ROBOVOX: Far-field Speaker Recognition by a Mobile Robot

Team FFTastic (ID: 27984)

***Abstract*—This challenge focuses on addressing the complex task of far-field-single channel speaker recognition using speech signals recorded by a mobile robot amidst various far-field challenging scenarios including ambient noise, internal robot noise and reverberation. The enrollment data are collected from a microphone that is closer to the speaker and the test data is collected from a microphone embedded within the robot. To confront this challenge, we employed a combination of trained and fine-tuned TitaNet-L models, along with various signal enhancement and denoising techniques. Our proposed approach resulted in achieving a DCF value of 0.4401 and an EER of 7.4139.**

I. Introduction

THE IEEE Signal Processing Cup 2024 RoboVox challenge focuses on text-independent far-field speaker recognition specifically on mobile robots, amidst the effects of noise and reverberation. The task requires designing of robust speaker recognition methodologies for mobile robotic environments using a set of far-field single-channel tracks. The dataset, from the ANR project RoboVox [1], features speakers at different distances, environments, and settings. This dataset enables assessing performance degradation in far-field microphones and explores speaker verification challenges in real-world scenarios.

This report consists of a comprehensive analysis of the competition dataset and the task, an explanation of the implemented methodology, and a discussion of the results obtained.

II. Pre-trained Speaker Recognition Model Selection

Speaker recognition has been a widely discussed area in research. As such, many models are available for this task that have been pre-trained with different datasets. The models used for this competition are *d-vector system* [2], *Deep Speaker model* [3] and *TitaNet-L model* [4].

After adopting these models in the RoboVox challenge, we obtained baseline scores using the given enrollment and test data as outlined in Table I.

Model	EER	DCF
d-vector system [2]	27.3713	0.9832
Deep Speaker model [3]	26.6841	0.9375
TitaNet-L model [4]	14.6922	0.8573

TABLE I: Baseline scores for the selected models. DCF and EER are the evaluation metrics.

Comparing the baseline scores for the selected models, TitaNet-L model outperforms the other two. Hence, we chose to advance with the pre-trained TitaNet-L model for further enhancements through the process of fine-tuning.

A. TitaNet-L Model

TitaNet-L [4] is a neural network architecture designed to extract speaker representations from audio data. It employs 1D depth-wise separable convolutions coupled with Squeeze-and-Excitation (SE) layers to capture both local features and global context from speech utterances. The model maps variable-length utterances to fixed-length embeddings called *t-vectors* which represent the identity of individual speakers.

The pre-trained model is trained for 250 epochs using SGD optimizer with a cosine annealing learning rate scheduler on multiple GPUs. We have outlined the various datasets used for training the model in Table II.

Dataset	# of Speakers	Duration (hrs)	# Utterances (K)
VoxCeleb1 [5]	1211	227	332
VoxCeleb2 [5]	5994	1895	2274
SRE [6]	3787	503	944
Fisher [7]	951	162	278
Switchboard [6]	2400	247	425
LibriSpeech [8]	2338	336	634
Total	16681	3373	4890

TABLE II: Statistics of each dataset used for training TitaNet-L model.

The combined datasets encompassed utterances from 16.6K distinct speakers, rendering TitaNet-L a robust and powerful model. Fig. 1 depicts the model architecture. We provide a concise overview of the essential components of the model as follows.

1) Encoder: Built upon the ContextNet ASR architecture [9], the model features an encoder-decoder structure. The encoder comprises prologue and epilogue blocks, mega blocks, and a sequence of depth-wise separable convolutional layers with SE layers.

2) Decoder and Embeddings: The high-level acoustic features extracted by the encoder undergo attentive statistics pooling layers for computing intermediate features. Subsequently, these features are input into the decoder to produce utterance-level speaker embeddings.

3) Loss Function: The model is trained end-to-end using additive angular margin (AAM) loss $\mathcal{L}$ given in (1) where m is the margin, s is the scale and θ_j is the angle between the final linear layer weight W_j and incoming feature x_i. Here m and s are predefined hyperparameters.

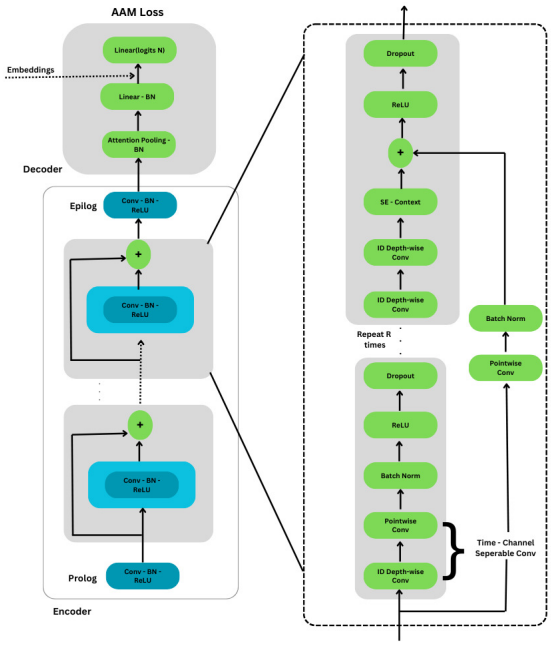


Fig. 1: The TitaNet-L model architecture [4].

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{yi}+m))}}{e^{s(\cos(\theta_{yi}+m))}+\sum_{j=1,j\neq yi}^{n}e^{s\cos\theta_j}} \quad (1)$$

This loss function helps to optimise the cosine distance between speaker embeddings.

Using the pre-trained TitaNet-L model, we obtained an EER of 14.6922 and DCF of 0.8573 as mentioned in Table I. The subsequent section outlines the fine-tuning process undertaken to enhance these outcomes.

III. Preprocessing Procedures

A. Enrollment Dataset

The competition's enrollment dataset contains recordings in French involving 75 speakers, comprising 225 recorded dialogues. However, during the final enrollments in the model, the complete enrollment dataset was not utilized. Upon careful examination of the enrollment dataset, it was identified that certain speaker samples were incomplete, leading to inaccuracies in the average embeddings calculated. To that end, the samples from speakers 6[1], 21, 40[2], and 71[3] were excluded. The samples from speakers 6 and 71 solely consist of noise, while the sample from speaker 40 predominantly includes the voice of the robot. Speaker 21 has all samples with no utterances. After eliminating all three samples from speaker 21, their cosine distances were replaced with zeros during the testing phase.

[1] `spk_6-6_11_0_0_d4_ch5`
[2] `spk_40-40_17_0_0_d1_ch5`
[3] `spk_71-71_10_0_1_d5_ch5`

B. Enrollment Dataset Preparation

1) Augmentation: To augment the number of samples in the enrollment dataset, various effects were introduced to the data. These variations encompassed adding noise, introducing babble, incorporating reverberation, and adjusting volume. These effects were applied to the enrollment dataset in diverse combinations.

- Adding noise: Two output files with additive white Gaussian noise were generated for each enrollment sample, featuring a signal-to-noise ratio (SNR) variation within the range of 5-15. We used `audioDataAugmenter()` function in MATLAB.
- Incorporating reverberation: Reverberation was introduced to the enrollment samples utilizing the `reverberator()` function in MATLAB. In this process, the *WetDryMix* was adjusted to 0.9, and the *SampleRate* was set to match that of the audio signal. All other parameters were retained at their default values.
- Introducing babble: Babble noise was generated by combining 10 random enrollment samples to create a unified babble clip. Employing this approach, babble clips were produced for each enrollment audio clip.
- Adjusting volume: The volume adjustment was carried out using the `audioDataAugmenter()` function in MATLAB. Two output files were generated for each enrollment sample, with the volume gain range varying from -20 dB to 20 dB.

2) Signal Enhancement: The initial method for signal enhancement involved filtering the dataset with a low-pass filter set at 5 kHz and a high-pass filter at 100 Hz. Both filters employed were FIR filters with a passband of 0 dB and a stopband attenuation of -80 dB.

Since the noise level was still high even after filtering, we used wavelet denoising on the enrollment dataset instead of filtering. We used Symlets wavelet of order 4 (sym4) with empirical Bayesian method with a Cauchy prior in MATLAB. By default, the sym4 wavelet is used with a posterior median threshold rule.

The silent intervals within the data samples contained distinct noise. Upon their successful removal, we got a more refined segment for further processing. Consequently, to differentiate active speech segments from silent periods, voice activity detection (VAD) was implemented for each enrollment clip. To that end, we evaluated both statistical and deep learning-based VAD methods and opted for the following statistical-based approach which uses a thresholding algorithm based on energy and spectral spread per analysis frame.

The algorithm detects the active voice regions of a given audio signal $x(n)$ by measuring the short-term energy (STE)

$$E(m) = \sum_{n=(m-1)N+1}^{mN} |x(n)|^2 \quad (2)$$

for frame m with length N and comparing it with a adjustable threshold T. If $E(m) > T$, frame m is categorized as speech; otherwise, it is classified as silence. The binary sequence

generated by thresholding is commonly referred to as the VAD mask, represented by

$$F(m) = \begin{cases} 1, & \text{if } E(m) > T, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Following the VAD process, the signals underwent amplitude normalization to ensure that the amplitudes of the audio signals fell within the range of -1 to 1.

A graphical illustration of the VAD process is given in Fig. 2.

C. Test Dataset Preparation

Due to the substantial noise in the test dataset, we employed the spectral subtraction method [10] just before wavelet denoising. This method involves subtracting the noise spectrum from the signal spectrum to derive a cleaner signal spectrum. As it requires a silent period at the beginning of the signals, a noise sample retrieved from the samples folder in the dataset was prepended to the dataset.

To this end, we used the Boll spectral subtraction method in MATLAB using `SSBoll79` function [4] [11]. The SSBoll79 method utilizes a sequence of steps to improve audio signals that are heavily contaminated with noise. Fig. 3(a) and 3(b) depict an audio signal before and after undergoing the spectral subtraction process.

Subsequently, VAD is employed to identify the segments of a given signal that contain speech. The segments identified in this manner were isolated from the original test dataset, creating a new dataset with the removal of silent sections. This newly formed dataset was then utilized for testing.

D. Sample File Utilization

The sample file contained audio dialogues between the speaker and the robot across 8 channels, with channel 5 experiencing the least issues and channel 4 encountering the most issues. Subsequently, channel 5 clips were utilized for model fine-tuning, while channel 4 clips served as a reference noise sample for the `SSBoll79` function [11].

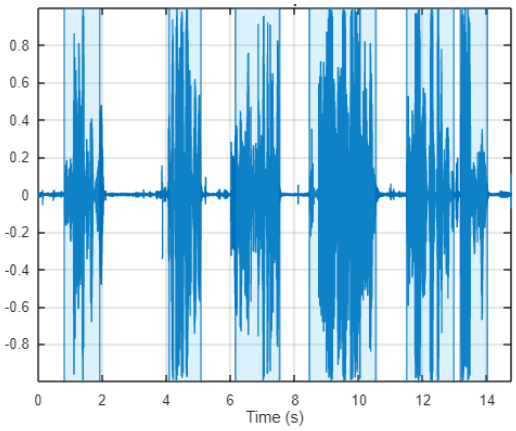
IV. Fine-tuning Procedures

In this section, we present three approaches used to fine-tune the pre-trained TitaNet-L model according to the competition dataset. For the fine-tuning process, we used the enrollment dataset described in Section III-A. For the convenience of referring, we label this dataset as *fine-tune dataset*. The encoder and decoder were configured with learning rates of 0.0005 and 0.001, respectively, for the entirety of the fine-tuning process.

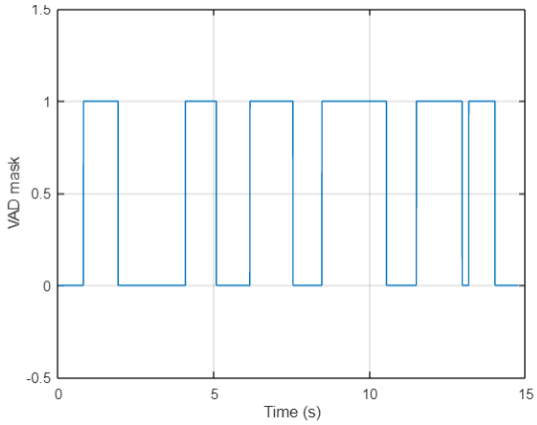
A. Approach 1

We first processed our *fine-tune dataset* with the VAD algorithm to filter out the silent periods. Then the resulting speaker recordings are augmented using the techniques presented in III-B1. The TitaNet-L model is then fine-tuned for 20 epochs using the augmented dataset.

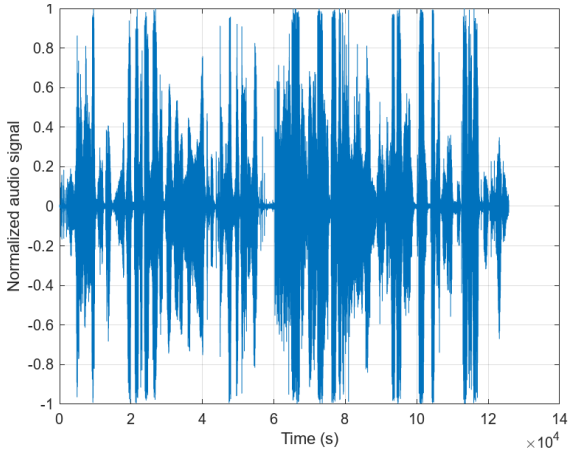
[4] After numerous trials, the threshold was determined to be 0.01 seconds.



(a) Detecting speech segments.

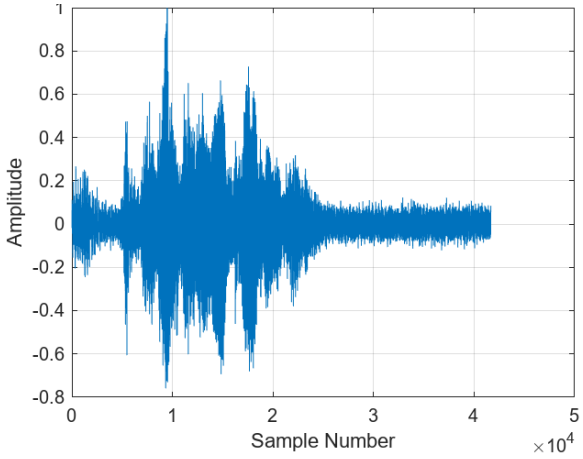


(b) VAD mask.

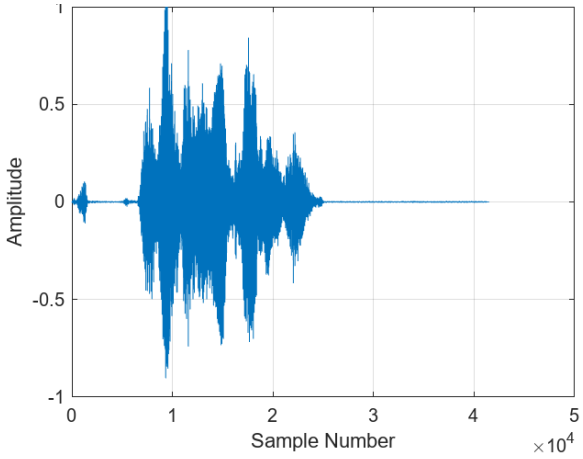


(c) Signal after the VAD process.

Fig. 2: VAD process.



(a) Original normalized test audio.



(b) Audio after spectral subtraction.

Fig. 3: Spectral subtraction process.

B. Approach 2

We first processed our *fine-tune dataset* with the VAD algorithm to filter out the silent periods. Then we performed normalization to obtain sample values between -1 and 1. The model is then fine-tuned for 100 epochs using the resulting dataset without any augmentation.

V. Results

After extensive experimentation, we observed improved performance with fine-tuned models specifically in trials featuring speakers 8, 9, 71, 72, 73, 74, 75, 76, and 77. We call this group of trials as the *special trials* for ease of reference. Consequently, we applied the fine-tuned models exclusively to these trials, which happen to constitute approximately 90% of the trials in the given trial set. For the remaining trials, we employed the original pre-trained TitaNet-L model without any fine-tuning.

We used the following two methods to generate the results.

- Method 1: The TitaNet-L model, subjected to the fine-tuning process outlined in Section IV-A, has been applied to the *special trials*. Conversely, for the remaining trials, the original pre-trained TitaNet-L model without any fine-tuning has been employed.
- Method 2: The TitaNet-L model, subjected to the fine-tuning process outlined in Section IV-B, has been applied to the *special trials*. Conversely, for the remaining trials, the original pre-trained TitaNet-L model without any fine-tuning has been employed.

Out of the two methods, Method 2 yielded the most favourable results.

Table III summarizes the results obtained from the methods described above. The baseline results refer to the outcomes obtained only using the pre-trained TitaNet-L model without any fine-tuning for all the trials.

Method	EER	DCF
Baseline	14.6922	0.8573
Method 1	11.6725	0.6627
Method 2	7.4139	0.4401

TABLE III: Summary of performance

It is worth emphasizing that substantial improvements in DCF and EER performance metrics can be achieved by engaging in fine-tuning and employing a blend of both fine-tuned and the original pre-trained TitaNet-L models.

VI. Conclusion

This technical report outlines our approach to the speaker verification task using the RoboVox single-channel dataset. We utilized both a pre-trained TitaNet-L model and its fine-tuned counterpart for speaker verification. With the approach detailed in this report, we attained a DCF value of 0.4401 and an EER of 7.4139 in the testing trials set.

Appendix A
CodaBench User Names

The following are the usernames of the team members who submitted the results to the CodaBench platform.

- omega
- hawki
- sandushan

References

[1] M. Mohammadamini, M. Rouvier, D. Matrouf, *et al.*, "RoboVox: Far-field speaker recognition by a mobile robot," in *Proc. IEEE Signal Processing Cup*, 2024.

[2] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, *Generalized end-to-end loss for speaker verification*, 2020. arXiv: 1710.10467 [eess.AS].

[3] C. Li, X. Ma, B. Jiang, *et al.*, *Deep speaker: An end-to-end neural speaker embedding system*, 2017. arXiv: 1705.02304 [cs.CL].

[4] N. R. Koluguri, T. Park, and B. Ginsburg, "Titanet: Neural Model for Speaker Representation With 1D Depth-Wise Separable Convolutions and Global Context," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2022-May, pp. 8102–8106, 2022. arXiv: 2110.04410.

[5] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," Jun. 2018.

[6] J. Godfrey and E. Holliman, *Switchboard-1 release 2 ldc97s62*, Linguistic Data Consortium, 1993.

[7] C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker, *Fisher english training speech part 1 transcripts*, Linguistic Data Consortium, 2004.

[8] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *ICASSP*, 2015.

[9] W. Han, Z. Zhang, Y. Zhang, *et al.*, "ContextNet: Improving convolutional neural networks for automatic speech recognition with global context," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2020-October, no. 1, pp. 3610–3614, 2020. arXiv: 2005.03191.

[10] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[11] E. Zavarehei, *Boll spectral subtraction*, https : / / www . mathworks . com / matlabcentral / fileexchange / 7675 - boll - spectral-subtraction, Retrieved February 3, 2024, 2024.