Anyone GAN Sing

Shreeviknesh Sankaran[1], **Sukavanan Nanjundan**[1], **G. Paavai Anand**
Department of Computer Science and Engineering,
SRM Institute of Science and Technology,
Chennai, India.

***Abstract*—The problem of audio synthesis has been increasingly solved using deep neural networks. With the introduction of Generative Adversarial Networks (GAN), another efficient and adjective path has opened up to solve this problem. In this paper, we present a method to synthesize the singing voice of a person using a Convolutional Long Short-term Memory (ConvLSTM) based GAN optimized using the Wasserstein loss function. Our work is inspired by WGANSing by Chandna et al. Our model inputs consecutive frame-wise linguistic and frequency features, along with singer identity and outputs vocoder features. We train the model on a dataset of 48 English songs sung and spoken by 12 non-professional singers. For inference, sequential blocks are concatenated using an overlap-add procedure. We test the model using the Mel-Cepstral Distance metric and a subjective listening test with 18 participants.**

***Index Terms*—Generative Adversarial Networks, Wasserstein-GAN, Convolutional-LSTM, Singing Voice Synthesis.**

I. Introduction

The problem of singing voice synthesis is similar to that of Text-to-Speech (TTS) synthesis, but the former is much more complicated than the latter. The complexity mainly arises from trying to mimic an extensive range of pitches and phonemes involved in the process of singing. TTS synthesis is primarily controlled by the words or syllables from the text. On the other hand, singing voice synthesis is controlled by a score component in addition to the syllables from the lyrics of the song. The score component determines the pitch and timing of the syllables from the lyrics; in other words, the score defines the flow of a song.

There are several models (Chandna et al., 2019 [1], Blaauw et al., 2019 [2], Hono et al., 2019 [3], Kaewtip et al., 2019 [4], Lee et al., 2019 [5], and Tamaru et al., 2019 [6]) that have successfully demonstrated the ability to synthesize singing voices of different test subjects. Our model is inspired by WGANSing: A Multi-Voice Singing Voice Synthesizer Based on Wasserstein-GAN by Chandna et al.

Generative Adversarial Networks (GANs) have had immense success in modeling the distribution of highly complex data and have produced state-of-the-art results in image generation [7], [8]. GANs have also been used for TTS synthesis and other such audio generation problems [9], [10]. But, the number of adaptations of GANs in the audio domain is much fewer when compared to the number of adaptations in the computer vision domain.

The singing voice can be considered as a sequence as there is a sequential flow of notes throughout a song. A song can be constructed only if there is some connection between any two notes throughout the song. Notes thrown around haphazardly without any real flow or connection between the notes can't be considered as "legitimate" songs, although some people may find that attractive. This connection between notes can be considered as a sequence, and thus the problem of singing voice synthesis can be approached using sequence prediction techniques such as Long Short-Term Memory (LSTM). In this paper, we propose a Convolutional-LSTM (ConvLSTM) based GAN with an architecture inspired by Chandna et al. to synthesize the singing voice of a person. The choice of using LSTMs stems from the fact that they can model and learn long-range dependencies efficiently [11].

Therefore, we present a block-wise generative model trained using the Wasserstein—GAN framework for singing voice synthesis. The block-wise nature combined with the convolutional network component enables the model to identify temporal dependencies, just like the inter-pixel dependencies that are identified by GANs in the case of image datasets.

II. GAN and Wasserstein-GAN

GAN belongs to the generative frameworks class of deep learning. Since their inception, they have been widely used in the computer vision domain to generate synthetic images and videos that are indistinguishable from real samples [12]–[15]. They consist of two networks (adversaries), a generator, and a discriminator which are trained simultaneously. The discriminator is trained to distinguish between synthesized data and real data, whereas the generator is trained to fool the discriminator by synthesizing data that resembles real data. Training of GAN can be formulated as a minimax game [16]. The discriminator, on the one hand, tries to maximize its reward, and the generator, on the other hand, tries to minimize the discriminator's reward or, in other words, tries to maximize the discriminator's loss. The loss function for GAN is shown in Eq. (1).

$$L_{GAN} = \min_{G}\max_{D} \mathbb{E}_{x\sim P_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim P_z(z)}[log(1 - D(G(z))] \quad (1)$$

where G denotes the generator, D denotes the discriminator, x is a sample from the real distribution, and z is the input to the generator, which may be noise or some conditional input and is taken from the distribution P_z.

As pointed out by by Arjovsky et al., while GANs have been efficient in generating images and videos, it has been noted that the above minimax loss function can cause the GAN to get stuck in the early stages of training when the job of the discriminator is easy. More problems, such as vanishing gradient, mode collapse, and instability, arise. To overcome such difficulties, Wasserstein-GAN (WGAN) can be used [17]. WGAN uses Earth-Mover distance as given in Eq. (2) to measure divergence between real and generated distributions. Moreover, WGANs use a critic instead of a discriminator. The critic does not classify inputs as real or fake; instead, it just approximates a distance score between two given distributions (here, the real distribution and the generated distribution).

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}(\|x - y\|) \quad (2)$$

The critic can be optimally trained and it does not saturate, thus converging to a linear function. The loss function for WGAN is shown in Eq. (3).

$$L_{WGAN} = \min_{G}\max_{D} \mathbb{E}_{y\sim P_r}[D(y)] - \mathbb{E}_{x\sim P_x}[D(G(z))] \quad (3)$$

The loss functions for both the critic and the generator become deceptively simple. The critic tries to maximize Eq. (4) – i.e., it tries to maximize the difference between its output for real data and its output for synthesized data. The generator tries to maximize Eq. (5) – i.e., it tries to maximize the critic's output for fake or synthesized data.

$$L_C = D(x) - D(G(z)) \quad (4)$$

$$L_G = D(G(z)) \quad (5)$$

where $D(x)$ represents the critic's output for a real instance, $G(z)$ represents the generator's output when given noise z as input for any other conditional input z, and $D(G(z))$ represents the critic's output for a fake instance.

We use a conditional version of the model, parameterized by the network and conditioned on a conditional vector, described in Sec. V and follow the training algorithm proposed in the original paper.

We use an extension of GAN model called Conditional GAN (CGAN) which takes an additional conditional vector as input [18]. Adding this conditional vector controls the output and guides the generator in modeling a probability distribution controlled by the vector. The framework is mentioned in Fig. 5 in Sec. V. The training algorithm for the CGAN is the same as mentioned in the base paper.

III. LSTM AND CONVOLUTIONAL-LSTM

Long Short-Term Memory (LSTM) is an Recurrent Neural Network (RNN) architecture that has been extensively used for various applications in Natural Language Processing (NLP) such as speech recognition and semantic parsing [19]. LSTMs are capable of learning order dependence and long-range dependencies in sequence prediction problems [11]. An LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate as shown in Fig. 1. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell.

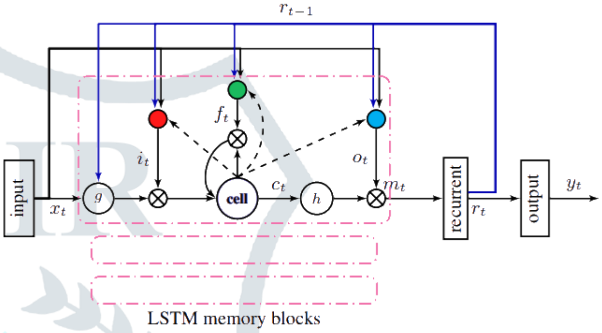

Fig. 1. A basic LSTM cell.

On the other hand, a Convolutional Neural Network (CNN) is a deep learning algorithm that is predominantly used in computer vision applications [20]. CNN is a regularized version of multilayer perceptron that is capable of efficiently extracting features and learning them. There are two parts to a CNN: convolution layers and a fully connected neural network that uses the output of the convolutions to predict the output. An example CNN is shown in Fig. 2.

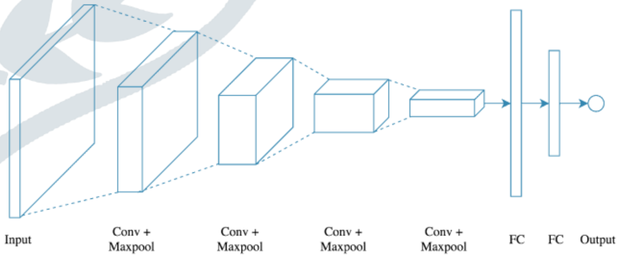

Fig. 2. A simple CNN architecture.

Convolutional Long Short-Term Memory (ConvLSTM) is an LSTM cell containing a convolution operation embedded inside it as shown in Fig. 3. It is an LSTM architecture designed explicitly for sequence prediction problems with spatial data, such as images or videos [21].

To take advantage of the abilities of both LSTM and CNN, we use ConvLSTM. ConvLSTM networks are capable of

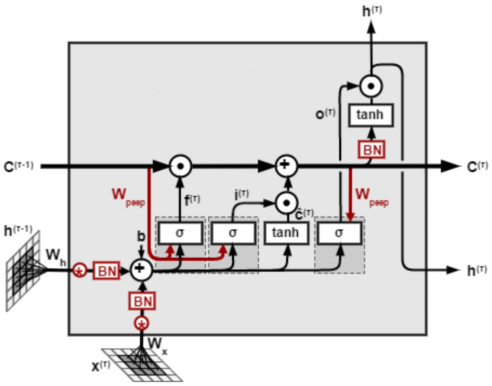

Fig. 3. A ConvLSTM cell.

learning long-range dependencies and extracting important features from data, both of which are required in the problem of singing voice synthesis.

IV. Related Work

The Neural Parametric Singing Synthesizer (NPSS) by Blaauw et al., is a modified version of WaveNet [22] which uses autoregressive architecture. The model features are produced by a parametric vocoder that separates the influence of pitch and timbre. As a result, this helps in training the model with datasets of comparatively smaller size while producing high-quality results, which are comparable to or sometimes even better than state-of-the-art concatenative methods.

Hono et al. present two methods for singing voice synthesis: one is a GAN-based architecture, and the other is a conditional GAN-based architecture. This models the inter-frame dependencies as opposed to the inter-block dependencies that are modeled by our model. This difference helps our model to produce more robust results than that of the model presented by Hono et al.

WGANSing by Chandna et al., which is the inspiration for our model, presents a multi-singer singing voice synthesizer. It uses an encoder-decoder based schema for the generator and an encoder schema for the discriminator network. The model produces results that are comparable to that of state-of-the-art models (NPSS in this case). They mention that the synthesis quality can be improved by using previously predicted block of features as a condition to the current batch of features which we have done so by using a ConvLSTM based GAN architecture for singing voice synthesis.

V. Proposed System

We adopt the same architecture used in the WGANSing paper, which is similar to the DCGAN. The main reason for this choice was to establish a baseline and make comparisons easier between models. One main difference between the WGANSing architecture and our architecture is that our generator network uses ConvLSTM layers instead of the CNN layers.

For the generator network, we use an encoder-decoder architecture consisting of 5 ConvLSTM layers each, as shown in Fig. 4. The whole network is similar to the U-Net architecture that is used for Biomedical Image Segmentation [23]. For the discriminator, we use the encoder block of the generator network alone. It asserts the presence of dependencies within a block.

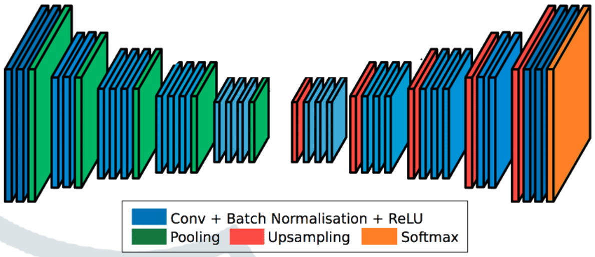

Fig. 4. The encoder-decoder architecture used in the generator network.

In the encoder block, we use fractionally-strided-convolutions instead of deterministic pooling functions. For example, if a 6x6 pixel image is processed by setting the stride to 3, and the kernel to 3x3, the resulting image is 2x2 in resolution. The inverse of this process begins by determining the spatial resolution and then performing the convolution. While it is not a mathematical inverse, the process is still useful in specific encoding mechanisms. Using this method increases the model's expressiveness ability.

Furthermore, the encoder-decoder schema leads to conditional dependence between the features of the generator output, within the predicted block of features. This approach implies implicit dependence between the features of a single block but not within the blocks themselves. Therefore, for inference, we use overlap-add of consecutive blocks of output features.

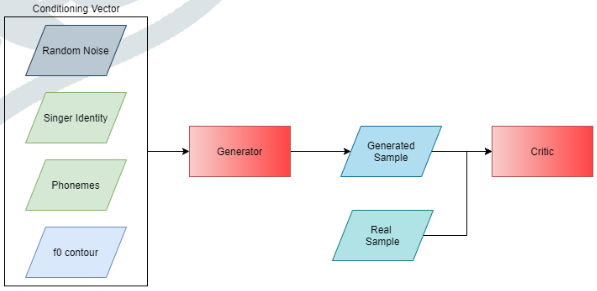

Fig. 5. The framework for the proposed model. A conditional vector is input to the generator. The critic is trained to identify a real sample.

As shown in Fig. 5, the generator network inputs a conditional vector consisting of random noise, identity, phonemes as a one-hot vector, and f0 contour. Using this conditional GAN, the singing voice of a person is generated.

VI. Dataset

The dataset used is the NUS-48E Sung, and Spoken Lyrics Corpus developed at the Sound and Music Computing Laboratory at the National University of Singapore [24]. The corpus is a 169-min collection of recordings of the sung and spoken lyrics of 48 (20 unique) English songs by 12 non-professional singers and a complete set of transcriptions and manual duration annotations at the phone-level for all recordings of sung lyrics, comprising of a total of 25,474 phone instances.

The corpus consists of 12 folders, one for each subject. Each of these folders consists of “sing” and “read” folders, which consist of 4 sung and corresponding spoken .wav files, and their time-aligned phone-level annotations in .txt files. The .wav and .txt files are converted into .hdf5 (hierarchical data format) files to make them easily accessible in Python. These files contain the phonemes and features of each corresponding .wav and .txt files, and thus the features can be used as inputs for the model.

VII. Evaluation Methodology

For objective evaluation, we use the Mel-Cepstral Distance metric [25] as shown in Eq. (6) and the results are presented in Tab. I. For subjective evaluation, we asked participants to listen to the songs generated by both the models and evaluate them on Audio Quality, Intelligibility, and Overall Score. We compared our model to the WGANSing model, both trained on the same dataset and for the same number of epochs – 750.

$$MC = \frac{10}{\ln 10}\frac{1}{T}\sum_{t=1}^{T}\sqrt{2\sum_{i}\left(C_{ti}-\hat{C}_{ti}\right)^{2}} \qquad (6)$$

We chose a total of 6 songs for the listeners, two songs of each gender without any voice change, two songs of each gender with voice change among the same gender and, two songs of each gender with voice change among opposite genders – i.e., male voiced song synthesized with a female voice and vice-versa.

VIII. Results

There were a total of 18 participants, who were all non-native English speakers and with ages in the range 20-22 in our study. The results of the study are shown in Figs. 6, 7, and 8.

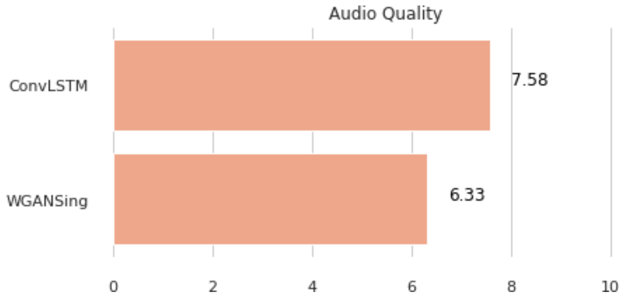

Fig. 6. Subjective test results for Audio Quality.

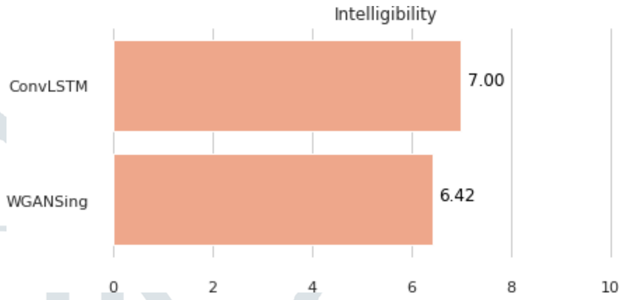

Fig. 7. Subjective test results for Intelligibility.

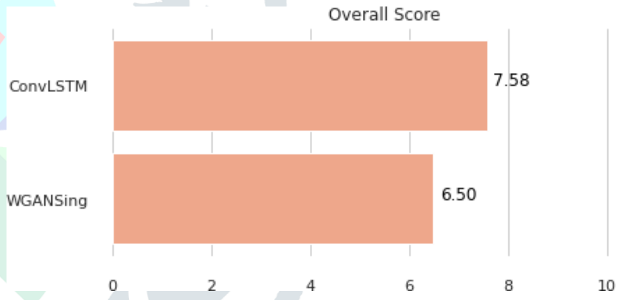

Fig. 8. Subjective test results for Overall Score.

From the above figures, it is observed that our model performs slightly better than the WGANSing model on all three attributes. This result is further corroborated by the objective measure presented in Tab. I.

From Figs. 6 and 7, it is also observed that while both models scored similarly on intelligibility, our model performs better when compared to WGANSing in terms of audio quality. This phenemenon can be mainly attributed to the fact that the songs generated with WGANSing had considerable noise between words or during pauses in the song. However, no such noise was heard in the songs generated by our model.

It is also observed that the model’s performance without voice change was better than the model’s performance with voice change. The model’s performance further declined when there was a voice change between different genders. Yet, even during the voice change, our model's performance was better than that of the WGANSing model.

TABLE I
MCD Results

Song	Anyone GAN Sing	WGANSing
MPUR 03	18.7567 dB	21.0440 dB
SAMF 13	14.3638 dB	14.6363 dB

IX. Conclusion

We have presented a multi-singer singing voice synthesizer using a ConvLSTM-based-conditional-GAN to model a block-wise sequence prediction problem. As this model is inspired by Chandna et al., we use the same block-wise methodology and architecture as used in their paper for the sake of comparison and testing. While our model seems to perform slightly better than the WGANSing model, both models seem to suffer in certain aspects such as lower audio quality and intelligibility in areas of high notes or pitch, and lower performance when there is a voice change, especially when there is a voice change between opposite genders.

On the whole, using LSTM cells along with convolutions have proven to be an improvement to the WGANSing model, which only had convolutions in an encoder-decoder based architecture. The improvement is mainly because of LSTM's sequence prediction capabilities, and as mentioned in Sec. I, song synthesis can be modeled as a sequence prediction problem.

We believe that the synthesis can be further improved by using algorithms to calculate and model an optimal match between two temporal sequences such as Dynamic Time Warping (DTW). This belief stems from the fact that there will be temporal misalignment between multiple sequences because of acceleration and deceleration during the course of an observation. For instance, using a nearest-neighbor classifier using DTW as the distance measure could improve performance significantly.

References

[1]P. Chandna, M. Blaauw, J. Bonada, and E. G ómez, "WGANSing: A Multi-Voice Singing Voice Synthesizer Based on the Wasserstein-GAN," 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, 2019, pp. 1-5.

[2]M. Blaauw and J. Bonada, "A neural parametric singing synthesizer," 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017), Stockholm, Sweden, 2017.

[3]Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing Voice Synthesis Based on Generative Adversarial Networks," 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, United Kingdom, 2019, pp. 6955-6959.

[4]K. Kaewtip, F. Villavicencio, F. Kuo, M. Harvilla, I. Ouyang, and P. Lanchantin, "Enhanced Virtual Singers Generation by Incorporating Singing Dynamics to Personalized Text-to-speech-to-singing," 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, United Kingdom, 2019, pp. 6960-6964.

[5]J. Lee, H. Choi, C. Jeon, J. Koo, and K. Lee, "Adversarially trained end-to-end Korean singing voice synthesis system," in Proceedings of INTERSPEECH, 2019, pp. 2588–2592.

[6]H. Tamaru, Y. Saito, S. Takamichi, T. Koriyama, and H. Saruwatari, "Generative Moment Matching Network-based Random Modulation Post-filter for DNN-based Singing Voice Synthesis and Neural Double-tracking," 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, 2019, pp. 7070-7074.

[7]S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," Proceedings of ICML 2016, vol. 48, pp. 1060–1069, 20–22 Jun 2016.

[8]A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[9]S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," arXiv preprint arXiv:1703.09452, 2017.

[10]L.C. Yang, S.Y. Chou, and Y.H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," arXiv preprint arXiv:1703.10847, 2017.

[11]S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," in Neural computation, 1997, pp. 1735-1780.

[12]T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 4396-4405.

[13]P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 5967-5976.

[14]J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2242-2251.

[15]H. Zhang et al., "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 5908-5916.

[16]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," arXiv preprint arXiv:1406.2661, 2014.

[17]M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," arXiv preprint arXiv:1701.07875, 2017.

[18]M. Mirza, S. Osindero, "Conditional Generative Adversarial Nets," arXiv preprint arXiv:1411.1784, 2014.

[19]A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., Montreal, Que., 2005, pp. 2047-2052 vol. 4.

[20]Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," in Shape, Contour, and Grouping in Computer Vision, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 1999, vol 1681.

[21]X. Shi, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, and W.C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in NIPS, 2015.

[22]A.v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," arXiv preprint arXiv:1609.03499, 2016.

[23]O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, Lecture Notes in Computer Science, vol 9351.

[24]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," 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Kaohsiung, 2013, pp. 1-9.

[25]R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing, Victoria, BC, Canada, 1993, pp. 125-128 vol.1.