INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

The DKU-Duke-Lenovo System Description for the Fearless Steps Challenge Phase III

Weiqing Wang[1], *Danwei Cai*[1], *Jin Wang*[2], *Qingjian Lin*[2], *Xuyang Wang*[2], *Mi Hong*[2], *Ming Li*[1*]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]AI Lab, Lenovo Research, Beijing, China

{weiqing.wang, danwei.cai, ming.li369}@duke.edu,
{wangjin19, linqj3, wangxy60, hongmi1}@lenovo.com

Abstract

This paper describes the systems developed by the DKU-Duke-Lenovo team for the Fearless Steps Challenge Phase III. For the speech activity detection (SAD) task, we employ the U-Net-based model which has not been used for SAD before, observing a DCF of 1.915% on the eval set. For the speaker identification (SID) task, we adopt the ResNet-SE and ECAPA-TDNN model, and we obtain a Top-5 accuracy of 86.21%. For the speaker diarization (SD) task, we employ several different clustering methods. Besides, domain adaptation, system fusion, and Target-Speaker Voice Activity Detection (TS-VAD) significantly improve the SD performance. We obtain a DER of 12.32% on track 2, and the major contribution is from our ResNet-based TS-VAD model. We finally achieve a first-place ranking for SD and SID and a second-place for SAD in the challenge.

Index Terms: Speech Activity Detection, Speaker Identification, Speaker Diarization, Target-Speaker Voice Activity Detection

1. Introduction

Fearless Steps Challenge 2021 Phase-3 (FS3) [1, 2, 3] is a speech challenge hosted by the Center for Robust Speech Systems (CRSS) at the University of Texas at Dallas. This challenge is designed for the digitization, recovery, and diarization of 19,000 hours of original analog audio data, as well as the development of algorithms to extract meaningful information from this multichannel naturalistic data resource [1, 2, 3]. Different from the previous Fearless Steps Challenge, FS3 includes more tracks in this competition: Speech Activity Detection (SAD), Speaker Identification (SID), Speaker Diarization (SD), Automatic Speech Recognition (ASR), and Conversational Analysis (CONV). In this paper, we focus on the SAD, SID, and SD tasks.

SAD is used for tagging the speech regions in the audio streams and often serves as the front-end of other speech processing modules. Recently, U-Net has shown excellent performance in image segmentation [4]. Considering that the SAD is to segment the speech and the nonspeech region in audio, which is similar to the image segmentation task, we adopt the U-Net in our SAD system. Furthermore, SpecAugment [5] is introduced as a data augment method to improve the performance. Finally, the result is smoothed by a Hidden Markov Model (HMM).

The SID task aims at the identification of a person from characteristics of voices. In the past few years, the performance of SID has been significantly improved with the i-vector-based method [6] and the speaker embedding modeling using deep neural networks [7, 8]. In the SID task, we employ the ResNet-SE [8, 9] and ECAPA-TDNN [10] model to get a robust speaker embedding.

SD is the task of splitting audio into homogeneous pieces which belong to the same speaker, and it aims to determine "who spoke when" in an audio or a video recording [11, 12, 13]. Generally speaking, the traditional SD system contains several sub-modules: SAD, segmentation, speaker embedding extraction and clustering, etc. For the SD task, we first employ the LSTM-based model [14] and the attention-based model [15] with spectral clustering. In addition, we also perform agglomerative hierarchical clustering (AHC) directly on the speaker embedding. Finally, we employ target-speaker voice activity detection (TS-VAD) [16, 17, 18] and system fusion [19, 20] to further improve the SD performance. The major contribution is our ResNet-based TS-VAD model, which takes the Deep ResNet vector as the speaker embedding.

The rest of this paper is organized as follows. Section 2, 3 and 4 introduce the SAD, SID, and SD task, respectively. Experimental results are presented in Section 5. Conclusion is provided in Section 6.

2. Speech Activity Detection

2.1. U-Net-based SAD

Although U-Net [21] has been used in multi-task learning for speech enhancement (SE) and SAD, the SAD module relies on the time-frequency masks obtained by U-Net-based SE. Here the SAD is regarded as a classification task and the U-Net is adopted to learn from the spectrum.

Figure 1 shows the structure of the U-Net, it consists of two paths: the encoding path (left side) and the decoding path (right side). In the encoding path, the input feature is firstly passed into the CNN block. It consists of 2 convolution layers with the kernel size of 3×3, the number of the kernel is 16. Batch Normalization (BN) [22] and Rectified Linear Unit (ReLU) [23] is applied after each convolution layer. The output of the block is passed to a 2×2 max-pooling layer, then the subsampled feature maps are passed to the next CNN block with the kernel number doubled. In the decoding path, the feature maps are passed to a deconvolution layer with the number of kernels halved and concatenate with the feature maps from the encoding path, then passed to the next CNN block with the kernel number halved. After the decoding path, a 1×1 convolution layer followed by a linear layer and softmax layer is applied to obtain the probability.

* Corresponding Author

http://dx.doi.org/10.21437/Interspeech.2021-235

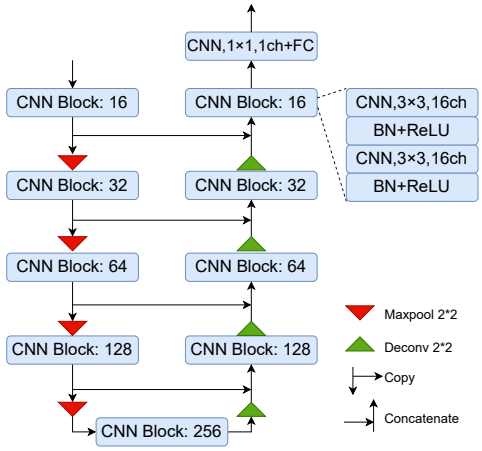

Figure 1: *Block diagram of the U-Net used for SAD [21]*

2.1.1. SpecAugment

SpecAugment [5] is a simple data augmentation method and has shown high effectiveness in enhancing the performance of the ASR task. By viewing the input features as an image, SpecAugment applies three basic augmentations: 1. time warping: A random point along the timeline passing through the center of the image is to be warped either to the left or right. 2. frequency masking: several consecutive frequency channels are masked. 3. time masking: several time steps are masked. In this paper, we only apply the frequency masking to the input feature.

2.1.2. HMM Smoothing

An HMM is used for final smoothing. It consists of 1 consecutive state for noise and 40 for speech, the probability of staying in the state and switching are both 0.5. The output of the U-Net is regarded as the emission probability of the HMM.

3. Speaker Identification

We train two neural network-based systems for the speaker identification task, i.e., ResNet-SE [8, 9] and ECAPA-TDNN [10]. The two systems are firstly trained on a large-scale speaker recognition dataset and then fine-tuned on the FS3 SID training data.

3.1. ResNet-SE

The ResNet-based system has the same network architecture as [24]. It contains a ResNet34 front-end pattern extractor, a global statistic pooling layer, and two fully connected (FC) layers to extract the 128-dimensional speaker embedding and classify speakers in the training set. Besides, we apply the Squeeze-and-Excitation [9] operation to explicitly model channel interdependencies. We train the network with additive angular margin (AAM) loss [25]: the re-scaling factor s is set to 32 and angular margin m is set to 0.2.

3.2. ECAPA-TDNN

We also adopt the same ECAPA-TDNN model as proposed in [10] with 1024 channels in the convolutional frame layers. The dimension of the speaker embedding is set to 128. AAM loss is used to train the network with a re-scaling factor s of 30 and angular margin m of 0.2.

3.3. Fine-tuning

The neural network-based system requires a large training dataset to obtain good model generalizability. The FS3 training data with only 218 speakers is not able to learn discriminant speaker representations. To improve model capacities and learn good speaker representations, we pre-train the systems on Voxceleb [26, 27] with 7323 speakers and then fine-tuned on the FS3 SID training data. During fine-tuning, we replace the final FC layer to classify 218 speakers in FS3 SID data. We firstly freeze other model parameters to solely train this FC layer until convergence. In the remaining training epoch, all parameters of the network are jointly optimized.

4. Speaker Diarization

4.1. Preprocessing

For SAD, we use the same method as mentioned in Section 2. The SAD model is trained from the data in the SD task.

For SID, we use the same ResNet-SE system from Section 3.1. We split the SD training and dev data into small segments that only contain a single speaker and remove those segments shorter than 2.0 seconds. Then we fine-tune the model on this split training set and evaluate it on the split dev set.

For segmentation, we perform uniform segmentation for all speech regions to generate the speaker-homogeneous segments. The most talkative speaker in each segment is assigned as the label.

After uniform segmentation, we try to merge two consecutive segments pair to a longer segment if the cosine similarity of these two segment embeddings is greater than a predefined **stop threshold**. Then, for those merged segments, we calculate the new speaker embedding by taking the mean of these merged segments embedding and merge the consecutive segments iteratively until the cosine similarity of any two consecutive segments is lower than the pre-defined stop threshold.

4.2. LSTM/Attention-based Similarity Measurement and Spectral Clustering

We only use uniform segmentation in this system. We employ an LSTM-based and an attention-based neural network to measure the similarity between two segments, respectively. For LSTM-based similarity measurement, the architecture and network configuration are the same as the LSTM model in [14]. For attention-based similarity measurement, the network architecture and training process is the same as the attentive vector-to-sequence (Att-v2s) scoring in [15]. In addition, we employ the augmentation method [28] on the training embedding, which can rotate the l2-normalized embedding by multiplying an orthonormal matrix. Finally, after we get the affinity matrix, we can employ the spectral clustering [29] to obtain the diarization results. Details can be found in [14, 15].

4.3. AHC-based Clustering

For agglomerative hierarchical clustering (AHC), we use cosine distance to measure the similarity between the embedding of two segments. Then we perform AHC on the similarity matrix with a relatively high **stop threshold** and get several clusters. The number of these clusters is greater than the number of speakers in most cases. Next, we split these clusters into 'long clusters' and 'short clusters' by setting a **duration threshold**,

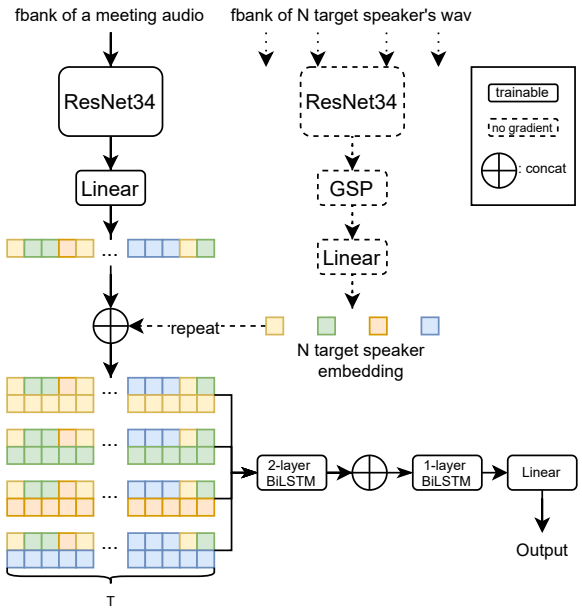

Figure 2: *The architecture of the TS-VAD model*

and then compute the mean of the embedding in each cluster as the center embedding, so that we can assign these short clusters to long clusters. Finally, if a short cluster is too different from all long clusters, which means that the distance between them is lower than a **speaker threshold**, we treat it as a new speaker. We employ grid search to find the best thresholds to obtain the lowest DER.

4.4. Target-speaker Voice Activity Detection

After we obtain the first round of diarization results, we try to perform target-speaker voice activity detection (TS-VAD) to further enhance the performance. The TS-VAD is first proposed in [16], and has shown a superior performance on the DIHARD III challenge [17, 18]. Unlike previous methods which use i-vector as speaker embedding, we employ deep ResNet vector as speaker embedding for TS-VAD. Previous work [16] show that x-vector works poorly in the TS-VAD task, but we find that if use a more complex front-end network to extract the frame-level speaker information, it also achieves a good performance. Figure 2 shows the architecture of our TS-VAD model. The GSP denotes the global statistical pooling layer.

4.5. System Fusion

We employ DOVER-Lap [20] as our fusion strategy, which further enhances the performance.

5. Experimental Results

5.1. SAD

5.1.1. Experimental Setup

The SAD model is trained on the FS3 training set. The input feature is the magnitude spectrum from a Short Time Fourier Transform (STFT), the window length is 25 ms and the frameshift is 10 ms, the hamming window is applied to each frame, and the FFT size is 512, therefore the dimension of the input feature of each frame is 257. Features are split into chunks for training, each chunk contains 400 frames, and the chunk shift is 200 frames.

Before feeding the feature into the model to train, the SpecAugment is applied. The maximum width of the frequency mask is 10, and the number of the mask is 2.

We train the model for 50 epochs with a batch size of 100. The cross-entropy is adopted to calculate the loss between the estimated probability and the ground-truth one-hot label. The Adam [30] optimizer is used with a learning rate of 0.001. The dev set is used for validation. The learning rate is decreased by a factor of 0.1 when the validation loss doesn't decrease for 3 epoch and the training will be terminated when the validation loss doesn't decrease for 10 epochs. The model with the lowest validation loss is selected as the best model and evaluated on the eval Set.

5.1.2. Results

Table 1 shows the results of the SAD task. The Detection Cost Function (DCF) on the Eval Set of U-Net is 2.243%, with the SpecAugment, the DCF reduces to 2.062%, and after HMM smoothing, the DCF is further reduced to 1.915%. Our best result ranks 2nd among all submissions. It takes 6.2 s for our system to process 30-minute audio with one GPU telsa m40, the Real-Time Factor (RTF) is 0.0034.

Table 1: *DCF of the models in SAD task*

System	Dev(%)	Eval(%)
U-Net	1.106	2.243
U-Net+SpecAugment	1.217	2.062
U-Net+SpecAugment+HMM	**0.937**	**1.915**
Baseline	-	15.610

5.2. SID

5.2.1. Experimental Setup

The pre-training data includes VoxCeleb 1 [26] and VoxCeleb 2 [27] with 1,276,888 utterances from 7,323 speakers. The original data with a sampling rate of 16,000 are downsampled to 8,000 to match the FS3 dataset. The FS3 speaker identification dataset includes 218 speakers with 27,336, 6,373, and 1,4077 utterances in the training, development, and evaluation set respectively.

We perform on-the-fly data augmentation [31] with MUSAN dataset [32]. The additive noise includes ambient noise, music, and babble noise. The babble noise is constructed by mixing three to eight speech files into one. For the convolutional noise, we use 40,000 simulated room impulse responses (RIR) from small and medium rooms in MUSAN.

For feature extraction, an 80-dimensional log Mel-spectrogram with a 25ms Hamming window and 10ms shifts is used. The duration between 2 to 4 seconds is randomly generated for each data batch during training.

For both ResNet-SE and ECAPA-TDNN systems, network parameters are updated using stochastic gradient descent (SGD) algorithm [33] with a momentum of 0.95. The learning rate is initially set to 0.1 and is divided by 10 whenever the training loss reaches a plateau.

5.2.2. Results

Table 2 shows the results of SID task. Top-K accuracy is used as the evaluation metric. The ResNet-SE and ECAPA-TDNN systems achieve similar performance. The equally weighted score-level fusion of the two systems further improves the identification performance. We obtain top-3 accuracy of 94.35% and 83.27% on the development and evaluation set respectively.

Table 2: *Top-K Accuracy [%] of SID task*

	Top 1	Top 2	Top 3	Top 4	Top 5
	Development Set				
ResNet-SE	81.50	90.54	92.67	93.93	94.55
ECAPA-TDNN	81.26	89.82	92.44	93.50	94.43
Fusion	81.81	91.97	94.35	95.26	95.94
	Evaluation Set				
Fusion	71.93	80.44	83.27	85.00	86.21

5.3. SD

5.3.1. Experimental Setup for SID and SAD model

The SAD model is based on U-Net and the SID model is the ResNet-SE-based system. The SAD model is trained on the SD training set. The SID model is trained on the Voxceleb 1&2 and finetuned on the SD training set.

5.3.2. Experimental Setup for LSTM/Att-v2s Scoring

For LSTM- and attention-based model, we use AMI [34], ICSI [35], voxconverse dev [36], ISL (LDC2004S05), NIST (LDC2004S09) and SPINE1&2 (LDC2000S87, LDC2000S96, LDC2001S04, LDC2001S06, LDC2001S08) for pre-training. The FS3 training set is used for finetuning.

We only use uniform segmentation with a length of 1.5s and a shift of 0.75s, and the training process is the same as [14, 15].

5.3.3. Experimental Setup for AHC

For AHC-based clustering, we use both uniform and AHC-based segmentation. We perform uniform segmentation for all speech region with length of $\{$1.5s, 1.0s, 0.5s$\}$ and shift of $\{$0.75s, 0.50s, 0.25s$\}$, where length is greater than shift. For AHC-based segmentation, the stop threshold is 0.4. For AHC-based clustering, the stop threshold is 0.4, the duration threshold is 5s, and the speaker threshold is 0.2. All of these parameters are tuned on the dev set.

5.3.4. Experimental Setup for TS-VAD

The training data of TS-VAD is simulated from SRE-databases including SRE 2004, 2005, 2006, 2008, and Switchboard. We first perform VAD on these datasets and extract the segments longer than 2 seconds. Then, we simulated about 1,600 hours of data using these segments, and each recording contains 4$\sim$8 speakers. After that, we simulated another 200 hours dataset on the FS3 training set as the finetuning set using the same simulation strategy. Finally, the model is finetuned again on the real FS3 training set. We employ the same data augmentation strategy with MUSAN corpus [32] as mentioned in the SID task.

In the experiments of TS-VAD, the number of target speaker N is set to 8. The parameters of the ResNet34 are initialized from the front-end of our pre-trained SD model. The acoustic features are 80-dimensional log Mel-filterbank energies with a frame length of 25ms and a shift of 10ms. The duration of signal in each batch is 32s, and the speaker embedding is extracted from 4s segments. We only train the model on the speech signal. The outputs are several posteriors that represent the presence probability of a speaker. The learning rate is set to 0.0001 during the training stage and 0.00001 during two finetuning stages with Adam [30] optimizer.

During the inference stage, we select the single-speaker segments from previous round of clustering-based results and extract embedding for each speaker. For each recording, if the number of speakers is less than 8, we randomly generate several vectors as the fake embeddings. If the number of speakers is greater than 8, we split the speakers into several groups that contain 8 speakers and obtain output for each speaker. The chunk size of each wav is set to 32s. Since the FS3 dataset does not contain many overlapped regions, we select the speaker with the largest posterior probability as the target speaker.

Table 3: *The DER (%) of different SD systems.*

Model	**Dev**		**Eval**	
	Track 1	**Track 2**	**Track 1**	**Track 2**
1 LSTM	21.48	13.56	-	-
2 Att-v2s	22.57	15.11	-	-
3 AHC (uni-seg)	20.83	13.33	-	-
4 AHC (ahc-seg)	21.39	14.21	-	-
5 TSVAD (round 0)	20.75	11.88	43.99	13.85
6 TSVAD (round 1)	20.94	11.99	-	-
Fusion (1+2+3+4)	20.39	12.70	44.56	14.63
Fusion (1+2+3+4+5)	-	11.81	-	12.83
Fusion (3+4+5)	**19.19**	**11.40**	**42.21**	**12.32**

5.3.5. Results

Table 3 shows the results of our speaker diarization system. The TS-VAD system shows the best performance in all tracks. After fusing, the performance is further improved, and our final submission is the fusion of systems 3, 4, and 5.

The results of track 1 show a big gap between the dev and eval set. The reason is that the SAD doesn't work well for some recordings in the eval set, which results in a DER of over 300% in some recordings.

6. Conclusion

In this paper, we present the SAD, SID, and SD systems for the FS3 competition. For all tasks, the main challenge is the low signal-to-noise ratio in some recordings. In addition, the domain mismatch of speaker embedding can significantly influence the performance of SD systems. The low SNR can be solved by data augmentation or fusing the results to get a robust system, and the domain mismatch can be addressed by finetuning the SID model on the SD training set. Besides, the ResNet-based TS-VAD also demonstrates a superior performance with the Deep ResNet-based speaker embedding.

7. References

[1] J. H. Hansen, A. Sangwan, A. Joglekar, A. E. Bulut, L. Kaushik, and C. Yu, "Fearless steps: Apollo-11 corpus advancements for speech technologies from earth to the moon," in *INTERSPEECH*, 2018.

[2] J. H. Hansen, A. Joglekar, M. C. Shekhar, V. Kothapally, C. Yu, L. Kaushik, and A. Sangwan, "The 2019 Inaugural Fearless Steps Challenge: A Giant Leap for Naturalistic Audio," in *INTERSPEECH*, 2019.

[3] A. Joglekar, J. H. Hansen, M. C. Shekar, and A. Sangwan, "Fearless steps challenge (fs-2): Supervised learning with massive naturalistic apollo data," *INTERSPEECH*, 2020.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[5] 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 *INTERSPEECH*, 2019.

[6] 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*, vol. 19, no. 4, pp. 788–798, 2010.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*, 2018.

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

[9] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," in *CVPR*, 2018.

[10] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *INTERSPEECH*, 2020.

[11] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[12] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[13] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *arXiv preprint arXiv:2101.09624*, 2021.

[14] Q. Lin, R. Yin, M. Li, H. Bredin, and C. Barras, "Lstm based similarity measurement with spectral clustering for speaker diarization," in *INTERSPEECH*, 2019.

[15] Q. Lin, Y. Hou, and M. Li, "Self-attentive similarity measurement strategies in speaker diarization," in *INTERSPEECH*, 2020.

[16] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny *et al.*, "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," in *INTERSPEECH*, 2020.

[17] Y. Wang, M. He, S. Niu, L. Sun, T. Gao, X. Fang, J. Pan, J. Du, and C.-H. Lee, "Ustc-nelslip system description for dihard-iii challenge," *arXiv preprint arXiv:2103.10661*, 2021.

[18] W. Wang, Q. Lin, D. Cai, L. Yang, and M. Li, "The dku-dukelenovo system description for the third dihard speech diarization challenge," *arXiv preprint arXiv:2102.03649*, 2021.

[19] A. Stolcke and T. Yoshioka, "Dover: A method for combining diarization outputs," in *ASRU*, 2019.

[20] D. Raj, L. P. Garcia-Perera, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, and S. Khudanpur, "Dover-lap: A method for combining overlap-aware diarization outputs," in *SLT*, 2021.

[21] G. W. Lee and H. K. Kim, "Multi-task learning u-net for single-channel speech enhancement and mask-based voice activity detection," *Applied Sciences*, vol. 10, no. 9, p. 3230, 2020.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[23] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[24] D. Cai, W. Cai, and M. Li, "Within-Sample Variability-Invariant Loss for Robust Speaker Recognition Under Noisy Environments," in *ICASSP*, 2020, pp. 6469–6473.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *CVPR*, 2019.

[26] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A Large-Scale Speaker Identification Dataset," in *INTERSPEECH*, 2017.

[27] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep Speaker Recognition," in *INTERSPEECH*, 2018.

[28] Q. Li, F. L. Kreyssig, C. Zhang, and P. C. Woodland, "Discriminative neural clustering for speaker diarisation," in *SLT*, 2021.

[29] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[31] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1038–1051, 2020.

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

[33] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *ICML*, 2013.

[34] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The ami meeting corpus: A pre-announcement," in *MLMI*, 2005.

[35] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *ICASSP*, 2003.

[36] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: speaker diarisation in the wild," in *INTERSPEECH*, 2020.