TIME-FREQUENCY NETWORKS FOR AUDIO SUPER-RESOLUTION

Teck Yian Lim[*]*, Raymond A. Yeh*[*]*, Yijia Xu, Minh N. Do, Mark Hasegawa-Johnson*

University of Illinois at Urbana Champaign, Champaign, IL, USA
Department of Electrical and Computer Engineering
{tlim11, yeh17, yijiaxu3, minhdo, jhasegaw}@illinois.edu

ABSTRACT

Audio super-resolution (*a.k.a.* bandwidth extension) is the challenging task of increasing the temporal resolution of audio signals. Recent deep networks approaches achieved promising results by modeling the task as a regression problem in either time or frequency domain. In this paper, we introduced Time-Frequency Network (TFNet), a deep network that utilizes supervision in both the time and frequency domain. We proposed a novel model architecture which allows the two domains to be jointly optimized. Results demonstrate that our method outperforms the state-of-the-art both quantitatively and qualitatively.

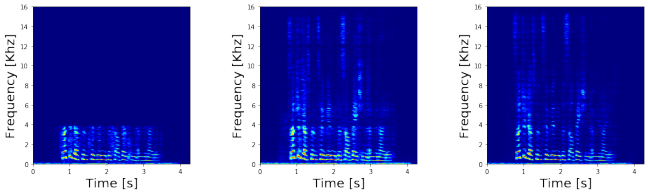
***Index Terms*—** Bandwidth extension, audio super-resolution, deep learning.

1. INTRODUCTION

Super-resolution (SR) is the task of reconstructing high-resolution (HR) data from a low-resolution (LR) input. This is a challenging task due to its ill-posed nature, especially when the upscaling factor is high. From tackling the SR problem we can gain understanding of the data priors, and lead to improvements in related areas such as compression and generative modeling.

Recently, image super-resolution algorithms have received strong attention in the computer vision community, and achieved remarkable success by modeling SR as a regression task with deep neural networks. In this work we explore the analogous SR task for audio data, (*i.e.* learning a mapping from LR to HR audio frames). To visualize the reconstruction, in Fig. 1 we show the spectrograms of the LR input, the HR reconstruction and the ground truth.

Prior works, such as, Li *et al.* [1] propose a deep neural network to learn the LR to HR mapping of spectral magnitude and completely ignoring the phase of the missing high frequency component. In [2], Kuleshov *et al.* propose a deep neural network to learn the LR to HR mapping directly in the time domain. While these models show promising results, each model only operates in either time or frequency domain and focuses on different aspect of the signal.



Input x Prediction $\hat{y}$ Ground Truth y

Fig. 1. Spectrogram corresponding to the LR input (frequencies above 4kHz missing), HR reconstruction, and the HR ground truth. Our approach successfully recovers the high frequency components from the LR audio signal.

To take advantage of both time and frequency domain information, we propose Time-Frequency Network (TFNet) a deep neural network, which chooses when to use the time and frequency information for audio SR.

At the first glance, modeling in both frequency and time domain seems like a redundant representation; From Parseval's theorem the ℓ_2 difference of prediction error, whether in the frequency or time domain is exactly the same. However, regression from LR to HR in time or frequency domain solves a very different problem. In the time domain, it is analogous to the image super-resolution task, mapping "audio patches" from LR to HR. On the other hand, SR in the frequency domain is analogous to the *semantic image inpainting* task [3, 4]. Given the low frequency components of a spectrogram, output the high frequency components, see Fig. 2 for illustration. Therefore, to exploit the best of both worlds, we propose to model audio SR jointly in both time and frequency domains.

Experiments on two datasets show that our approach outperforms, the state-of-the-art methods on quantitatively metrics and qualitatively the reconstructions are more natural.

2. RELATED WORK

Bandwidth Extension

The task of audio super-resolution is studied as bandwidth extension by the speech community. Various approaches

[*]Indicating Equal Contribution. Audio samples and software will be released on author's website at http://tlim11.web.engr.illinois.edu/.

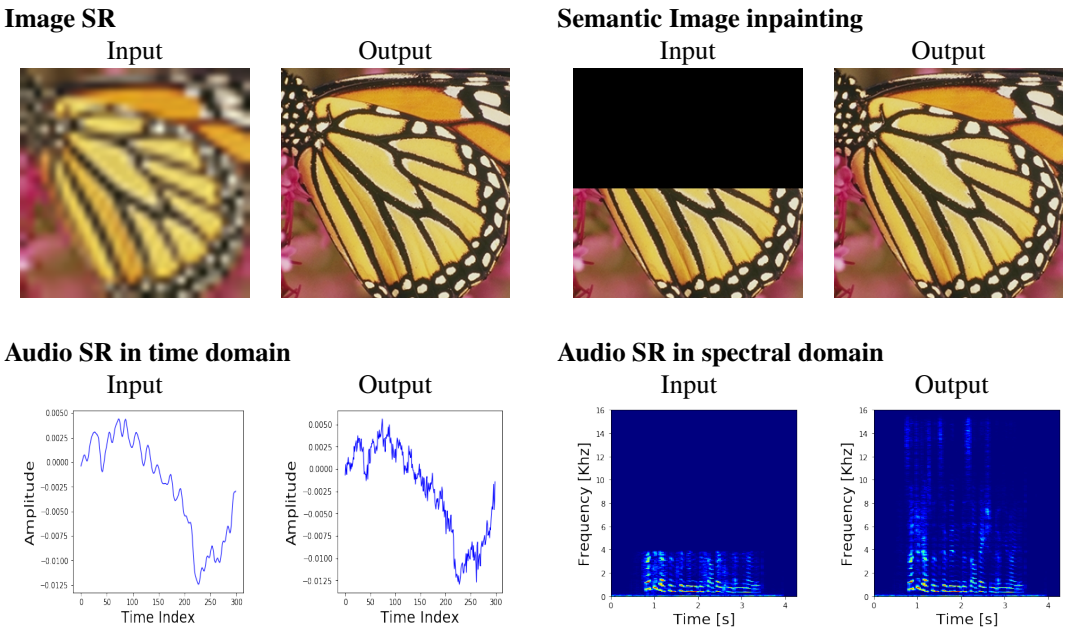


Fig. 2. Illustration of the input output for image SR, semantic image inpaiting, and audio SR in time and frequency domain. Audio SR in time domain is analogous to image SR, where "edges" are missing in the LR input. On the other hand, Audio SR in spectral domain can be viewed as image inpainting of spectrograms, *i.e.*, given the bottom low frequency "image", predict the remaining image.

have been proposed to estimate the high-frequency component using the low frequency band [5]. For example, linear mappings [6, 7], mixture models [8, 9, 10], and neural networks [11, 12, 1, 2].

Deep Nets for Single Image Super-resolution

Deep convolution neural networks (CNNs) have been the state-of-the-art for single image super-resolution. Many architectures have been proposed [13, 14, 15]. The models are all fully convolutional and with skip/residual connections from the earlier layers.

Deep Nets for Semantic Image Inpainting

Deep neural network has also demonstrated strong performance in the task of semantic image inpainting. Using CNNs, [3, 4] demonstrated the possibility of predicting masked regions in an image. Similar to super-resolution, the models are, again, fully convolutional. Taking inspiration from these models, our deep network architecture also follows a similar design principles.

3. APPROACH

We formulate audio SR as a regression task, *i.e.*, predict the HR audio frames , $y \in \mathbb{R}^L$, given the LR audio frames, $x \in \mathbb{R}^{L/R}$, where R is a down-sampling factor.

3.1. Time-Frequency Network

We propose Time-Frequency Network (TFNet), a fully differentiable network that can be trained end-to-end. As illustrated in Fig. 3, let Θ be all the parameters in the model, our model consists of a fully convolutional encoder-decoder based network $\mathcal{H}(x;\Theta)$. For a given LR input x, $\mathcal{H}$ predicts the HR audio reconstruction, $\hat{z}$, and the HR spectral magnitude $\hat{m}$. Using our proposed spectral fusion layer we synthesize the final output.

Spectral Fusion Layer

The spectral fusion layer combines the $\hat{z}$ and $\hat{m}$ to output the final reconstruction $\hat{y}$, shown below:

$$M = w \odot |\mathscr{F}(\hat{z})| + (1 - w) \odot \hat{m},$$

$$\hat{y} = \mathscr{F}^{-1}(Me^{j\angle\mathscr{F}(\hat{z})}),$$

where $\mathscr{F}$ denotes the Fourier transform, $\odot$ is a element-wise multiplication and w is a trainable parameter.

This layer is differentiable and can be trained end-to-end. The key advantage is that this layer enforces the network to explicitly model the waveform's spectral magnitude, while remaining of the model can model phase in the time domain.

Our design of the network architecture comes from the observation that convolution layers can only capture local relationships, and are particularly good at capturing visual features. When we visualize the magnitude and phase using of short-time Fourier transform , there are clear visual structures in the magnitude but not the phase; hence, we only model the magnitude in the spectral domain.

Spectral Replicator

As previously mentioned, convolutional layer typically captures local relationships, (*i.e.*, the range of the input-output relationship is limited by the receptive field). This causes an issue as we want the output's high frequency component to

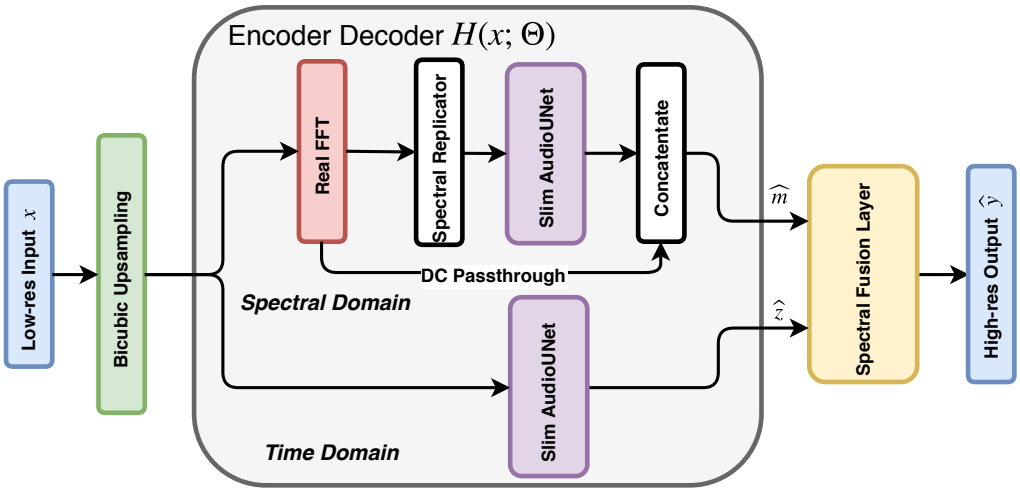
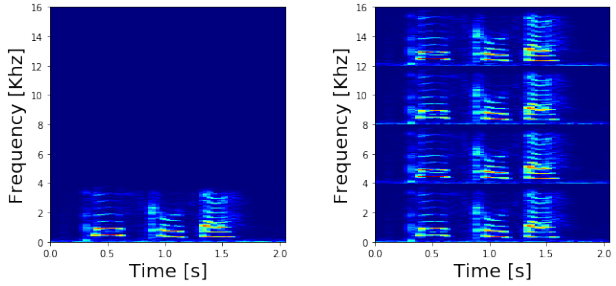


Fig. 3. Overall pipeline of Time-Frequency Network. TFNet utilizes both the time and frequency domain to accomplish audio SR. TFNet contains a branch which explicitly models the reconstruction's spectral magnitude, while the other branch models the reconstruction in time domain. The output of the two branches are finally combined with our Spectral fusion layer to synthesize the high resolution output.



Input **Output**

Fig. 4. Illustration of the spectral replicator layer on 4x SR task. The low frequency components are replicated four times to replace the zeros.

depend on the input's low frequency components. For example, when upsampling by a factor of four, the receptive field needs to be at least $\frac{3}{4}$ of the total frequency bins, which will require either very large kernels, or many layers. To address this issue of receptive field, we replicate the available low frequency spectrum into the high frequency spectrum, which are initially all zeros, as illustrated in Fig. 4.

Loss Function

For training our network, we utilize the ℓ_2 reconstruction loss with weight decay. The overall objective function is to minimize the following loss function with respect to Θ:

$$\mathcal{L} = \sum_{(x,y)\in\mathcal{D}} \|y - \hat{y}(x)\|_2 + \lambda \|\Theta\|_2 \,, \qquad (1)$$

where $\mathcal{D}$ is the training set of all (LR, HR) pairs, and λ is the weighting hyperparameter for the regularizer, chosen to be 0.0001 in all our experiments.

3.2. Implementation Details

Preprocessing

For training, we performed silence filtering to discard sequences below an energy threshold of 0.05 computed as $\sum_{n=0}^{N-1} x[n]^2$. We found that this improves training convergences and stabilizes the gradient. For testing and evaluation, we do not filter out the silences.

Network Architecture

Our network consists of two branches with similar architectures; a time domain branch and a frequency domain branch. For a fair comparison, our network follows the architecture design patterns from AudioUNet [2], consisting of encoder and decoder blocks. To keep the model size approximately the same, the number of filters are halved in each of the branch. Our network takes segments of 8192 length of audio as input.

For the frequency domain branch, we performed a Discrete Fourier Transform (DFT) on the sequence. Since all audio signals are real values, we discarded all components corresponding to negative phase, resulting in 4097 Fourier coefficients. Lastly, we take the magnitude of these coefficients.

As previously mentioned, the high frequency components of the input are zeros, thus using the spectral replicator, we replace the zero values with copies of the low frequency components. Specifically, for 4x upsampling we repeat the 1$^{\text{st}}$ component to the 1024$^{\text{th}}$ component at 1025 to 2048, 2049 to 3072 and finally 3073 to 4096. The 0$^{\text{th}}$ component (DC component) is passed directly through the network and fused at the end.

Training Details

We use the popular Adam optimizer [16] for training our network. The starting learning rate is $3e^{-5}$, we used the polynomial learning rate decay scheduling with rate of 0.5. All our

models are trained for 500,000 steps.

4. EXPERIMENTS

Datasets and Preperation

We evaluate our method on two datasets: The VCTK dataset [17] and Piano dataset [18].

The VCTK dataset contains speech data from 109 native speakers of English. Each speaker reads out approximately 400 different sentences, and sentences also different from speaker to speaker, which totals to 44 hours of speech data.

Following the previous works [2], we split the data into 88% training 6% validation, and 6% testing, with no speaker overlap.

For each of the files in the data set, we resampled the audio into a lower sampling rate by performing a low-pass filter with cut-off frequency at the Nyquist rate of the target lower sampling rate. This LR sequence is then upsampled back to the original rate via bi-cubic interpolation. To prepare the training (LR, HR) pairs, we extract 8192 samples length subsequences with 75% overlap from the resampled signal and its corresponding original signal.

For the VCTK dataset of with 16kHz sampling rate, this corresponds to subsequences of approximately 500ms with the start of every subsequence 125ms apart from each other. 50% of the remaining sequences is then discarded as the resulting data set is simply too large to train effectively.

Furthermore, to understand whether the model performance is affected by data diversity, we formed a new dataset (VCTK_s) which only includes speaker one subset of VCTK. This contains approximately 30 minutes of speech. The audio data are provided at the sampling rate of 16kHz.

Piano dataset contains 10 hours of Beethoven sonatas at the sampling rate of 16kHz. Due to the repetitive nature of music, we split the Piano dataset at file level for a fair evaluation.

Evaluations

For evaluation, we compute similarity metrics of Signal to Noise Ratio (SNR) and Log-Spectral Distance(LSD).

The SNR captures a weighted difference between the prediction and the ground-truth data in the time domain. On the other hand, LSD captures the difference between the prediction and the ground-truth in the frequency domain [19].

$$\text{LSD}(y, \hat{y}) = \frac{10}{L} \sum_{l=1}^{L} \left\| \log_{10} \mathscr{F}(y_l) - \log_{10} \mathscr{F}(\hat{y_l}) \right\|_2 , \quad (2)$$

where the subscript l denotes the index of short windowed segments of the audio.

Results

We compare our approach with three different baselines, a simple bicubic interpolation and two deep network methods using the reported results in [1, 2] in Tab. 1. In particular, we experimented with different rates of downsampling, starting at the rate of 4, where the degrade in quality becomes audible. For the VCTK, our approach outperforms in the baseline methods by approximately 1.5dB in SNR for the 4x upsampling case. For 8x upsampling even outperforms the baseline's 6x upsampling results by 1.5dB SNR. On the Piano dataset, our method performs on par with the baseline method. It is to note that the number of parameters in [2] is the same as our model; This further demonstrates that our model's architecture is more effective in its representation.

Model	Rate	VCTK_s	VCTK	Piano
Bicubic	4	14.8 / 8.2	13.0 / 14.9	22.2 / 5.8
Li *et al*. [1]	4	15.9 / 4.9	14.9 / 5.8	23.0 / 5.2
Kuleshov *et al*. [2]	4	17.1 / 3.6	16.1/ 3.5	**23.5** / 3.6
Ours	4	**18.5 / 1.3**	**17.5 / 1.27**	23.1 / **3.4**
Bicubic	6	10.4 / 10.3	9.1 / 10.1	15.4 / 7.3
Kuleshov *et al*. [2]	6	14.4 / 3.4	10.0 / 3.7	16.1 / 4.4
Bicubic	4	9.9 / 20.5	8.7 / 18.34	14.5 / 11.59
Ours	8	**15.0 / 1.89**	**12.0 / 1.90**	**15.69 / 9.64**

Table 1. Quantitative comparison on the test set at different upsampling rate. Left/right results are SNR/LSD.

Model	Rate	VCTK
Time Branch Only	4	11.71 / 4.89
Spectral Branch Only	4	7.73 / 1.5
Both Branches	4	17.5 / 1.27

Table 2. Ablation study evaluating the performance each of the time and spectral branch. Left/right results are SNR/LSD.

Detailed Analysis

Furthermore, to confirm that our network architecture utilizes both the time and frequency domain, we conduct an ablation study. We evaluate the model performance by removing the time or frequency domain branch, shown in Tab. 2. For the spectral branch, we assumed zero phase for the high frequency components during reconstruction.

5. CONCLUSION & FUTURE WORK

In this paper, we proposed Time-Frequency Network (TFNet), a deep convolutional neural network, which utilizes both time and frequency domain for the task of audio super resolution. We emprically demonstrated the superior performance of our novel spectral replicate and fusion layers compare to existing approaches. Lastly, TFNet has demonstrated that having a redundant representation helps the modeling for audio SR. We believe that the empirical results of the proposed method are interesting and promising, which warrant further theoretical and numerical analysis. Furthermore, we hope to generalize this observation to other audio tasks, such as audio generation, where the current state-of-the-art, WaveNet, [20] is a time domain approach.

6. REFERENCES

[1] Kehuang Li, Zhen Huang, Yong Xu, and Chin-Hui Lee, "Dnn-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Proc. INTERSPEECH*, 2015.

[2] Volodymyr Kuleshov, S Zayd Enam, and Stefano Ermon, "Audio super-resolution using neural networks," 2017.

[3] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros, "Context encoders: Feature learning by inpainting," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Raymond A. Yeh*, Chen Chen*, Teck Yian Lim, Schwing Alexander G., Mark Hasegawa-Johnson, and Minh N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, * equal contribution.

[5] Bernd Iser and Gerhard Schmidt, "Bandwidth extension of telephony speech," *Speech and Audio Processing in Adverse Environments*, pp. 135–184, 2008.

[6] Yoshihisa Nakatoh, Mineo Tsushima, and Takeshi Norimatsu, "Generation of broadband speech from narrowband speech using piecewise linear mapping," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[7] Yoshihisa Nakatoh, Mineo Tsushima, and Takeshi Norimatsu, "Generation of broadband speech from narrowband speech based on linear mapping," *Electronics and Communications in Japan (Part II: Electronics)*, vol. 85, no. 8, pp. 44–53, 2002.

[8] Geun-Bae Song and Pavel Martynovich, "A study of hmm-based bandwidth extension of speech signals," *Signal Processing*, vol. 89, no. 10, pp. 2036–2044, 2009.

[9] Hyunson Seo, Hong-Goo Kang, and Frank Soong, "A maximum a posterior-based reconstruction approach to speech bandwidth expansion in noise," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6087–6091.

[10] Saeed Vaseghi, Esfandiar Zavarehei, and Qin Yan, "Speech bandwidth extension: Extrapolations of spectral envelop and harmonicity quality of excitation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 3, pp. III–III.

[11] Juho Kontio, Laura Laaksonen, and Paavo Alku, "Neural network-based artificial bandwidth expansion of speech," *IEEE transactions on audio, speech, and language processing*, vol. 15, no. 3, pp. 873–881, 2007.

[12] Bernd Iser and Gerhard Schmidt, "Neural networks versus codebooks in an application for bandwidth extension of speech signals," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[14] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral)*, June 2016.

[15] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[16] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] Junichi Yamagishi, "English multi-speaker corpus for cstr voice cloning toolkit," `http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html`, 2012.

[18] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," 2016, cite arxiv:1612.07837.

[19] Augustine Gray and John Markel, "Distance measures for speech processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 380–391, 1976.

[20] Aron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Arxiv*, 2016.