A NEURAL NETWORK-BASED HOWLING DETECTION METHOD FOR REAL-TIME COMMUNICATION APPLICATIONS

Zhipeng Chen[1], *Yiya Hao*[1], *Yaobin Chen*[1], *Gong Chen*[1], *Liang Ruan*[2]

[1]NetEase CommsEase AudioLab, Hangzhou, Zhejiang, China
[2]NetEase GrowthEase, Hangzhou, Zhejiang, China
[1]*chenzhipeng02@corp.netease.com* ,[2]*ruanliang@corp.netease.com*

ABSTRACT

Howling arises from acoustic coupling between the speaker and the microphone when it creates positive feedback. Traditional public addressing systems and hearing aids devices detect and suppress the howling using conventional howling features. However, conventional howling features in real-time communication (RTC) suffer from nonlinearities and uncertainties such as various speaker/microphone responses, multiple nonlinear audio processing, unstable network transmission jitter, acoustic path variations, and environmental influences. In howling detection, the signal processing methods using specific temporal-frequency characteristics are ineffective for RTC scenarios. This paper proposes a convolutional recurrent neural network (CRNN) based method for howling detection in RTC applications, achieving excellent accuracy with low false-alarm rates. A howling dataset was collected and labeled for training purposes using different mobile devices, and the log Mel-spectrum is selected as input features. The proposed method achieves an 89.46% detection rate and only a 0.40% false alarm rate. Furthermore, the model size of the proposed method is only 121kB and has been implemented in a mobile device running in real-time.

Index Terms — howling detection, real-time communication (RTC), neural network, Convolutional Recurrent Neural Network (CRNN)

1. INTRODUCTION

Howling is generated in audio amplifier systems such as public addressing systems [1] and hearing aids [2] when acoustic coupling diverges by positive feedback [3]. The howling varies dramatically because of the various environment, including the positions of the microphones and loudspeakers, the movements of talkers, and others [4]. Recently, with the rapid development of real-time communication (RTC) applications, howling has become a severe problem. In RTC scenarios (e.g., online meetings, online education, online diagnosis, Etc.), once two or more devices are closed to others, the howling will occur.

Conventional signal processing methods for howling detection mainly depend on temporal-frequency features [5] like Peak-to-Threshold Power Ratio (PTPR), Peak-to-Average Power Ratio (PAPR), Peak-to-Harmonic Power Ratio (PHPR), Peak-to-Neighboring Power Ratio (PNPR), Interframe Peak Magnitude Persistence (IPMP), and Interframe Magnitude Slope Deviation (IMSD). More advanced methods take advantage of NINOS² [6], spectral flatness measure (SFM) [1], temporal power spectra [4], and voice activity detection (VAD) [2].

However, even several methods such as VAD [2] can slightly reduce the missing rate, the traditional detectors still suffer serious false alarm problems, especially for music signals. Moreover, the conventional signal processing features mentioned above are inadequately compelling in RTC applications. This problem is due to the multiple nonlinear audio processing such as acoustic echo cancellation (AEC), noise suppression (NS), and active gain control (AGC). Furthermore, the unstable network transmission jitter, acoustic path variations, and environment influences decrease the functionality of the conventional features for howling detection. Howling may have much more complex features like discontinuity, diffuseness, multi howling frequencies, and howling frequency shift in this case.

Since neural network models are widely used in acoustic scene classification and sound event detection (such as DCASE challenges [7,8,9]), we propose a convolutional recurrent neural network (CRNN) based method for howling detection in RTC applications. Scholars have summarized deep learning-based methods for acoustic scene classification [10] and sound event detection [11], which show that convolutional neural network (CNN) and CRNN [12] are efficient and primary network architectures for those tasks. Newest tasks focus on limited model size for real-time applications [13,14,15] and weak/no labeled data in real scenarios [16,17,18].

Two significant contributions of this work are (i) the optimized CRNN-based model for howling detection and (ii) the collection and labeling of the howling dataset for training purposes using realistic mobile devices. For the howling detection task, three major problems are taken into account. (i) The requirement of the false alarm is restricted in practical scenarios since the false alarm would frequently interrupt the conversations. (ii) The howling detector needs to be robust and generalized in different hardware since the howling features are device-dependent. (iii) For real-time applications,

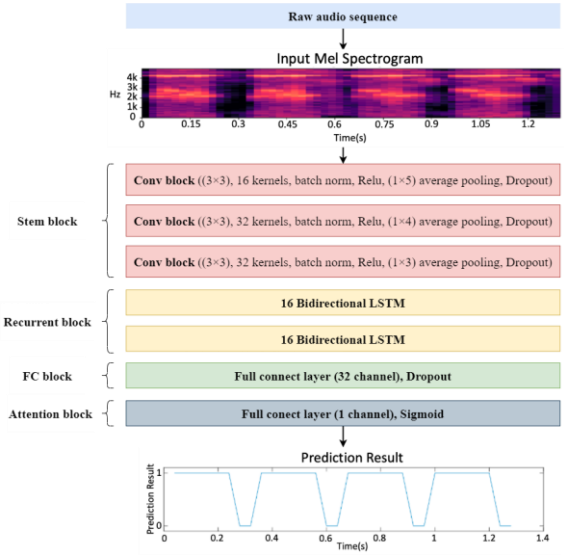

FIG.1. Network diagram of the proposed CRNN-based howling detection model.

Table 1 Network architecture of the proposed CRNN-based howling detection model.

Name	Layers	Output Shape
Input layer	Input: log-mel spectrogram	1×32×60
Stem block	(3×3,Conv2D ,@16,ReLu,BN) 1×5 average pooling layer, dropout	16×32×12
	(3×3,Conv2D ,@32,ReLu,BN) 1×4 average pooling layer, dropout	32×32×3
	(3×3,Conv2D ,@32,ReLu,BN) 1×3 average pooling layer, dropout	32×32×1
Recurrent block	(32 BiLSTM cells)×2	32×32×1
FC block	Linear, @32, dropout	32×32×1
Attention block	Linear, @1	1×32×1

a lightweight model is needed. To solve the problems above, we propose a CRNN-based howling detector with optimized binary cross-entropy loss for an excellent false alarm rate and several regularization modifications for the device-dependent problem. Moreover, a howling dataset is collected and labeled, consisting of 52-hours actual recordings based on different mobile devices. Based on our study, it is the first howling dataset for training purposes. Therefore, we are pushing it to be public and open-source, facilitating the research in this area.

Finally, we evaluate the proposed model using the unseen recordings, which are different from the training data. The comparisons include a conventional method [2], a CNN-based detector [10], and a residual convolutional recurrent neural network (RCRNN) detector [19]. The proposed method achieves an 89.46% detection rate and 0.40% false alarm rate, which outperforms other comparisons. The model size of the proposed method is only 121kB, which satisfies practical RTC applications' requirements.

2. PROPOSED METHOD

2.1 Features

In acoustic scenes classification problems, real-imaginary coefficients in frequency-domain are widely used by scholars [20]. However, we choose Log Mel spectrogram as our input features rather than real-imaginary coefficients. The first reason is that the Log Mel spectrogram contains fewer features, which helps maintain a low computation complexity. Secondly, even real-imaginary coefficients have all signal frequency information, the Log Mel spectrogram picks and emphasizes the critical information while filtering out the others that may confuse the model in the training process. Log Mel spectrogram has proven effective in related tasks [10,21], as it is motivated by human auditory perception and provides a more compact spectral representation of sounds.

2.2 Network architecture

Fig. 1 shows the block diagram of the proposed howling detection framework. The model typically follows CRNN architecture with task-specific modifications. Raw audio signals are resampled uniformly to 16 kHz and segmented into consecutive frames with a 40ms hop size. Then, 60 log Mel-spectrogram coefficients are extracted and normalized, and 32 frames are grouped to make a (32×60) spectral image as model input features. The figure shows that the model consists of four blocks: stem, recurrent, full connect, and attention. The stem block consists of three convolutional blocks with 16, 32, and 32 kernels, respectively. Each convolutional block has (3×3) kernels with a stride of (1×1), and it is followed by batch normalization, Relu activation, and an average pooling layer along the frequency dimension by 5, 4, and 3, respectively. An additional dropout layer is used at the end of each convolutional block to relieve overfitting and device-dependent problems. The recurrent block consists of two Bidirectional long short-term memory (BiLSTM) blocks to learn the temporal context information. The full connect block uses a forward layer as a transition, and a final attention block with sigmoid activation outputs the howling probabilities. A const threshold obtains the final howling detection results in frame-level.

2.3 Model optimizations and modifications

Three major problems are considered in howling detection for RTC applications. First, to reduce the false alarm rate, a weighted binary cross-entropy loss is designed to guide background propagation,

$$L=-\alpha \cdot p \cdot \log q-(2-\alpha)\cdot(1-p)\cdot\log(1-q) \quad (1)$$

where L is the weighted binary cross-entropy loss, p is the target (howling for 1 and no howling for 0), q is the predict results, and α is the weight ($0< \alpha<2$). When $\alpha= 1$, L equals binary cross-entropy. When $\alpha< 1$, false alarm errors contribute more to the loss, which can help our task. α directs the balance of detection and false alarm rates.

Second, regularization methods are applied to relieve device-dependent problems. Dropout layers [22] are applied

to the end of all except the last attention block. Parameter regularization is also used during the training process.

Third, for real-time application, a limited sequence length of 32 frames (1.28s) is chosen instead of 10s audio pieces used in polyphonic sound event detection tasks [16]. Channels and kernels are carefully designed to reduce the model size, and model compression methods like model pruning and quantization are also applied.

Detailed model hyperparameters are listed in Table 1.

3. EXPERIMENTAL SETUP

3.1. Dataset

Considering the particularity of the howling detection task in RTC applications, real-recording data is needed instead of synthesized data in traditional howling detection tasks [23]. However, based on our study, there is no available howling dataset for training purposes. Therefore, a real-recording system is built up. Both howling and other signal data are captured in over 30 mobile devices, with different acoustic structures and audio processing corresponding to the devices. The devices include mobile phones, tablets, and laptops in Android, iOS, Windows, and macOS systems. Background signals include speech, music, environment sound, and noise. Around 52h data are collected, while 48h data are prepared for training and validation, and the rest 4h are for testing. All data are labeled manually and cut into 1.28s audio clips.

3.2. Implementation Details

Data augment methods like time stretch, pitch shift, frameshift, added Gaussian noise, time mask, and frequency mask [24] are applied. However, results trained without data augmentation have a preferable false alarm with a reasonable loss of detection rate. Two models are trained with and without data augmentation as a comparison and the baseline to verify data augment functionality.

We train the models for 500 epochs using Adam optimizer with weight decay to 0.01, mini-batch size to 32, and learning rate starting with 0.02 and decaying by half if the validation loss does not decrease for 10 epochs. Weighted binary cross-entropy loss is used as the loss function, and α is set to 1 by default. A threshold of 0.5 is set to decide active howling in the system outputs.

3.3. Model comparisons

We compare our method with three other methods. We also evaluate some modifications in our network, including the model compression results.

3.3.1 Models comparisons

Three models are applied as comparisons. First, a signal processing method based on the combination of PTPR, PAPR, PNPR, and IPMP features is implemented as a traditional howling detection comparison. We also applied VAD [2] with a 10 milliseconds frame size. Second, a CNN-based method with 3 convolutional layers is implemented as another comparison. Finally, a RCRNN model [19] with a residual convolutional block and CBAM-based attention is evaluated and compared. For neural network methods, all settings are the same as the proposed method, such as epoch number and batch size.

3.3.2 Network modifications

Four tests are measured, including the effect of data augment, the importance of parameter regularization, the use of dropout layers, and the results of different alpha values in the weighted binary cross-entropy loss.

3.3.3 model compression

To compress the proposed model, we apply model pruning and quantization schemes. The pruning is based on a one-shot pruning method based on the L1 filter [25]. Additional fine-tuning is used to enhance the pruned model's performance. We also quantize the pruned model with the quantization-aware training (QAT) [26] method to compress the model size.

3.4. Evaluation metrics

The performance of SED models is usually compared in objective measures, such as F1-score, error rate (ER), and PSDS [27]. A shallow false alarm rate is required for the howling detection task with a relatively high detection rate. False alarm rate also reflects the performance of the device-dependent problem. F1 score is computed as an additional evaluation feature as well. Definitions of TP, FP, FN, and F1 score are as follows:

TP: the label is true, and the corresponding prediction result is also true;

FP: the label is false, but the corresponding prediction result is true;

FN: the label is true, but the corresponding prediction result is false;

F1 score is defined as the harmonic mean of precision (P) and recall (R), which is formulated as

$$F1 = 2PR/(P+R) \tag{2}$$

where $P = TP/(TP+FP)$, $R = TP/(TP+FN)$ (3)

All the evaluation metrics are calculated based on frame-level results.

4. MEASURED RESULTS

This section presents the results for all the experiments described in Section III. All the experiments are trained on NVIDIA TITAN RTX GPUs.

4.1. Evaluation results for model comparisons

Detection rate, false alarm rate, and F1 score results for compared models are presented in Table 2. The detection rate for the signal processing method is relatively low as features for howling in RTC applications are different from

conventional scenarios. Moreover, the false alarm rate is relatively high, especially in music signals, for its limited respective field and broken howling features.

For neural network-based methods, detection rates are improved. For the CNN model, without the help of recurrent block for temporal context information, the false alarm rate is over 5 times than CRNN model. The RCRNN model has a similar performance as the CRNN model, with little degrade in detection rate and increase in false alarm rate. More complicated models which may help in polyphonic sound event detection tasks may not contribute much to the howling detection task. Signals like whistles and birdcalls, which sound like howling and have similar spectrum features, are likely to be falsely detected. For most common speech, music, and environmental sounds, we can successfully avoid detecting them as howling sounds.

4.2. Results for network modifications

Four network modification tests are carried out, and their results are listed in Tables 3 and 4. Table 3 represents the first three experiments' results. Data augmentation is helpful for sound event detection rate significantly with around 3.87% improvement, with the sacrifice of false detections by almost 3.5 times. In our specific howling detection task, data augmentation is not preferable.

Methods for overfitting, like parameter regularization and dropout, are used to relieve device-dependent problems or reduce false alarms in another way. False alarm rate increases by 2.34% and 1.68% for experiments 2 and 3.

Finally, weighted binary cross-entropy loss is evaluated with different alpha values. As shown from Table 4, with the decrease of alpha, the false alarm rate is reduced respectively with the degradation of detection rate. With the increase of alpha, the detection rate is improved with the increase of the false alarm rate. Alpha lower than 1 is preferable in this task. This experiment shows the effectiveness of weighted binary cross-entropy loss to balance the detection and false alarm rates, which may be instructive for related tasks.

4.3 Results for model compression

To compress the proposed model, we utilize two model compression schemes of pruning and quantization. A one-shot pruning method based on L1filter and QAT quantization method are combined and fine-tuned. Finally, an 87.84% detection rate and 0.49% false alarm rate are achieved with the model size compressed from 121kB to 39kB.

Table 2 Detection rate, false alarm rate, and F1 score results for compared models.

Model	Detection rate	False alarm rate	F1 score
Conventional method [2]	17.04%	14.96%	24.69%
CNN model [10]	87.72%	2.29%	91.86%
RCRNN model [19]	88.80%	0.58%	93.66%
CRNN model	**89.46%**	**0.40%**	**94.15%**

Table 3 Detection rate, false alarm rate, and F1 score results for network modifications.

Model	Detection rate	False alarm rate	F1 score
baseline	89.46%	**0.40%**	94.15%
baseline with data augment	93.33%	1.39%	95.57%
baseline without regularization	95.69%	2.74%	95.88%
baseline without dropout layer	96.97%	2.08%	96.06%

Table 4 Detection rate, false alarm rate, and F1 score results for different alpha values in the weighted binary cross-entropy loss.

alpha	Detection rate	False alarm rate	F1 score
1.4	93.08%	1.71%	95.23%
1.2	92.51%	1.37%	95.16%
1.0	**89.46%**	**0.40%**	**94.15%**
0.8	87.72%	0.33%	93.24%
0.6	87.60%	0.38%	93.14%
0.4	87.03%	0.29%	92.88%

5. CONCLUSION

We have proven the proposed method's advantage over conventional signal processing-based methods due to the optimized neural architecture and real-recorded training dataset. For the optimization, firstly, a weighted binary cross-entropy loss is designed to balance the detection and false alarm rates. Second, regularization methods like parameter regularization and dropout layers are applied to relieve device-dependent problems, which also helps reduce the false alarm rate. Finally, pruning and quantization methods are used for real-time applications. Moreover, a howling dataset consisting of about 52h howling data based on multiple devices is created, facilitating the research in this area. Plans to make this howling dataset publicly available are under consideration.

The proposed method is successfully implemented on platforms under different operating systems, including iOS, Android OS, Windows, and macOS. The proposed method runs in real-time with extremely low computation due to its lightweight structure. For every 10 milliseconds of input data at a 16 kHz sampling rate, the largest process time is 62.5 microseconds, measured on Huawei X10 with Android 11 OS.

6. REFERENCES

[1] Jithin, T., KK Mohamed Salih, and A. R. Jayan. "Real Time Suppression of Howling Noise in Public Address System." Procedia Technology 24 (2016): 933-940.

[2] Khoubrouy, Soudeh A., and Issa Panahi. "A method of howling detection in presence of speech signal." Signal Processing 119 (2016): 153-161.

[3] Nyquist, Harry. "Regeneration theory." Bell system technical journal 11.1 (1932): 126-147.

[4] Lee, Jae-Won, and Seung Ho Choi. "Low-complexity howling detection based on statistical analysis of temporal spectra." International Journal of Multimedia and Ubiquitous Engineering 8.5 (2013): 83-92.

[5] Van Waterschoot, Toon, and Marc Moonen. "Fifty years of acoustic feedback control: State of the art and future challenges." Proceedings of the IEEE 99.2 (2010): 288-327.

[6] Mounir Abdelmessih Shehata, Mina. "Acoustic Event Detection: Feature, Evaluation and Dataset Design." (2020).

[7] Mesaros, Annamaria, et al. "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge." IEEE/ACM Transactions on Audio, Speech, and Language Processing 26.2 (2017): 379-393.

[8] Mesaros, Annamaria, Toni Heittola, and Tuomas Virtanen. "Acoustic scene classification: an overview of DCASE 2017 challenge entries." 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC). IEEE, 2018.

[9] Politis, Archontis, et al. "Overview and evaluation of sound event localization and detection in DCASE 2019." IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2020): 684-698.

[10] Abeßer, Jakob. "A review of deep learning based methods for acoustic scene classification." Applied Sciences 10.6 (2020).

[11] Xia, Xianjun, et al. "A survey: neural network-based deep learning for acoustic event detection." Circuits, Systems, and Signal Processing 38.8 (2019): 3433-3453.

[12] Cakır, Emre, et al. "Convolutional recurrent neural networks for polyphonic sound event detection." IEEE/ACM Transactions on Audio, Speech, and Language Processing 25.6 (2017): 1291-1303.

[13] Martín-Morató, Irene, et al. "Low-complexity acoustic scene classification for multi-device audio: analysis of DCASE 2021 Challenge systems." arXiv preprint arXiv:2105.13734 (2021).

[14] Kim, Byeonggeun, et al. QTI submission to DCASE 2021: Residual normalization for device imbalanced acoustic scene classification with efficient design. DCASE2021 Challenge, Tech. Rep, 2021.

[15] Yang, Chao-Han Huck, et al. "A Lottery Ticket Hypothesis Framework for Low-Complexity Device-Robust Neural Acoustic Scene Classification." arXiv preprint arXiv:2107.01461 (2021).

[16] Turpault, Nicolas, et al. "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis." (2019).

[17] Zheng, Xu, Han Chen, and Yan Song. Zheng ustc teams submission for dcase2021 task4 semi-supervised sound event detection. DCASE2021 Challenge, Tech. Rep, 2021.

[18] Kim, Nam Kyun, and Hong Kook Kim. "Self-training with noisy student model and semi-supervised loss function for dcase 2021 challenge task 4." arXiv preprint arXiv:2107.02569 (2021).

[19] Kim, Nam Kyun, and Hong Kook Kim. "Polyphonic Sound Event Detection Based on Residual Convolutional Recurrent Neural Network With Semi-Supervised Loss Function." IEEE Access 9 (2021): 7564-7575.

[20] Küçük, Abdullah, et al. "Real-time convolutional neural network-based speech source localization on smartphone." IEEE Access 7 (2019): 169969-169978.

[21] Hao, Yiya, et al. "A real-time music detection method based on convolutional neural network using Mel-spectrogram and spectral flux." INTER-NOISE and NOISE-CON Congress and Conference Proceedings. Vol. 263. No. 1. Institute of Noise Control Engineering, 2021.

[22] Li, Xiang, et al. "Understanding the disharmony between dropout and batch normalization by variance shift." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[23] Mounir Abdelmessih Shehata, Mina, Giuliano Bernardi, and Toon van Waterschoot. "Howling Corrupted Music and Speech dataset (HCMS)." (2020).

[24] Park, Daniel S., et al. "Specaugment: A simple data augmentation method for automatic speech recognition." arXiv preprint arXiv:1904.08779 (2019).

[25] Li, Hao, et al. "Pruning filters for efficient convnets." arXiv preprint arXiv:1608.08710 (2016).

[26] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[27] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, “A framework for the robust evaluation of sound event detection,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barce-lona, Spain, May 2020, pp. 61–65.