INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

# Introducing Self-Supervised Phonetic Information for Text-Independent Speaker Verification

*Ziyang Zhang, Wu Guo, Bin Gu*

The Department of Electronic Engineering and Information Science (EEIS)
University of Science and Technology of China, Hefei, China

zyzhang182@mail.ustc.edu.cn, guowu@ustc.edu.cn, bin2801@mail.ustc.edu.cn

## Abstract

This paper presents a novel multi-task learning framework by introducing self-supervised phonetic information for deep speaker embedding extraction. The primary task is still to classify speakers, but we consider an auxiliary task to identify phoneme boundaries in speech signals following the Noise Contrastive Estimation principle. To further utilize self-supervised information to assist speaker feature learning, the features of intermediate layers in the main task are refined by the features of corresponding layers in the auxiliary task through masking and biasing operations. We use the VoxCeleb1 and CN-Celeb datasets for performance evaluation, which consistently verifies the efficacy of the proposed method.

**Index Terms**: speaker verification, self-supervised learning, multi-task learning, noise contrastive estimation

## 1. Introduction

Speaker verification (SV) aims to verify a person's identity using speech/voice signals. There are two types of speaker verification: text-dependent and text-independent SV [1]. The latter has been widely used since it does not require the speaker to use a particular passphrase, and this paper focuses on text-independent SV. In the past decade, deep neural networks (DNN) based method, known as x-vector [2], has become the mainstream approach in the text-independent SV. Various architectures, including the time-delay neural network (TDNN) [3], the convolutional neural network (CNN) [4], and the ResNet [5, 6, 7], have been successfully applied to different SV tasks.

In principle, speech signals implicitly contain speaker traits and phonetic contents [8, 9]. The mixing of two kinds of information has a significant influence on SV research. Numerous researchers have recently attempted to use phonetic content information in the field of SV. Multi-task learning (MTL) based deep neural networks are used to implement state-of-the-art research in this area [9, 10, 11]. In [9], phonetic vectors generated from the additionally constructed ASR branch, are connected to an x-vector network. In [10], a phoneme-aware attention pooling method is proposed to better capture long-term variations in speaker characteristics. In [11], a phonetic attention mask (PAM) is applied to dynamically assign weights produced by the digit recognition branch to speaker features. Most of the above methods performed joint training of speaker classification and phonetic content-related auxiliary tasks under supervision, by introducing phonetic information into the deep layer of the speaker network to help obtain speaker clues. However, phoneme recognizer consumes many computing resources and may generate inaccurate labels due to the mismatch between training and testing scenarios. In addition, in the field of SV, datasets containing both transcripts of contents and the speaker's identity are scarce. Using two distinct datasets for alternate training may lead to suboptimal results in speaker embedding learning.

Unsupervised or self-supervised learning (SSL) methods are viable alternatives when supervised training data is scarce. Recently, SSL has shown promissing results in speech processing, especially in automatic speech recognition (ASR) [12, 13, 14]. The pre-trained models trained with SSL, such as Wav2vec2.0 and WavLM [15], have been successfully applied to different downstream speech tasks. Contrastive learning [16] is commonly-used in SSL. In wav2vec2.0 [13], the potential representation is distinguished from distractors through the contrastive loss, for learning effective speech representation. In [17], Noise Contrastive Estimation (NCE) loss is applied to distinguish between pairs of adjacent frames and random non-adjacent frames for identifying spectral changes on an unsupervised phoneme boundary detection task.

Although self-supervised approaches liberate the acquisition of labels, their performance still cannot match with supervised models if they do not include supervised training or fine-tuning. This paper focuses on obtaining deep discriminative speaker embedding with joint supervised and self-supervised learning from scratch using the database with only speaker labels. The proposed network has two branches with almost the same architecture. One branch is used to classify speakers with supervised learning, and the other is used to identify spectral changes in the signal using the NCE loss. With the NCE loss, the auxiliary task can learn phonetic information in a self-supervised way. In order to enhance the deep speaker embedding extraction with the auxiliary self-supervised information, the features of intermediate layers within the main task are refined by the features of the corresponding layers in the auxiliary task through multiplying and biasing operations. We carry out experiments on the VoxCeleb1 and CN-Celeb datasets. Compared with the baseline system, the proposed system can achieve a consistent performance improvement.

The remainder of this paper is organized as follows. The x-vector baseline system is illustrated in Section 2. The proposed method is presented in Section 3. Sections 4 and 5 present the experimental setup, results, and analysis, respectively. Finally, conclusions are given in Section 6.

## 2. Baseline system

The detailed configuration of our baseline is shown in Table 1. It uses the Resnet34 backbone [18] to extract speaker representations at the frame level. The input acoustic feature first passes through a convolutional layer with a kernel size of 7×7 and a stride of 2×2. The four residual stages include repeated stacking of 3×3 convolution kernels, and the number of channels

10.21437/Interspeech.2023-1558

Table 1: *The ResNet34 baseline structure.*

| Layer name | Structure | Output size |
|---|---|---|
| Conv_0 | 7 × 7, 32, stride 2 | $T/2 \times 32 \times 32$ |
| Conv1_x | $\begin{bmatrix} 3\times3 & 32 \\ 3\times3 & 32 \end{bmatrix} \times 3$ | $T/2 \times 32 \times 32$ |
| Conv2_x | $\begin{bmatrix} 3\times3 & 64 \\ 3\times3 & 64 \end{bmatrix} \times 4$ | $T/2 \times 16 \times 64$ |
| Conv3_x | $\begin{bmatrix} 3\times3 & 128 \\ 3\times3 & 128 \end{bmatrix} \times 6$ | $T/2 \times 8 \times 128$ |
| Conv4_x | $\begin{bmatrix} 3\times3 & 256 \\ 3\times3 & 256 \end{bmatrix} \times 3$ | $T/2 \times 4 \times 256$ |
| statistic pooling& 2 fully connected layers, AM-Softmax | | |

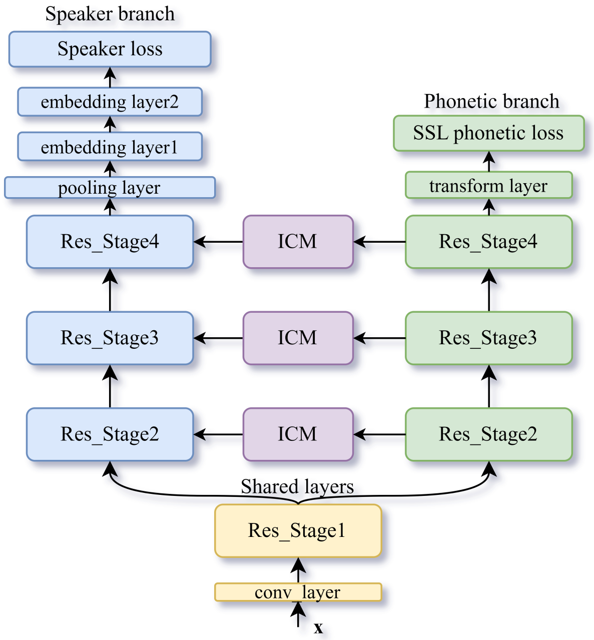

Figure 1: *The proposed ResNet34 structure with dual branches and information control module (DBICM).*

is set to 32, 64, 128, and 256, respectively. Down-sampling is only performed in the first block of each residual stage with a stride of 1×2. Afterward, the statistics pooling layer converts the frame-level features into a fixed-length utterance-level speaker representation. The utterance-level features are transformed into speaker embeddings by two fully connected layers. Finally, the classification solution is given by the AM-Softmax [19] output layer.

## 3. Proposed method

As depicted in Figure 1, the proposed network adopts a Dual-Branch structure with the auxiliary information control module (ICM), termed by DBICM in this work. DBICM mainly comprises three modules: the main-task speaker-classification branch based on the conventional ResNet34 architecture, the auxiliary-task self-supervised learning branch using the cloned architecture of the main task except for some top layers, and the layer-wise ICM between the two branches.

The dual branches encode the speaker and phonetic content features through the AM-Softmax and the NCE losses, respectively. As we know, the bottom layers of deep neural networks model the low-level information, and the top layers extract high-level information. For simplicity, we share the first stage of ResNet34 between the two tasks since they both extract similar low-level information from acoustic features. The two branches' top stages of the ResNet34 extract complementary information. The output features from the auxiliary task are transformed in ICM, which is used to refine the features of the main task's corresponding layers through masking and biasing operations.

### 3.1. The auxillary self-supervised task

As mentioned above, SSL can learn effective speech signal representations without requiring transcriptions of the training set. To balance the performance and requirement of computing resources, we adopt a relatively simple SSL training scheme, which was shown to be effective on unsupervised phoneme boundary detection [17, 20]. The NCE loss is widely used for SSL. In this work, NCE is applied by comparing the similarity between adjacent and non-adjacent frames of utterances. We can learn contextual, especially phonetic, information from this meticulously crafted loss function.

We use a similar ResNet34 structure in the main task to implement the layer-wise ICM in this work. To obtain the latent speech representations, we add a transform layer to the final stage of ResNet34 in the SSL branch. Through the upsampling and flattening operations in the transform layer, the feature map's dimension is changed to 2D, and the time axis size is restored to $T$:

$$Z = \text{Transform}(H) = \text{Flatten}(\text{Upsampling}(H)), \tag{1}$$

where $Z = (\boldsymbol{z}_1, \boldsymbol{z}_2 \ldots \boldsymbol{z}_T) \in \mathbb{R}^{T\times G}$ represents the latent speech representations of the SSL branch. $T$ and $G$ represent the number of frames and the feature dimension, respectively. $H \in \mathbb{R}^{T/2\times F\times C}$ denotes the feature map output by the last residual stage. $F$ and $C$ respectively denote the frequency and channel dimensions.

The contrastive loss can be formulated for each latent speech representation feature $\boldsymbol{z}_i \in \mathbb{R}^G$, given by:

$$L_c(\boldsymbol{z}_i) = -\log \frac{e^{\mathrm{sim}(\boldsymbol{z}_i,\boldsymbol{z}_{i+1})}}{\sum_{\boldsymbol{z}_j\in\{\boldsymbol{z}_{i+1}\}\cup D_K(\boldsymbol{z}_i)} e^{\mathrm{sim}(\boldsymbol{z}_i,\boldsymbol{z}_j)}}, \tag{2}$$

where $D_K(\boldsymbol{z}_i)$ represents a set of $K$ non-adjacent frames of $z_i$. The positive sample pair $(\boldsymbol{z}_i, \boldsymbol{z}_{i+1})$ and negative sample pair $(\boldsymbol{z}_i, \boldsymbol{z}_j)$, $\boldsymbol{z}_j \in D_K(\boldsymbol{z}_i)$, respectively, represent frames that are adjacent to the current frame and frames that are not adjacent to it. Calculating the cosine value, $\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \boldsymbol{z}_i^T \boldsymbol{z}_j / \|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|$, yields the similarity between two vectors. The total NCE loss function is then given by:

$$L_c = \frac{1}{NT} \sum_N \sum_T L_c(\boldsymbol{z}_i), \tag{3}$$

where $N$ denotes the total number of features in training.

### 3.2. Information control module (ICM)

Since the two branches in DBICM have similar structures but different training objectives, the SSL branch can provide complementary information for speaker feature learning on the main branch. To refine the feature maps of the main branch at stage 2/3/4, we design layer-wise ICMs, see Figure 1.

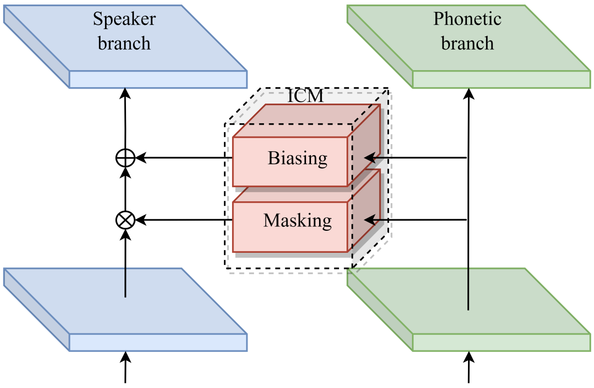

Figure 2: *The Structure of ICM.*

As shown in Figure 2, ICM dynamically generates two sets of parameters, $w^l$ and $b^l$, to calibrate the feature maps of the $l^{th}$ stage on the main branch through masking and biasing operations, which are calculated as:

$$w^l = 1 - \sigma\left(\mathrm{Conv}\,2D\left(H^l\right)\right), \tag{4}$$

$$b^l = 1 - \mathrm{Conv}\,2D\left(H^l\right), \tag{5}$$

$$\widetilde{H}_s^l = \left(H_s^l \otimes w^l\right) \oplus b^l, \tag{6}$$

where $H_s^l$ and $\widetilde{H}_s^l$ denote the feature maps of the $l^{th}$ stage on the main branch before and after calibration, and $H^l$ are the features of the corresponding stage on the SSL branch. Conv2D denotes 2D CNN, which uses a $1\times1$ convolution kernel. Additionally, $\sigma(.)$ denotes the sigmoid function, and the element-wise multiplication and addition are represented by $\otimes$ and $\oplus$.

### 3.3. Overall loss function

In combination with the original speaker classification loss, the final loss function can be written as:

$$L = L_{spk} + \lambda_c L_c, \tag{7}$$

where $L_{spk}$ and $L_c$ denote the AM-Softmax loss for speaker classification and the contrastive loss for SSL, respectively. $\lambda_c$ is the hyper-parameter.

## 4. Experimental Setups

### 4.1. Datasets and Evaluation Indicators

VoxCeleb1 [21] and CN-Celeb [22, 23] are used throughout experiments. VoxCeleb1 contains over 100,000 utterances for 1,251 celebrities, extracted from online videos. The speakers vary in professions, ages, ethnicities, and accents. In the experiments, the development part of Voxceleb1 is used as the training dataset, which contains 148,642 utterances from 1,211 speakers. There are 40 speakers and 4,874 utterances in the Voxceleb1 test part. The CN-Celeb corpus contains speeches by Chinese celebrities. The entire dataset is divided into CN-Celeb.T and CNC-Eval: the former is used for training, consisting of 632,736 utterances from 2,793 speakers, 1285 hours in total, the latter is used as the test set.

Considering the databases are established in complex acoustic environments with background noise, far-field, and irregular durations, we also evaluate the necessity of data augmentation for comparison. Augmented data with reverb, noise, music, and babble are used to increase the diversity of the training data.

The system's performance evaluation metrics are the equal error rate (EER) and the minimum cost detection function (minDCF) with a prior target probability $P_{tar}$ of 0.01.

### 4.2. Implementation details

The experiments use 64-dimensional log-mel filter-bank (Fbank) energy features as acoustic features. In addition, all the features are processed with mean normalization and energy-based voice active detection (VAD) over a 3-second sliding window. In the training set, Voxceleb1 and CN-Celeb utterances are randomly cropped to lengths of 2-4 s. 64 utterances with the same duration are assigned to the same mini-batch.

All models are built using the Tensorflow toolkit [24]. Kaldi Toolkit [25] is used for, e.g., data processing, feature extraction, and the PLDA [26] backend. The network is optimized using the Adam optimizer, and the learning rate gradually decreases from 1e-3 to 1e-4. The AM-softmax [19] is used as the loss function for speaker classification, and the margin $m$ and scaling factor $s$ are set to 0.15 and 30, respectively. We build three systems for comparison, and the configurations of each system are listed as follows:

**Baseline**: ResNet34 structure with AM-Softmax loss.

**DB-Phone**: Multi-task learning framework with supervised phoneme classification as an auxiliary task. The "FisherMono" model of the BUT phoneme recognizer [27], which is independent of the multi-task network training process, is employed for phoneme label recognition of each speech frame. The frame-level network of two branches is the same as the baseline, sharing the layers in the first residual stage and before. For the auxiliary branch, the feature $Z$ in Equation 1 output by the transform layer passes through the Softmax layer is used to calculate the cross-entropy loss with corresponding phoneme labels.

**DBICM**: The proposed structure in this paper, i.e. the ResNet34 network with dual branches and information control module for calibrating the feature maps of the speaker classification branch, as shown in Figure 1. ICM is employed after each residual stage. In the SSL branch, the number of negative samples in Equation 2 is set as $K = 3$. Comparing the DBICM and DB-Phone systems, the number of model parameters is approximately similar.

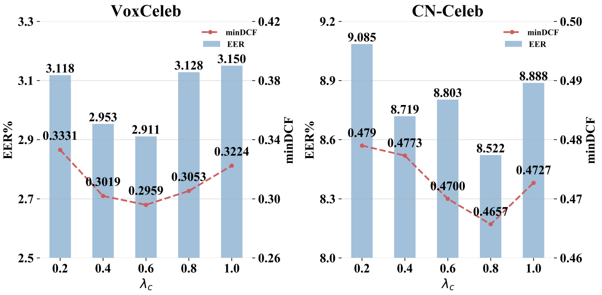

Figure 3: *The Results of DBICM on VoxCeleb and CN-Celeb datasets with Different $\lambda_c$.*

Table 2: *Speaker verification results on the VoxCeleb datasets with different systems.*

| System | Data | VoxCeleb | |
|---|---|---|---|
| | Aug | EER% | minDCF |
| Baseline | × | 3.934 | 0.3882 |
| DB-Phone | | 3.712 | 0.4186 |
| DBICM | | 3.388 | 0.3559 |
| Baseline | ✓ | 3.388 | 0.3525 |
| DB-Phone | | 3.102 | 0.3384 |
| DBICM | | 2.911 | 0.2959 |
| SLA-DV [28] | ✓ | 3.747 | 0.3658 |
| AM-PPL [29] | | 3.321 | 0.3234 |

Table 3: *Speaker verification results on the CN-Celeb datasets with different systems.*

| System | Data | CN-Celeb | |
|---|---|---|---|
| | Aug | EER% | minDCF |
| Baseline | ✓ | 9.107 | 0.4951 |
| DB-Phone | | 8.910 | 0.4971 |
| DBICM | | 8.522 | 0.4657 |

# 5. Result

## 5.1. Hyper-parameter selection

In Equation 7, the hyper-parameter $\lambda_c$ regulates the proportion of main and auxiliary tasks. We evaluate the impact of different $\lambda_c$, and the experimental results are depicted in Figure 3. When $\lambda_c$ equals 0.6 or 0.8, the system achieves the best performance on VoxCeleb1 or CN-Celeb. Unless otherwise specified, we will keep this choice of hyper-parameter for the proposed DBICM system in the sequel.

## 5.2. Performance of different systems

Then, we compare the proposed DBICM with comparison systems and several recent approaches. On VoxCeleb1, DB-Phone and DBICM, two MTL methods, can improve over the baseline, as shown in Table 2. This demonstrates that the usage of auxiliary information in SV is effective. The proposed DBICM outperforms the other two comparison systems on EER and minDCF metrics, regardless of the inclusion of data augmentation. This is due to the fact that the SSL branch can effectively introduce the phonetic information to help speaker extraction, while the inaccurate labels recognized by the phoneme recognizer in DB-Phone affect the robustness of the SV.

We also list the results of two recent experiments combining phonetic information into SV. In [28], the maximum mean difference loss is utilized to alleviate the mismatch between the speaker and phoneme subnetworks. [29] adopts a multi-task learning approach to build an auxiliary branch with the pseudo-phoneme label loss. We directly cite the published results. The proposed DBICM outperforms these two methods. When data augmentation is used, the proposed DBICM achieves the best results with an EER of 2.911% and minDCF of 0.2959. Compared with the baseline, we can obtain a 14.1% and 16.1% reduction in EER and minDCF metrics, respectively.

Furthermore, we conduct experiments on CN-Celeb to demonstrate the robustness and generalization ability of the proposed DBICM algorithm. For simplicity, we only list results with data augmentation in Table 3. The proposed DBICM can also achieve consistent improvements on the CN-Celeb dataset. However, due to its more complex scenarios and the domain mismatch issue, the performance improvement on CN-Celeb is smaller than that on VoxCeleb1.

Table 4: *Effects of different operators in ICM, where " w/o " means "without".*

| System | VoxCeleb | | CN-Celeb | |
|---|---|---|---|---|
| | EER% | minDCF | EER% | minDCF |
| DBICM | 2.911 | 0.2959 | 8.522 | 0.4657 |
| w/o Biasing | 3.028 | 0.3068 | 8.859 | 0.4744 |
| w/o Masking | 3.102 | 0.3174 | 8.865 | 0.4812 |
| w/o Biasing & Masking | 3.298 | 0.3215 | 9.006 | 0.4931 |

## 5.3. Evaluating the different operators in ICM

As shown in Equation 6, the masking and biasing operators are the two operators in ICM. Therefore, we examine the individual effect by removing them separately. The results are shown in Table 4, where the performance of the complete DBICM system is also shown in the first row. Note that the system with the removal of both operations is still a multi-task framework.

As shown in Table 4, removing any operator results in noticeable performance degradation, implying the necessity of the operators in ICM. Furthermore, removing the masking operator results in a more significant performance drop than the other case, indicating that the multiplication operation plays a more important role in the ICM.

Table 5: *Results of applying ICM at different stages.*

| Stage | VoxCeleb | | CN-Celeb | |
|---|---|---|---|---|
| | EER% | minDCF | EER% | minDCF |
| 2nd | 3.287 | 0.3599 | 9.333 | 0.4879 |
| 3rd | 3.070 | 0.3458 | 9.017 | 0.4818 |
| 4th | 2.975 | 0.3160 | 8.634 | 0.4707 |
| 2,3,4 | 2.911 | 0.2959 | 8.522 | 0.4657 |

## 5.4. Applying ICM at different stages of ResNet

As ResNet34 consists of four residual stages with distinct feature dimensions, we evaluate the effect of applying ICM to different stages. Specifically, we apply ICM to the second, third, and fourth residual stages of the ResNet backbone. The results of DBICM with data augmentation are listed in Table 5.

In the deeper stages of the two branches, the features have higher-level information, which can provide more complementary information for speaker characteristics modeling. Finally, applying the ICM to all three residual stages can obtain the best result. A reasonable explanation is that different stages can provide different levels of information.

# 6. Conclusions

In this paper, we proposed a novel MTL framework to learn robust deep speaker embedding, where the SSL is used as an auxiliary task. In addition, information from the auxiliary task is used to calibrate the deep features of the main task branch using ICM modules. Experiments on Voxceleb1 and CN-Celeb prove that the proposed method can consistently improve the speaker verification system.

# 7. Acknowledgements

This work was supported by the USTC-CMB Joint Laboratory of Artificial Intelligence.

# 8. References

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

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

[3] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, vol. 2017, 2017, pp. 999–1003.

[4] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[5] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[6] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 74–81. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2018-11

[7] B. Gu and W. Guo, "Dynamic convolution with global-local information for session-invariant speaker representation learning," *IEEE Signal Processing Letters*, vol. 29, pp. 404–408, 2021.

[8] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, and J. Černocký, "On the Usage of Phonetic Information for Text-Independent Speaker Embedding Extraction," in *Proc. Interspeech 2019*, 2019, pp. 1148–1152.

[9] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker Embedding Extraction with Phonetic Information," in *Proc. Interspeech 2018*, 2018, pp. 2247–2251.

[10] Y. Liu, Z. Li, L. Li, and Q. Hong, "Phoneme-Aware and Channel-Wise Attentive Learning for Text Dependent Speaker Verification," in *Proc. Interspeech 2021*, 2021, pp. 101–105.

[11] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Neural acoustic-phonetic approach for speaker verification with phonetic attention mask," *IEEE Signal Processing Letters*, vol. 29, pp. 782–786, 2022.

[12] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[14] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[15] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[16] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[17] F. Kreuk, J. Keshet, and Y. Adi, "Self-Supervised Contrastive Learning for Unsupervised Phoneme Segmentation," in *Proc. Interspeech 2020*, 2020, pp. 3700–3704.

[18] 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.

[19] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[20] H. Liu, L. P. Garcia Perera, A. Khong, S. Styles, and S. Khudanpur, "PHO-LID: A Unified Model Incorporating Acoustic-Phonetic and Phonotactic Information for Language Identification," in *Proc. Interspeech 2022*, 2022, pp. 2233–2237.

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

[22] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[23] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vipperla, T. F. Zheng, and D. Wang, "Cn-celeb: multi-genre speaker recognition," *Speech Communication*, vol. 137, pp. 77–91, 2022.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *Osdi*, vol. 16, no. 2016. Savannah, GA, USA, 2016, pp. 265–283.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[26] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[27] A. Silnova, P. Matejka, O. Glembek, O. Plchot, O. Novotnỳ, F. Grezl, P. Schwarz, L. Burget, and J. Cernockỳ, "But/phonexia bottleneck feature extractor." in *Odyssey*, 2018, pp. 283–287.

[28] J. Wang, T. Lan, J. Chen, C. Luo, C. Wu, and J. Li, "Phoneme-aware adaptation with discrepancy minimization and dynamically-classified vector for text-independent speaker verification," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6737–6745.

[29] M. Niu, L. He, Z. Fang, B. Zhao, and K. Wang, "Pseudo-phoneme label loss for text-independent speaker verification," *Applied Sciences*, vol. 12, no. 15, p. 7463, 2022.