Acoustic Bandwidth Extension by Audio Deep Residual U-Net

Linlin Ou
[1] *Computer Network Information Center*
[1] *Chinese Academy of Sciences*
2 School of computer science and technology
2 University of Chinese Acadamy of Science
Beijing, China
oulinlin@cnic.cn

Yuanping Chen
Computer Network Information Center
Chinese Acadamy of Sciences
Beijing, China
ypchen@cashq.ac.cn

***Abstract*—Mobile communication systems often rely on traditional channels with narrow bandwidth bottlenecks, reducing the transmitted audio quality. Due to the size of the network and the heterogeneity of devices, using high-quality audio codecs in real-world scenarios to transmit higher sample rate audio is difficult to practice. This paper proposes an approach by which communication nodes can extend the bandwidth of the band-limited input signal of a low sample rate codec, an audio synthesis neural network: ARUNet (Audio deep Residual U-Net), that integrates the advantage of residual learning, and U-Net is introduced to compensate for the high-frequency part of the live input audio clips. In addition, ARUNet has made outstanding achievements in various instrument solo data sets, with SNR close to 40dB and VGG distance close to 32. We explore the impact of our architectural design and demonstrate considerable improvements in terms of both perceptual and objective metrics.**

Keywords—Audio Bandwidth Extension, Audio enhancement, Deep Neural Network

I. INTRODUCTION

Audio bandwidth extension (BWE) aims to improve speech fidelity by inferencing the high-resolution (HR) speech given the low-resolution (LR) speech. Low-resolution speech is ordinary for limitations such as low sampling rate and compression. [1] defined speech super-resolution (SR) as BWE. BWE is a critical technique for the real world, which has motivated various literature on speech enhancement.

Inspired by the UNet [2] and ResNet [3] and referred to the generator architecture of MelGAN , we designed an end-to-end audio enhancement model ARUNet. The model is adequately convolutional and maps SR waveforms to HR ones, and raw waveforms are extracted features by multiple encoder layers, which refer to the generator architecture of MelGAN [4]. Then, the extracted features are decoded into the inverse process of extended audio through symmetric architecture.

In training, testing and verification, we also use data augmentation technology based on random low-pass bandwidth filters and apply them to different data sets without repetition to increase the generalization of the model [5], to adapt to data damage in different channels. In addition, we also try different regularization methods, including batch normalization and exit, to verify their promotion on training efficiency.

We propose ARUNet, a lightweight acoustic enhancement method; the processing speed of the waveform is improved as much as possible while ensuring the quality of bandwidth expansion.Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

II. METHODOLOGY

A. *U-style architecture*

The proposed ARUNet is trained in a fully supervised fashion using waveform frame pairs $\{X^{LR}, \bar{X}^{HR}\}$, where $X^{LR} = \{x_n^{LR}\}$ and $\bar{X}^{HR} = \{\bar{x}_n^{HR}\}$ denote low-resolution waveform and corresponding high-resolution ones, respectively. The sampler slices the original audio $X^{LR} = \{x_n^{LR}\}$ by segments $X^{LR} = \{x_m^{LR}\}$ according to the preset sliding window length m as the input of ARUNet. This model does not consider the frequency expansion task of audio, so the number of output audio frame sequence remains unchanged. We let f denote the underlying function modeled by ARUNet:

$$\begin{aligned} f: X^{LR} &= (x_t^{LR}, x_{t+1}^{LR}, \dots, x_{t+m-1}^{LR}, x_{t+m}^{LR}) \\ \mapsto \bar{X}^{HR} &= (\bar{x}_t^{HR}, \bar{x}_{t+1}^{HR}, \dots, \bar{x}_{t+m-1}^{HR}, \bar{x}_{t+m}^{HR}) \end{aligned} \tag{1}$$

t is the earliest timestamp of the frame processed by the model.As illustrated in Fig. 1. ARUNet mainly consist of five encoders and corresponding decoders . We define is all the processing before the residual block (ignoring their internal precedence temporarily) in and , while denotes residual block in them. Subsequently, we define the encoding process as follow:

$$\begin{cases} E_0 = R(F(X^{LR})) + F(X^{LR}) \\ E_k = R(F(E_{k-1})) + F(E_{k-1}), k \in [1,4] \end{cases} \tag{2}$$

The bottleneck module plays the role of connecting the encoder and decoder. However, its internal block arrangement is similar to the decoder, because the first pixel shuffle upsampling [30] block is introduced into this module to keep

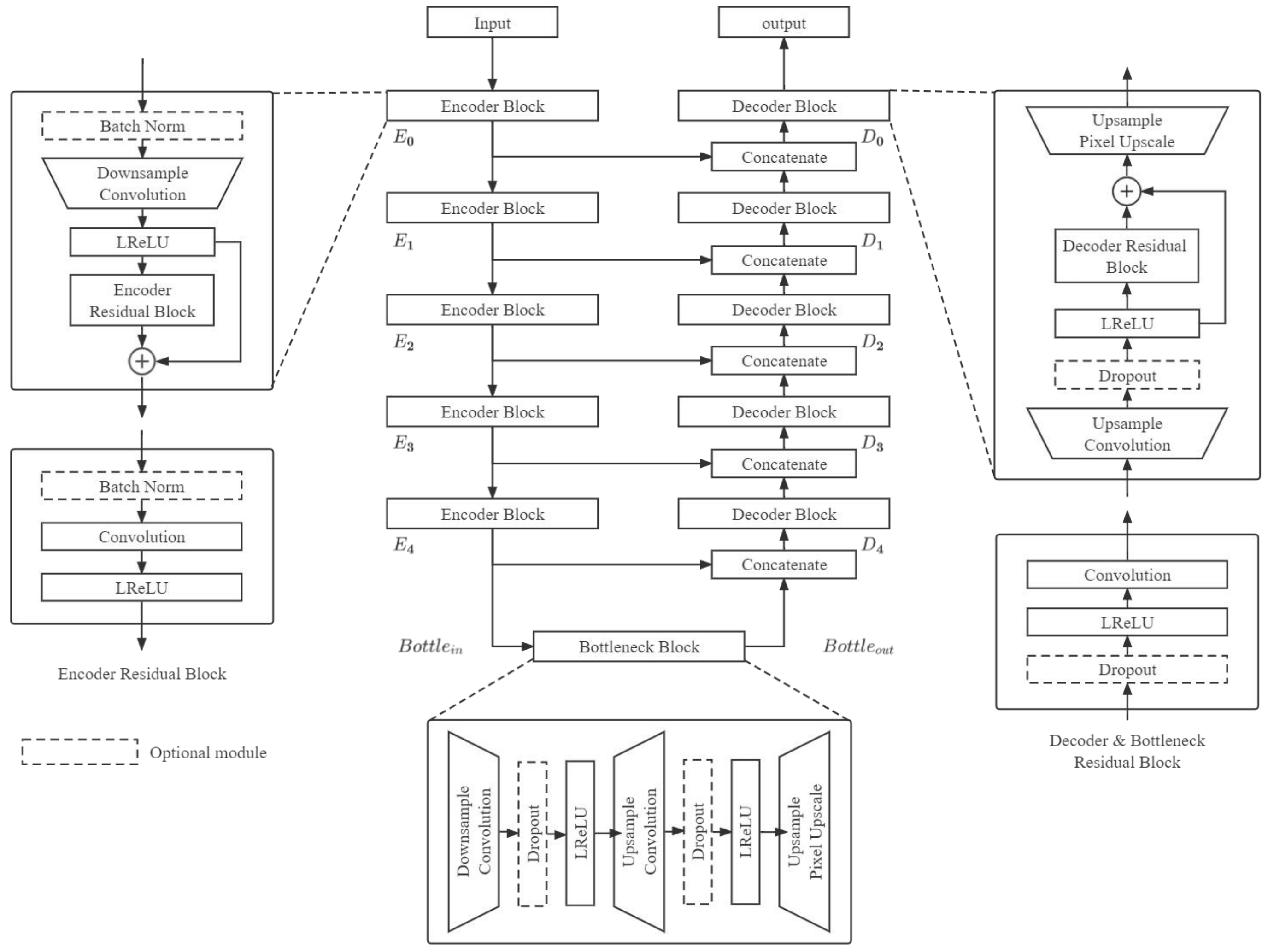

Fig. 1. The Architecture of ARUNet. The central part of the figure is the backbone of ARUNet. On the left is the stack of encoder blocks, on the right is the stack of decoder blocks, and a bottleneck module is connected in the middle. The internal structures of encoder block, decoder block and bottleneck module are extended and displayed next to them. At the bottom of encoder block and decoder block, the implementation details of the residual block used by them are shown.

the size of data which sent to the first stage decoder is valid. Φ is the pixel shuffle upsampling block. Consequently, we define the encoding process in below:

$$\begin{cases} Bottle_{in} = E_4 \\ Bottle_{out} = \Phi(F^{-1}(F(Bottle_{in}))) \end{cases} \tag{3}$$

We are now prepare to generate the X^{HR} by feeding the decoders with the extracted features and the outputs of encoders. Here Σ stands for the concatenate process. As displayed in Fig. 1, As the reverse process of encoding, we define decoding process as follow:

$$\begin{cases} D_4 = \Phi(R^{-1}(F^{-1}(\Sigma(Bottle_{out}, E_4))) + F^{-1}(\Sigma(Bottle_{out}, E_4))) \\ D_k = \Phi(R^{-1}(F^{-1}(\Sigma(D_{k+1}, E_k))) + F^{-1}(\Sigma(D_{k+1}, E_k))), k \in [0,3] \\ \overline{X}^{HR} = D_0 \end{cases} \tag{4}$$

In summary, at an encoding step, ARUNet halves the spatial dimension and doubles the filter size; during upsampling, this is reversed. This bottleneck architecture is inspired by auto-encoders, known to promote the model to learn a hierarchy of features. In audio tasks, bottom layers are able to extract wavelet-style features, and the top ones may correspond to phonemes [31]. The model is entirely convolutional and may process input sequences of arbitrary length.

B. Encoder and decoder

Each encoder or decoder block in ARUNet comprises a convolution layer, and an LReLU activates the layer, a regularization layer, and a residual layer. However, the specific internal network is also different due to the different subtasks they handle.

Residual block A common issue with training vanilla feed-forward neural networks with various layers is vanishing gradient. Gradient back-propagated to the initial layers approaches zero due to repeated multiplications. As shown in Fig. 1, one residual block appears after the convolution upsampling block or the deconvolution downsampling block. Residual network [3] used residual blocks to eliminate this problem by only modeling a fraction of the difference between their inputs and outputs. Because U-style network architecture needs a certain depth to ensure feature extraction in different degrees, we introduce residual blocks into encoder and decoder modules according to the advantages of the residual mechanism. It is worth mentioning that, except that the number of convolution filters remains unchanged (the number of inpu t and output channels is equal) and the step size is 1, the parameters and arrangement order of all network blocks are consistent with the settings in the encoder or decoder block.

Regularization Dropout is a simple method to prevent overfitting, where activation units are dropped based on a fixed probability [32], introducing noise in the hidden layers and preventing excessive co-adaptation. Referring to [33], we do not put dropout and batch normalization in the same module but use batch norm first and then drop out to avoid variance shift in training and testing during normalization. [28] uses dropout layers instead of batch normalization. We followed this method in ARUnet, as shown in Fig 1, by setting dropout layers after each upsampling convolutional layer or using batch normalization to reduce the impact of noise in the data in order to reduce the over fitting phenomenon in the process of downsampling feature extraction.

Convolution layer Each block $k = 1,2,\ldots,K$ contains $max(2^{6+K}, 512)$ convolutional filters of length $min\left(2^{7-k}+1, 9\right)$ and a dilation of 1 and a stride of 2 or 1. When the step size is 2, the number of filters is twice (convolution layer) or half (deconvolution layer) of the previous layer, and the data is upsampled or downsampled; When the step size is 1, the number of filters is the same as that of the previous layer, and the convolution layer or deconvolution layer is used for the residual block.

C. *Data augmentation*

Data augmentation is usual in image-based tasks and mostly applied geometric transformations such as rotating, flipping, and cropping [34] . [5] proposed a data augmentation method where many different types of filters are used during training, instead of operating geometric transformations in audio signals. Research shows that this method is able to improve the performance of bandwidth expansion tasks.We adopt this method and add more filters to improve the robustness of the model.

D. *Training Details*

We train the proposed ARUNet model using Adam with $\mathcal{L}_2$ and decoupled weight decay [35] by setting $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The first learning rate is set to 5×10^{-4} and is gradually decayed following the scheme of Cosine annealing with restart [36] set to 10^{-7} and the restart performs at every 20000 iterations. Our model weights are initial randomly with values drawn from the normal distribution with zero mean and unit variance. The batch size is 8. We train our model on four Nvidia V100 with batch size set to 8. We recode the avarage training loss every 5000 iterations.Traning samples are created by randomly selecting audio from the whole training set and then sampling the audio from

TABLE I. FILTERS SETTING FOR DATA AUGMENTATION

Purpose	**Single-filter (No data augmentation)**	**Table Column Head (Data augmentation)**
Training	Chebyshev-1,6	Chebyshev-1,6 Chebyshev-1,8 Chebyshev-1,10 Chebyshev-1,12 Bessel, 6 Bessel, 12 Elliptic, 6 Elliptic, 12
Validation with seen filter(s)		
Validation with unseen filter	Butterworth, 6	Butterworth, 6
Testing with unseen filter		
Training with seen filter	Chebyshev-1,6	Chebyshev-1,6

the sliding window from the beginning. We define the frame length of audio as 20 ms, hop length as 10 ms, and sliding window length as 20 ms, according to [37]. In this way, three complete frames and two half frames can be obtained by one sampling.

The mean-square error (MSE) is computed between the estimated frame sequence $\bar{X}^{HR}$ and the ground truth $X^{H}R$, here n is the number of frames in the sliding sampling window:

$$L(\bar{X}^{HR}, X^{HR}) = \frac{1}{n}\sum_{i=0}^{n}\left(\bar{x}_i^{HR} - x_i^{HR}\right)^2 \tag{5}$$

III. EXPERIMENT

A. *Evaluation Metrics*

The mean-square error (MSE) is computed between the estimated frame sequence $\bar{X}^{HR}$ and the ground truth $X^{H}R$, here n is the number of frames in the sliding sampling window:

$$SNR(x, \bar{x}) = 10lg\frac{\|x\|_2^2}{\|x-\bar{x}\|_2^2} \tag{6}$$

where x is the reference signal and $\bar{x}$ is its approximation. The signals are used in their stereo forms while calculating the 2-norms. In order to provide extra insight into performance, we evaluate the perceptual quality of the output audio samples by using the VGG distance, Li et al applied it for the evaluation of music enhancement [27] . The VGG distance between two audio sequences is defined as the distance between their embeddings created by the pre-trained VGGish network on audio classification [26] . [38] shows that the distance between deep embeddings correlates better to human validation, compared to Perceptual Evaluation of Speech Quality (PESQ) [39] and Virtual Speech Quality Objective Listener (ViSQOL) [40] . We modify VGGish opensource implementation [41] . According to [38], we emplay the mean absolute distance to define the VGG distance:

$$VGG(x, \bar{x}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}_i| \tag{7}$$

where x is the reference signal, $\bar{x}$ is its approximation; y and $\bar{y}$ are their embeddings, respectively. n is the size of the embedding tensors, depending on the length of x.

B. Datasets

ARUNet is intended to be agnostic to musical style so that any uncompressed full-bandwidth musical content could be used as training material. However, we select the following two publicly available datasets to allow reproducibility: Mancini Piano Dataset [42] (26.7 minutes) and MAESTRO Dataset v3 [43] (201 hours). The former is the test set which contains only piano samples from Johann Sebastian Bach, and the uncompressed audio is of CD-quality (49kHz, 16-bit PCM stereo). The latter is the training set and verification set, based on recordings from the International Piano Competition, and the uncompressed audio is of CD-quality or higher (44.1-48 kHz, 16-bit PCM stereo). We uniformly convert all the datasets to 48 kHz sampling rate.

C. Results

In this sub-section, we select two open-source baseline models: U-Net and ResNet [5], because we hope that the comparison of verification and test results will reflect the progressiveness of ARUNet in taking the advantages of both.

Validation In Fig. 3. we comparing input and model output samples against the ground-truth to measure the input and output SNR levels.Since the inputs are not affected by training, their SNR level stays constant throughout and constitutes a baseline. The seen and unseen filters are detailed in Table I. The SNR levels are computed by taking the average across multiple filters. Apparently, our model ensures continuous optimization while ensuring the convergence of the effect, and there is no deterioration of the output results that may be caused by over fitting with the increase of iterations like ResNet.

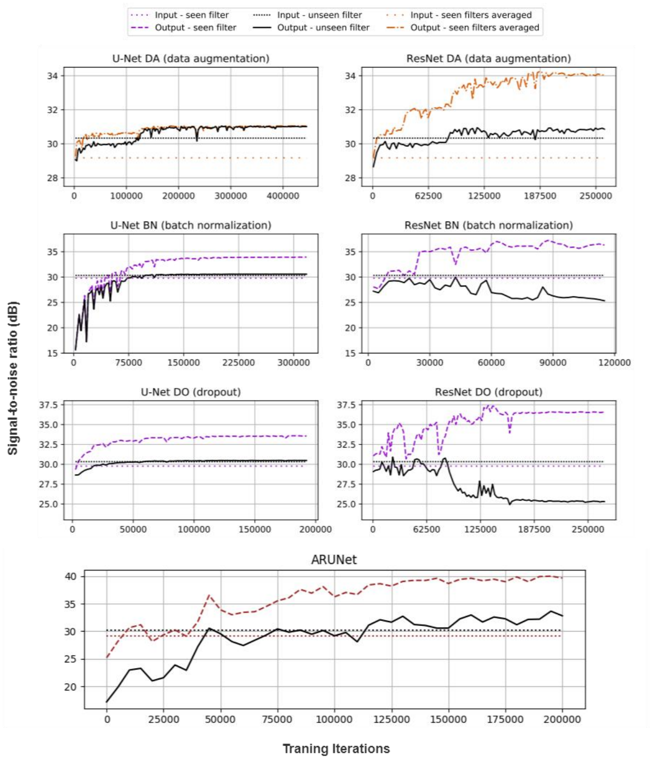

Fig. 2. Validation performance of baselines and our model throughout training.

TABLE II. Output signal-to-noise ratio (SNR) and absolute VGG distances (VGG) on the test dataset

Filter	Experiment	SNR	Δ SNR	VGG	- Δ VGG
Cheby-shev1,6 (seen filter)	Input	27.86	/	46.55	/
	U-Net	30.34	+2.47	41.04	+5.51
	U-Net DA	29.78	+1.91	44.29	+2.26
	U-Net BN	30.90	+3.30	41.52	+5.03
	U-Net DO	30.39	+2.62	41.51	+5.04
	Res-Net	34.94	+7.08	39.02	+7.53
	Res-Net DA	30.48	+2.62	40.11	+6.43
	Res-Net BN	34.37	+6.50	39.41	+7.14
	Res-Net DO	35.27	+7.41	37.23	+9.32
	ARUNet	**39.82**	**+11.96**	**33.73**	**+12.82**
Butter-worth,6 (unseen filter)	Input	27.37	/	47.11	/
	U-Net	28.55	+1.18	41.90	+5.21
	U-Net DA	29.00	+1.63	44.80	+2.31
	U-Net BN	28.77	+1.40	42.06	+5.06
	U-Net DO	28.62	+1.24	42.34	+4.78
	Res-Net	21.96	-5.41	47.12	-0.01
	Res-Net DA	29.16	+1.78	40.52	+6.59
	Res-Net BN	23.23	-4.14	46.38	+0.73
	Res-Net DO	22.10	-5.27	46.15	+0.96
	ARUNet	**32.05**	**+4.68**	**38.97**	**+8.14**

Testing Table II displays the evaluation result of baselines and our model. For SNR, ΔSNR and −ΔVGG higher is better and for VGG lower is better. DA, BN, and DO correspond to data augmentation, batch normalization, and dropout, respectively. The best performing model is ARUNet, improving upon the input SNR by 11.96 dB and 4.68 dB with both seen and unseen filter.

Visualization It is concluded from the validation subsection that ResNet has the best performance among baselines, so we will take it as baseline in this section. The inspection of the figure reveals quite different behavior of the ARUNet compared to the baseline in Fig. 2. Arunet has well restored the details of the target. ResNet has generally achieved this goal, but there is an over enhanced problem. It can be seen that there is a lot of noise in the results, and the overall bandwidth expansion is excessive. This phenomenon further reflects the superiority of ARUNet.

IV. Conclusion

Deep learning techniques based on neural networks have been successful at solving under-defined problems in signal processing such as audio bandwidth extension, image super-resolution, in-painting, et al. Learning-based methods often perform better in this context than general-purpose algorithms because they leverage sophisticated domain-specific models of the appearance of natural signals.

In this paper, we proposed new techniques that utilize this insight to upsample audio signals. Our technique extends previous work on transfering the general deep learning neural network to the task of audio bandwidth expansion, it

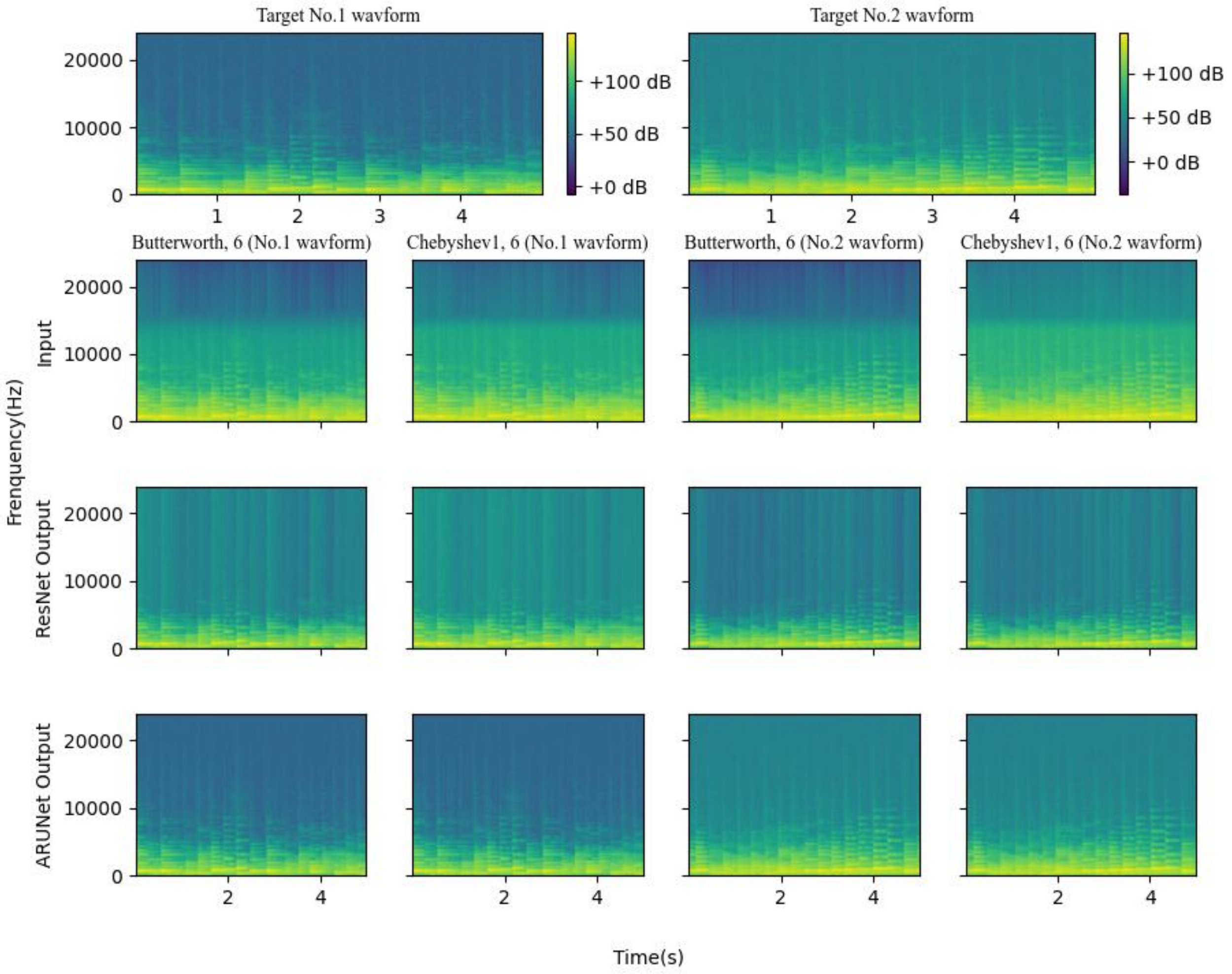

Fig. 3. Spectrograms of the sample audio segments. We randomly selected two audio clips with a length of 5 seconds in the test results. The subgraph in the first line shows the spectrum of the original audio, the second line is the analog low-pass input generated by unseen and see filters respectively, and the third and fourth lines are the processing results of baseline and arunet respectively.

outperforms previous similar approaches on non-vocal music. Our approach is fast and simple to implement, and has applications in streaming media, text-to-speech generation and audio enhancement. It also demonstrates the effectiveness of end-to-end architectures on an audio generation task, suggesting new directions for generative audio modeling.

REFERENCES

[1] P. Ekstrand, “Bandwidth extension of audio signals by spectral band replication,” 2002.l Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International conference on medical image computing and computer-assisted intervention, 2015, pp. 234–241.

[3] 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.

[4] K. Kumar et al., “Melgan: Generative adversarial networks for conditional waveform synthesis,” Advances in neural information processing systems, vol. 32, 2019.

[5] S. Sulun and M. E. Davies, “On filter generalization for music bandwidth extension using deep neural networks,” IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 1, pp. 132–142, 2020.

[6] P. J. Jax, Enhancement of bandlimited speech signals: Algorithms and theoretical bounds. Mainz, 2002.

[7] Y. M. Cheng, D. O’Shaughnessy, and P. Mermelstein, “Statistical recovery of wideband speech from narrowband speech,” IEEE Transactions on Speech and Audio Processing, vol. 2, no. 4, pp. 544–548, 1994.

[8] Y. Nakatoh, M. Tsushima, and T. Norimatsu, “Generation of broadband speech from narrowband speech using piecewise linear mapping,” 1997.

[9] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, “Speech enhancement via frequency bandwidth extension using line spectral frequencies,” in 2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (cat. No. 01CH37221), 2001, vol. 1, pp. 665–668.

[10] Y. Yoshida and M. Abe, "An algorithm to reconstruct wideband speech from narrowband speech based on codebook mapping." in ICSLP, 1994, vol. 94, pp. 1591−1594.

[11] J. Epps and W. H. Holmes, " A new technique for wideband enhancement of coded narrowband speech," in 1999 IEEE workshop on speech coding proceedings. Model, coders, and error criteria (cat. No. 99EX351), 1999, pp. 174−176.

[12] P. Jax and P. Vary, " On artificial bandwidth extension of telephone speech," Signal Processing, vol. 83, no. 8, pp. 1707−1719, 2003.

[13] P. Bauer and T. Fingscheidt, " An HMM-based artificial bandwidth extension evaluated by cross-language training and test," in 2008 IEEE international conference on acoustics, speech and signal processing, 2008, pp. 4589−4592.

[14] G.-B. Song and P. Martynovich, "A study of HMM-based bandwidth extension of speech signals," Signal Processing, vol. 89, no. 10, pp. 2036−2044, 2009.

[15] K.-Y. Park and H. S. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *2000 IEEE international conference on acoustics, speech, and signal processing. Proceedings (cat. No. 00CH37100)*, 2000, vol. 3, pp. 1843–1846.

[16] A. H. Nour-Eldin and P. Kabal, " Mel-frequency cepstral coefficient-based bandwidth extension of narrowband speech," 2008.

[17] D. Bansal, B. Raj, and P. Smaragdis, " Bandwidth expansion of narrowband speech using non-negative matrix factorization. " in INTERSPEECH, 2005, pp. 1505−1508.

[18] D. L. Sun and R. Mazumder, " Non-negative matrix completion for bandwidth extension: A convex optimization approach," in 2013 IEEE international workshop on machine learning for signal processing (MLSP), 2013, pp. 1−6.

[19] B. Iser and G. Schmidt, " Neural networks versus codebooks in an application for bandwidth extension of speech signals," 2003.

[20] Y. Wang, S. Zhao, W. Liu, M. Li, and J. Kuang, "Speech bandwidth expansion based on deep neural networks," 2015.

[21] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," 2015.

[22] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," 2015.

[23] S. Kim and V. Sathe, "Bandwidth extension on raw audio via generative adversarial networks," arXiv preprint arXiv:1903.09027, 2019.

[24] S. Kim and V. Sathe, "Bandwidth extension on raw audio via generative adversarial networks," arXiv preprint arXiv:1903.09027, 2019.

[25] J. Sautter, F. Faubel, M. Buck, and G. Schmidt, "Artificial bandwidth extension using a conditional generative adversarial network with discriminative training, " in ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2019, pp. 7005−7009.

[26] S. Hershey et al., " CNN architectures for large-scale audio classification, " in 2017 ieee international conference on acoustics, speech and signal processing (icassp), 2017, pp. 131−135.

[27] Y. Li, B. Gfeller, M. Tagliasacchi, and D. Roblek, "Learning to denoise historical music," arXiv preprint arXiv:2008.02027, 2020.

[28] V. Kuleshov, S. Z. Enam, and S. Ermon, " AUDIO SUPER-RESOLUTION USING NEURAL NETS, " arXiv preprint arXiv:1708.00853, 2017.

[29] T. Y. Lim, R. A. Yeh, Y. Xu, M. N. Do, and M. Hasegawa-Johnson, "Time-frequency networks for audio super-resolution," in 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2018, pp. 646−650.

[30] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1874−1883.

[31] Y. Aytar, C. Vondrick, and A. Torralba, " Soundnet: Learning sound representations from unlabeled video," Advances in neural information processing systems, vol. 29, 2016.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, " Dropout: A simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929−1958, 2014.

[33] X. Li, S. Chen, X. Hu, and J. Yang, " Understanding the disharmony between dropout and batch normalization by variance shift, " in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2682−2690.

[34] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation, " in 2018 IEEE symposium series on computational intelligence (SSCI), 2018, pp. 1542−1547.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2018.

[36] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation," 2018.

[37] T. Theodorou, I. Mporas, and N. Fakotakis, "An overview of automatic audio segmentation," International Journal of Information Technology and Computer Science (IJITCS), vol. 6, no. 11, p. 1, 2014.

[38] P. Manocha, A. Finkelstein, R. Zhang, N. J. Bryan, G. J. Mysore, and Z. Jin, " A differentiable perceptual audio metric learned from just noticeable differences," 2020.

[39] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs, " in 2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (cat. No. 01CH37221), 2001, vol. 2, pp. 749−752.

[40] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, " ViSQOL: The virtual speech quality objective listener, " in IWAENC 2012; international workshop on acoustic signal enhancement, 2012, pp. 1−4.

[41] " Models/research/audioset/vggish at master · tensorflow/models, " GitHub. Accessed: Jun. 15, 2022. [Online]. Available: https://github.com/tensorflow/models

[42] "PianoDataset," UCSD. Accessed: Jun. 15, 2022. [Online]. Available: http://deepyeti.ucsd.edu/cdonahue/wavegan/data/mancini_piano.tar.gz

[43] " The MAESTRO Dataset. " Accessed: Jun. 15, 2022. [Online]. Available: https://magenta.tensorflow.org/datasets/maestro

AUTHORS' BACKGROUND

Your Name	Title*	Research Field	Personal website
Linlin Ou	master student	Audio/Video AI Enhancement, Audio/Video Coding	
Yuanping Chen	full professor	Key technology research, data intelligence applications for large information systems	