumani and P. Laface, "Speaker Recognition Using E–Vectors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(4), 2018, 736–48.

[10] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4), 2011, 788–98.

[11] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of I-Vector Length Normalization in Speaker Recognition Systems," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2011, 249–52.

[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[13] Q. B. Hong, C. H. Wu, M. H. Su, and H. M. Wang, "Sequential Speaker Embedding and Transfer Learning for Text-Independent Speaker Identification," in *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, 827–32.

[14] Q. B. Hong, C. H. Wu, H. M. Wang, and C. L. Huang, "Combining Deep Embeddings of Acoustic and Articulatory Features for Speaker Identification," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, 7589–93.

[15] Q. B. Hong, C. H. Wu, H. M. Wang, and C. L. Huang, "Statistics Pooling Time Delay Neural Network Based on X-Vector for Speaker Verification," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, 6849–53.

[16] M. T. S. Al-Kaltakchi, W. L. Woo, S. S. Dlay, and J. A. Chambers, "Study of Statistical Robust Closed Set Speaker Identification with Feature and Score-Based Fusion," in *IEEE Statistical Signal Processing Workshop*, 2016, 1–5.

[17] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "I-Vector Based Speaker Recognition on Short Utterances," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2011, 2341–4.

[18] T. Kinnunen, L. Juvela, P. Alku, and J. Yamagishi, "Non-Parallel Voice Conversion Using I-Vector PLDA: Towards Unifying Speaker Verification and Transformation," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, 5535–9.

[19] K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining Acoustic and Articulatory Feature Information for Robust Speech Recognition," *Speech Communication*, 37(3–4), 2002, 303–19.

[20] Z. Kozhirbayev, B. A. Erol, A. Sharipbay, and M. Jamshidi, "Speaker Recognition for Robotic Control via an IoT Device," in *2018 World Automation Congress (WAC)*, 2018, 1–5.

[21] C. H. Lee and S. M. Siniscalchi, "An Information-Extraction Approach to Speech Processing: Analysis, Detection, Verification, and Recognition," *Proceedings of the IEEE*, 101(5), 2013, 1089–115.

[22] M. Li, J. Kim, A. Lammert, P. K. Ghosh, V. Ramanarayanan, and S. Narayanan, "Speaker Verification Based on the Fusion of Speech Acoustics and Inverted Articulatory Signals," *Computer Speech & Language*, 36, 2016, 196–211.

[23] N. Li, D. Tuo, D. Su, Z. Li, and D. Yu, "Deep Discriminative Embeddings for Duration Robust Speaker Verification," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2018, 2262–6.

[24] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker Embedding Extraction with Phonetic Information," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2018, 2247–51.

[25] A. McCree, G. Sell, and D. Garcia-Romero, "Extended Variability Modeling and Unsupervised Adaptation for PLDA Speaker Recognition," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2017, 1552–6.

[26] V. Mitra, G. Sivaraman, C. BarteIs, H. Nam, W. Wang, C. Espy-Wilson, D. Vergyri, and H. Franeo, "Joint Modeling of Articulatory and Acoustic Spaces for Continuous Speech Recognition Tasks," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, 5205–9.

[27] V. Mitra, V. W. Wang, C. Bartels, H. Franco, and D. Vergyri, "Articulatory Information and Multiview Features for Large Vocabulary Continuous Speech Recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, 5634–8.

[28] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A Largescale Speaker Identification Dataset," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2017.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, and J. Silovsky, "The Kaldi Speech Recognition Toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[30] S. J. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007, 1–8.

[31] R. Rasipuram and M. Magimai-Doss, "Articulatory Feature Based Continuous Speech Recognition Using Probabilistic Lexical Modeling," *Computer Speech & Language*, 36, 2016, 233–59.

[32] J. Rohdin, A. Silnova, M. Diez, O. Plchot, and L. Matejka P.and Burget, "End-to-End DNN Based Speaker Recognition Inspired by I-Vector and PLDA," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, 4874–8.

[33] L. Sari, S. Thomas, M. Hasegawa-Johnson, and M. Picheny, "Pre-Training of Speaker Embeddings for Low-Latency Speaker Change Detection in Broadcast News," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 6286–90.

[34] H. P. Shen, J. F. Yeh, and C. H. Wu, "Speaker Clustering Using Decision Tree-based Phone Cluster Models with Multi-Space Probability Distributions," *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5), 2011, 1289–300.

[35] S. M. Siniscalchi, T. Svendsen, and C. H. Lee, "Toward a Detector-Based Universal Phone Recognizer," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, 4261–4.

[36] D. Snyder, D. Garcia-Romero, and D. Povey, "Time Delay Deep Neural Network-Based Universal Background Models for Speaker Recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2015, 92–7.

[37] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2017, 999–1003.

[38] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-Speaker Conversations Using X-Vectors," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 5796–800.

[39] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, 5329–33.

[40] G. Srinivasan, A. Illa, and P. K. Ghosh, "A Study on Robustness of Articulatory Features for Automatic Speech Recognition of Neutral and Whispered Speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 5936–40.

[41] Y. Tang, G. Ding, J. Huang, X. He, and B. Zhou, "Deep Speaker Embedding Learning with Multi-Level Pooling for Text-Independent Speaker Verification," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 6116–20.

[42] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep Neural Networks for Small Footprint Text-Dependent Speaker Verification," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, 4052–6.

[43] J. Villalba, N. Brummer, and N. Dehak, "Tied Variational Autoencoder Backends for I-Vector Speaker Recognition," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2017, 1004–8.

[44] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Transactions on Audio, Speech, and Language Processing*, 37(3), 1989, 328–39.

[45] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker Diarization with LSTM," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, 5239–43.

[46] S. Wang, Z. Huang, Y. Qian, and K. Yu, "Discriminative Neural Embedding Learning for Short-Duration Text-Independent Speaker Verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11), 2019, 1686–96.

[47] C. H. Wu, H. P. Shen, and C. S. Hsu, "Code-Switching Event Detection by Using a Latent Language Space Model and the Delta-Bayesian Information Criterion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(11), 2015, 1892–903.

[48] C. H. Wu, H. P. Shen, and Y. T. Yang, "Chinese-English Phone Set Construction for Code-Switching ASR Using Acoustic and DNN-Extracted Articulatory Features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4), 2014, 858–62.

[49] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How Transferable are Features in Deep Neural Networks?" In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2014, 3320–8.

[50] D. Yu, S. M. Siniscalchi, L. Deng, and C. H. Lee, "Boosting Attribute and Phone Estimation Accuracies with Deep Neural Networks for Detection-Based Speech Recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, 4169–72.

[51] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification," in *Proceedings of the International Speech Communication Association (Interspeech)*, 2018, 3573–7.