Pdl team SVDD submission

Hemlata Tak, Ricardo Casal, Ganesh Sivaraman, Elie Khoury

Pindrop, Atlanta, GA, USA

{Hemlata.Tak,rcasal,gsivaraman,ekhoury}@pindrop.com

Abstract

In SVDD challenge our team focused on CtrSVDD track task. Our goal to build the most robust ensemble of different fake singing detection systems by minimizing the EER on SVDD dev set. This report describes the details of our challenge submission. In particular, we present the results of each individual systems on SVDD dev set and a final fusion results on SVDD dev and evaluation set. Our best fusion achieves 0.41% on SVDD development set and 3.4% EER on SVDD evaluation set.

Index Terms: singing deepfake detection

1. Introduction

Singing voice deepfake detection (SVDD) challenge aim is to promote the development of robust fake singing detector. To highlight the importance of improving the detection of fake singing voices, Zang et al. [1] proposed the task of singing voice deepfake (SingFake) detection. Singing voice deepfake detection presents unique challenges not found in speech deepfake detection. Unlike speech, singing involves adherence to melodies and rhythms that alter the pitch and duration of phonemes. Given these distinct characteristics of singing voices, the detection system designed for speech are not directly well suited for detecting fake singing voices. How to design a robust fake singing detection model tailored for the singing voice domain is a emerging topic in research community.

2. System setup

2.1. Front-end features

The front-end features refers to the pre-processing part of the network that converts raw audio samples into acoustic features, which we used in backend classifier to make predictions.

LFCC Linear Frequency Cepstral Coefficients. LFCC features extracted from 25ms frame-blocking with 10ms shift, a filterbank with 20 linearly-scaled filters and 20 static, delta and double-delta features, thereby giving 60-dimensional feature vectors.

LFB Linear filterbank features. LFBs are a direct compressed version of the short-time Fourier transforms (STFT).We extract 40-dimensional linear filter bank (LFB) features extracted from 32 ms window length with a 8 ms shift and a 512-point FFT.

SSL We used wav2vec2.0 self-supervised based front-end features. We have explored the wav2vec 2.0 XLS-R(1B) model [2][1] in our submission. wav2vec 2.0 XLS-R is a large-scale cross-lingually pre-trained model trained on extensive speech datasets.

2.2. Architectures

We used three different architectures. Each described in the following.

x-vector The x-vector [3] deep neural network uses a stack of convolutional layers followed by a temporal pooling layer. In our submission x-vector architecture adopted from speechbrain recipe [2]. A feature extraction layer is first used to decompose raw waveform into LFCC representations as input to the x-vector. We slightly modified the x-vector parameters rather than using default parameters in Speechbrain. We slightly modified a number of output time-delay neural network (tdnn) channels to ([256, 256, 256, 256, 512]) and the number of neurons to 512 in the hidden layer. The final logits output by the network indicates the likelihood that a given song audio is real (bona fide).

ResNetSE34 We used a ResNet34 model descibed in [4] with squeeze-and-excitation (SE) blocks [5] (ResNetSE34). A feature extraction layer is first used to decompose raw waveform into LFB representations. Four convolutional layers with SE blocks are then used to extract compact, deep features. After a flattening operation, attentive statistics pooling (ASP) layer is used to aggregate temporal frames and to project the variable length input to a fixed-length embedding vector. After the ASP layer the channel-wise weighted standard deviation is calculated and concatenated to the weighted mean to extract the 256-dimensional embeddings. Finally, a fully-connected linear layer with two neurons is used to make predictions.

wav2vec2 We used SSL based front-end (wav2vec2.0 XLSR 1B) and simple backend comprises ASP layer and three fully-connected linear layers followed by batch-normalization and SeLU activation function.

3. Experimental setup

3.1. Datasets

CtrSVDD dataset has three independent subsets: train; development; evaluation. Spoofed singing samples in each dataset is generated using a set of different singing voice synthesis (SVS) and singing voice conversion (SVC). Deepfake attacks in the training and development set were created with a set of 8 different generation methods (A01-A08), whereas those in the evalu-

[1] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec/xlsr

[2] https://github.com/speechbrain/speechbrain/blob/develop/speechbrain/lobes/models/Xvector.py

ation set were created with a set of 6 deepfake generation methods (A09-A16). Additional datasets were also used for training as allowed in challenge [6]. Additional singing datasets: Soundboard [3], NUS-48E [7], OpenSinger [8], CSD [9], URSing [4], Acappella [5], FSD [10], Audioset (acapella Synthetic singing categories) [6], JukeBox [11], CN-Celeb [7], Artist20 [8]. Further pre-processing audio samples in described in following section.

3.2. Audio prepossessing

The audio was segmented into 4 second chunks. Audio chunks with less than 2 seconds were filtered out and chunks between 2 to 4 seconds were padded by repeating the signal. A model trained on Audioset [9] was used to generate a singing score for each segment. A threshold, selected empirically, was used to filter out segments that most likely do not contain any vocal information.

3.3. Data-augmentation

We used RawBoost [12] Data-augmentaion tool [10] to add nuisance variability on-the-fly to the training data. RawBoost adds variation in the form of: i) linear and non-linear convolutive noise; ii) impulsive signal-dependent additive noise; iii) stationary signal-independent additive coloured noise. We exactly used the same configuration and parameters reported in the original work [12]. In our submission we used a combination of linear and non-linear convolutive noise and impulsive signal-dependent additive noise for ResNet34 and SSL based systems. Additionally, we also added music noise to the training data.

3.4. Implementation details

All the described systems were trained using Pytorch framework and Adam optimizer. A weighted sampler was used to balance the classes and adjust the importance of certain datasets. We have trained the x-vector model for 50 epoch with a batch size of 64 and a cosine annealing learning rate ranging from 5e-5 to 1e-6. ResNetSE34 model was trained for 30 epochs with learning rate of 1e-3 and, the last SSL-based model was trained for 3 epochs with a batch-size of 20 and a lower learning rate of 1e-6 to avoid model over-fitting. Since SSL pre-training is performed with only real speech data (with no fake samples), as we know singing deepfake detection performance is expected to improve with fine tuning using in-domain real and deepfake singing datasets. Hence, we performed SSL fine-tuning using singing training datasets. Our hypothesis is that fine-tuning on singing datasets will improve generalization. We used the Equal Error Rate (EER) as our evaluation matric which is determined by using a threshold on the produced scores where the false acceptance rates and the false rejection rates are equal.

[3] www.101soundboards.com/boards/tts#song_generators

[4] zenodo.org/records/6404999

[5] labrosa.ee.columbia.edu/projects/artistid/

[6] https://research.google.com/audioset/dataset/synthetic_singing.html

[7] www.cnceleb.org/

[8] labrosa.ee.columbia.edu/projects/artistid/

[9] https://huggingface.co/MIT/ast-finetuned-audioset-10-10-0.4593

[10] github.com/TakHemlata/RawBoost-antispoofing

Table 1: *Results are presented in terms of EER using SVDD development and evaluation set.*

system	Dev EER (%)	Eval EER (%)
ResNetSE34	0.71	-
x-vector	5.62	-
SSL (wav2vec2 XLSR)	2.31	-
Fusion (Random-forest)	0.41	3.44

3.5. Fusion details

A score-level fusion was perform at segment-level. A 3-Fold approach was used for hyperparameter tuning of a RandomForest classifier with restricted estimators, maximum depth, and minimum samples per leaf. The final file-level score was generated using the median of the fused segmental-scores.

4. Results

The testing results on SVDD dev and eval data are presented in the Table 1. These results reflect the capability of our fake singing detection systems used in fusion. From this Table 1 we can see how our single systems also achieved the good performance on SVDD development set and further improved by using a fusion of all three systems. Amongst all the single systems our ResNet34 based system achieves the lowest overall EER on development set.

5. References

[1] Y. Zang, Y. Zhang, M. Heydari, and Z. Duan, "Singfake: Singing voice deepfake detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 156–12 160.

[2] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. INTERSPEECH*, 2022.

[3] 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.

[4] Y. Kwon, H.-S. Heo, B.-J. Lee, and J. S. Chung, "The ins and outs of speaker recognition: lessons from voxsrc 2020," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5809–5813.

[5] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[6] Y. Zhang, Y. Zang, J. Shi, R. Yamamoto, J. Han, Y. Tang, T. Toda, and Z. Duan, "Svdd challenge 2024: A singing voice deepfake detection challenge evaluation plan," *arXiv preprint arXiv:2405.05244*, 2024.

[7] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2013, pp. 1–9.

[8] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao, "Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3945–3954.

[9] S. Choi, W. Kim, S. Park, S. Yong, and J. Nam, "Children's song dataset for singing voice research," in *International Society for Music Information Retrieval Conference (ISMIR)*, vol. 4, 2020.

[10] Y. Xie, J. Zhou, X. Lu, Z. Jiang, Y. Yang, H. Cheng, and L. Ye, "Fsd: An initial chinese dataset for fake song detection," in

ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024, pp. 4605–4609.

[11] A. Chowdhury, A. Cozzo, and A. Ross, "Jukebox: A multilingual singer recognition dataset," *arXiv preprint arXiv:2008.03507*, 2020.

[12] H. Tak, M. Kamble, J. Patino, M. Todisco, and N. Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6382–6386.