Singer to Singer Conversion Based on Generative Adversarial Network

Victor Maurin
Department of
Mechanical Engineering
Stanford University
vicmau@stanford.edu

Yimeng Qin
Department of
Mechanical Engineering
Stanford University
yimengq@stanford.edu

Shanlin Chen
Department of
Mechanical Engineering
Stanford University
shanlinc@stanford.edu

Abstract

In this project, we first leveraged the paper using Deep U-Net convolutional neural network to separate the vocal audio from the background instrument sound. With this method, we successfully extract the 7 vocal data from Michael Bublé, Adiana Grande and Ed Sheeran. The dataset is then input into the CycleGAN-VC2 algorithm to realize the singing style transfer. The choice of dataset and hyperparameter is critical to synthesis results. Then, the vocal data is combined back with the instrumental background to synthesize the final results.

1 Introduction

Voice conversion (VC) refers to digitally modifying one's (source) speech so that it appears to be spoken by another one (target). It is a significant aspect of Artificial Intelligence as its broad application area touches on information security, entertainment, rehabilitation, increasing productivity, and many other topics [1]. In recent years, Voice Conversion techniques has been greatly advanced by developments in Deep Neural Network (DNN) [2]. In this project, we focus on applying Deep Learning techniques, specifically the use of Generative Adversarial Networks (GANs) in speech conversion and the use of Convolution Neural Networks (CNN) in human voice separation, to tackle the problem of singer-to-singer conversion. Our algorithm includes a Convolution U-Net model for human voice separation and a CycleGAN model for singing voice conversion so that the input would be a song sung by the source signer, and the output will be the same song sung by the target singer. Our model demonstrates the effectiveness of transfer learning from a plain-speech converting task to singing-voice converting and provides a creative method of generating deep-faked songs.

2 Related work

At the early stage of Voice Conversion, people focused on using parallel training to match the linguistic features. For example, Gaussian Mixture Model (GMM)[3], non-negative matrix factorization (NMF)[4]. However, this technique requires a high quality parallel training database and usually results with an over-smoothing spectrum, which yields artificial feeling [5]. DNN, on the other hand, due to the nature of learning from large datasets, is able to provide a more natural conversion. At the same time, it frees us from understanding the intermediate speech representation[2]. However, a conventional DNN failed to capture the temporal behavior of speech[6]. CycleGAN(Generative Adversarial Network), proposed by Kaneko and Kameoka, demonstrated its benefit of training an unparalleled dataset by capturing a set of audio features from their own domain [7].

CS230: Deep Learning, Spring 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Features

3.1 Description of Training Set

To create the data set, we have selected two singers that have very different voice features: Ariana Grande and Michael Bublé. After preprocessing (refer to Section X), we totally have three songs from Michael Bublé ("Forever Now", "Have yourself a merry little Christmas", and "Home") and three songs from Ariana Grande ("7 Rings", "Last Christmas", and "One Last Time"). Each vocal signal is chopped into 25 sub-data, where each consists of 6s audio files. After cross validation, we found that "Forever Now" and "One Last Time" generated the best results, which is described in the later section. To compare the inter/intra gender factor, we applied the vocal remover algorithm to extract 34 sub-vocal data from Michael Bublé-"Forever Now" and Ed Sheeran "Perfect".

4 Methods

4.1 Voice Separation

To obtain the high quality dataset, we leverage the vocal separation algorithm ("vocal remover") based on the deep U-net convolutional neural network [8] [9]. The vocal remover algorithm is able to clearly separate a song into the singer's vocal part and the instrument's part. With this algorithm, we successfully extract the vocal audio from 6 selected songs (3 songs from Ariana Grande and 3 songs from Michael Bublé).

The algorithm first applied short time fourier transform to (STFT) the raw song (WAV format) to convert the raw data into the multiple spectrograms. Similar to the U-net architecture, the spectrograms are fed into the neural network (NN) [Figure.1]. In the encoding stage, a stack of convolutional layers are applied to shrink the size of the data but increase the channel size. Then in the decoding stage, upsampling layers are used to recover the audio to the original size.

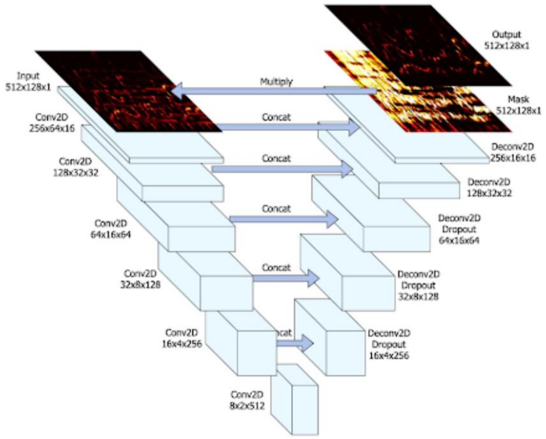

Figure 1: Vocal Removal U-net Convolutional Layer Algorithm [9]

The loss function of the voice separation algorithm can be defined as:

$$\mathcal{L}(X, Y_j) = ||f(X, \theta_j) \odot X - Y_j||_{norm,1} \tag{1}$$

Where, the subscript $j = v$ (vocal) or i (instrument). X, Y and θ notates the original audio signal, target audio output, and mask parameter generated by the model. And $f(X, \theta)$ represents the output of the neural network.

4.2 Voice Conversion

In this task, We build on the work of Kaneko et al. (2019), who implemented CycleGAN-VC2, a CycleGAN-based voice conversion model for non-parallel speaker pairs [10]. Based on the architecture of this algorithm coded using PyTorch [11], we develop both an inter-gender singing voice conversion model between Ariana Grande and Michael Bublé and an intra-gender conversion between Michael Bublé and Ed Sheeran. After the conversion, the converted vocal is merged with the separated background music of the source to finish the song conversion.

Our dataset is made of audio files. To process them and encode them, we are using three libraries that allow us to effectuate the calculations needed: pyWORLD, Librosa, and Soundfile. After having encoded our audio files, we feed our CycleGAN model with them.

For the CycleGAN model, the generator is built on the 2-1-2D CNN structure. In this generator network, 2D convolution layers are used for downsampling and upsampling; then, a 1x1 convolution layer is added to adjust the dimension. The 6 1D residual blocks stacked in the middle are mainly responsible for the voice feature conversion process. The dimensions of each convolution layer are stated in the following Figure.2.Such combination benefits the voice conversion as the 2D CNN preserves the original structures as it converts features, while the 1D CNN is more feasible for capturing dynamical change [10]. The discriminator is built on the convolutional *Patch*GAN classifier [10]. The *Patch*GAN discriminator uses 2D CNN to first downsample the input sound file and uses a convolution last layer to determine the fakeness at the scale of the patch.

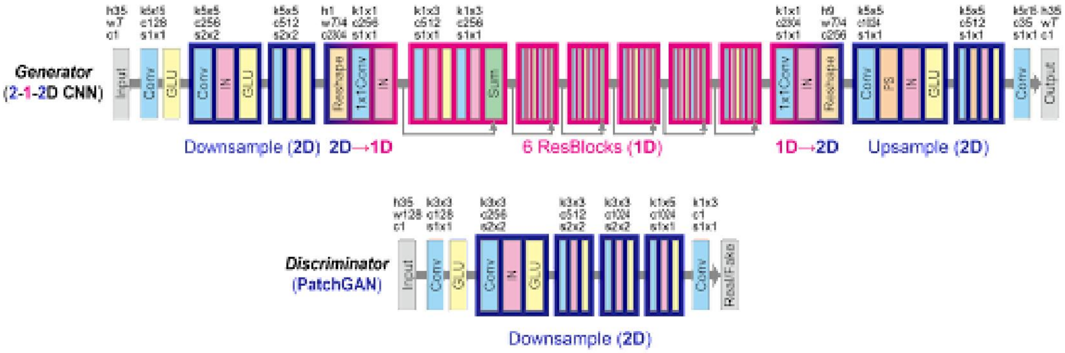

Figure 2: Network architectures of generator and discriminator. h, w, and c denote height, width, and the number of channels; k and s denote kernel size and stride size in convolution layers. IN, GLU, and PS indicate instance normalization, gated linear unit, and pixel shuffler.[10]

The objective function of the above CycleGAN model includes three parts: a two-step adversarial loss, a cycle-consistency loss, and an identity-mapping loss. Denoting features of the source singer as X and that of the target singer as Y, the objective function can be expressed as

$$G^* = \arg\min_G \arg\max_D \mathcal{L}_{overall} \tag{2}$$

in which

$$\begin{aligned}\mathcal{L}_{overall} = \mathcal{L}_{adv}(G_{X\to Y}, D_Y) + \mathcal{L}_{adv}(G_{Y\to X}, D_X) + \mathcal{L}_{adv2}(G_{X\to Y}, G_{Y\to X}, D'_Y) + \\ \mathcal{L}_{adv2}(G_{X\to Y}, G_{Y\to X}, D'_X) + \lambda_{cyc}\mathcal{L}_{cyc}(G_{X\to Y}, G_{Y\to X}) + \lambda_{id}\mathcal{L}_{id}(G_{X\to Y}, G_{Y\to X})\end{aligned} \tag{3}$$

In this expression, λ_{cyc} and λ_{id} are the trade-off parameters between the two corresponding losses.

Two-Step Adversarial Loss consists of a common one-step adversarial loss that measures whether a converted feature, such as $G_{X\to Y}(x)$, is indistinguishable from its target Y,

$$\mathcal{L}_{adv}(G_{X\to Y}, D_Y) = \mathbb{E}_{y\sim P_Y(y)}[logD_Y(y)] + \mathbb{E}_{x\sim P_X(x)}[log(1 - D_Y(G_{X\to Y}(x)))] \tag{4}$$

and an additional discriminator D' loss which measures whether the circularly converted feature $(G_{X\to Y}(x), G_{Y\to X}(y))$ is indistinguishable from the original source X [10].

$$\begin{aligned}\mathcal{L}_{adv2}(G_{X\to Y}, G_{Y\to X}, D'_X) = &\mathbb{E}_{x\sim P_X(x)}[logD'_X(x)] \\ &+ \mathbb{E}_{x\sim P_X(x)}[log(1 - D'_X(G_{Y\to X}(G_{X\to Y}(x))))]\end{aligned} \tag{5}$$

Cycle-consistency loss is implemented to preserve the structural consistency between features of the Generator's input and output.

$$\begin{aligned}\mathcal{L}_{cyc}(G_{X\to Y}, G_{Y\to X}) = &\mathbb{E}_{x\sim P_X(x)}[||G_{Y\to X}(G_{X\to Y}(x)) - x||_{norm,1}] \\ &+ \mathbb{E}_{y\sim P_Y(y)}[||G_{X\to Y}(G_{Y\to X}(y)) - y||_{norm,1}]\end{aligned} \tag{6}$$

Since both forward-inverse and inverse-forward mappings are simultaneously learned, this loss function will encourage the model to develop an optimal (X, Y) pair through circular conversion [12].

Identity-mapping loss

$$\begin{aligned}\mathcal{L}_{id}(G_{X\to Y}, G_{Y\to X}) = &\mathbb{E}_{y\sim P_Y(y)}[||G_{X\to Y}(y) - y||_{norm,1}] \\ &+ \mathbb{E}_{x\sim P_X(x)}[||G_{Y\to X}(x) - x||_{norm,1}]\end{aligned} \tag{7}$$

is implemented to further preserve linguistic information.

5 Experiments/Results/Discussion

The authors suggested some hyperparameters to use. For both male-to-male and female-to-male trainings, we used the learning rates: $1e^{-4}$ for the discriminator, and $2e^{-4}$ for the generator, as well as the betas of the Adam optimizer: $\beta_1 = 0.5$, and $\beta_2 = 0.999$. We also used a linear learning rate decay, which starts at 20,000 iterations, and which for each iteration higher than 20,000 subtracts the value of the learning rate by (α/200,000), being either the learning rate of the generator or of the discriminator. Also, the kernel size, the padding, and the stride of every convolutional layer were chosen based on the suggestion of the authors [10]. Finally, the constant values of the generator and discriminator loss functions were chosen according to the paper: $\lambda_{id} = 5$, and $\lambda_{cyc} = 10$.

For the other parameters, we have tried to optimize them to tend to better results. As a reminder, the CycleGAN-VC2 has been designed for voice-to-voice conversion based on speeches, and not on songs. This difference is very important, as the voices of singers are much more dynamic and have much more features than voices from speeches. This is therefore harder for the generator to identify and match the correct voices, when generating certain parts of songs (for instance, the generator was not able to generate a proper voice for the lyrics "One last time" sung by Ariana Grande, in any of our training sessions. See the files "OneLastTime_VocalsFrom36.0to42.0" from the "Final Project Audio Files" link in Appendix). The parameters we tried to optimize were: The mini-batch size, the number of residual layers, and the number of epochs we trained for.

For the mini-batch size, the author suggested the use of 1.[10] However, we decided to try several other values to see what impact it would have on the training. The values used were all powers of 2, and were ranging from 1 to 32. The use of higher values was motivated by a faster decrease of the generator loss function with less oscillations, which gave the expected result. However, the quality of the generated audio files was decreasing as we increased the mini-batch size (more robotization of the voice, the voices features were not identified as well, etc...). This motivated us to change back the mini-batch size to 1 which gave better results for higher generator loss function values. In fact, it seems that with a mini-batch size of 1, we encourage the network to learn the features of the voices independently from the features of the audio files that the network is using to train on at each epoch. Indeed, we obtained much better and much clearer voices this way [See the "Final Project Audio Files" link in Appendix]. It also led us to see that decreasing the generator loss function was becoming a bad metric at some point of the training, as there was not a clear correlation between the loss function value and the quality of the songs when the loss function started to converge (values around 5 or 6). This analysis can also be transposed to the number of epochs. Indeed, we also tried to train for a very high number of epochs (up to 20,000). However, we did not observe a huge improvement for a number of epochs ranging from 5000 to 20,000. The generation was sometimes better, sometimes not, than the previous generations. You can see how the loss functions look like for one of our trainings on figure 3. The x-axis being the number of iterations and not the number of epochs. On this figure, the quality of the audio was increasing as the Generator loss function decreased down to a value of around 6. Then, the quality of the audio became decorelated with the value of the loss functions. This is something we observed for both intragender and intergender conversion.

The parameter which optimization was certainly the most successful was the number of residual layers. In fact, the authors explained that the residual layers were helping to capture the dynamical change of the voices [10]. Since singer voices are really dynamic, as explained before, we decided to increase the number of layers from 6 to 8 for the female-to-male conversion to see if we could improve the quality, which led to an overall improvement of the voices. For male-to-male conversion, we decided to keep the number of layers to 6 due to the lack of time. In the female-to-male case, It

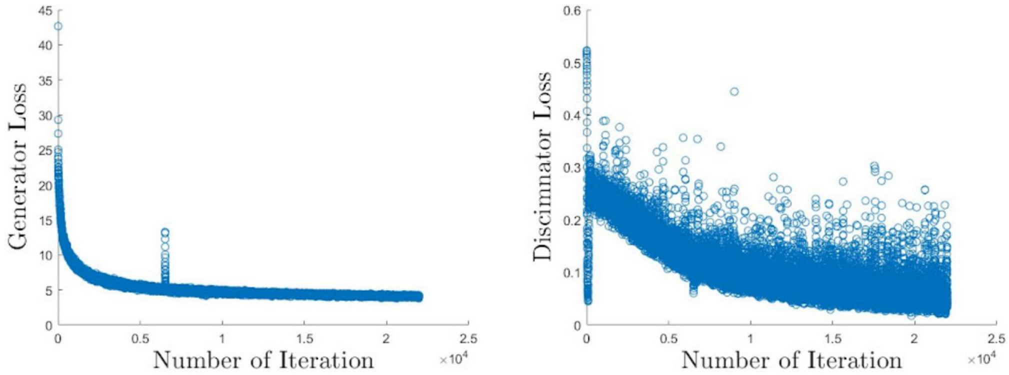

Figure 3: a.) Generator Loss b.) Discriminator Loss for Ariana Grande - Michael Bublé conversion

seems that the CycleGAN model was able to better capture the voices features of each singer when generating the songs with this number of hidden layers. This is definitely something we would have tried on the male-to-male conversion if we had more time.

Overall, qualitatively, converted vocal in male-to-male conversion sounds much more natural than that of the female-to-male conversion [See the "Final Project Audio Files" link in Appendix]. That was expectable since the voices of Ed Sheeran and Michael Bublé are really similar. For Ariana Grande and Michael Bublé, the conversion is promising but still needs some improvement. It seems that for some pitches, the network is not able to perfectly convert the voices, which results in some wrong notes compared to the original songs or simply some missing lyrics. Intragender conversion also seems to give better results in the author paper than intergender conversion. However, one big conclusion we can make out of this project is that song to song conversion is really different from speech to speech conversion, as songs are much more complex (the pitches are changing highly, the voices are very dynamic, etc. . .)

6 Conclusion/Future Work

To conclude, we have seen that the CycleGAN-VC2 algorithm allows a very efficient and pretty fast voice conversion with unparalleled training sets. The basic algorithm is capable of identifying the overall pitch and features of each singer's voice, and generating them (with some robotization) with high fidelity. Since the voices are really dynamic, adding residual layers to the generator seemed to have increased the voice quality, and decreased the robotization. One downside of this was that the training was longer. It is also interesting to see the intragender (Michael Bublé - Ed Sheeran) generation gave much better than the the intergender (Michael Bublé - Ariana Grande) generation. Those difficulties have led us to one important piece of information: Singer to singer conversion is much more difficult than a simple voice to voice conversion. Conversion is certainly harder for the generator, since songs have much more features than speeches. The pitch can vary a lot (We can really hear this in Ariana Grande's songs), there are also many more features that the generator has to learn, etc. . . Overall, songs for which the pitch was varying a lot were definitely the hardest for the generator to generate.

If we had more time, we would certainly continue to optimize the number of residual layers. We would also refine the padding, the stride, and the kernel size to see what effect they would have on the quality of the songs. Finally, we would certainly try other artists to continue to understand how the algorithm works in order to refine it.

7 Contributions

All authors contributed equally to the project. Victor Maurin contributed mainly to the implementation of the CycleGAN-VC2 model and the hyperparameter tuning. Yimeng Qin focused on the dataset preparation, implemented the voice separation and synthesis, as well as contributed to the hyperparameter tuning. Shanlin Chen was responsible for the model training on AWS, as well as contributed largely to the submittables. All three team members have participated in literature review, model training, and writing submittables. A special thank to our TA, Allan Zhou, for his advice on model selection, experiment strategies, and hyperparameter optimization.

References

[1]Sisman, Berrak, et al. "An Overview of Voice Conversion and Its Challenges: From Statistical Modeling to Deep Learning." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, 2021, pp. 132–157., https://doi.org/10.1109/taslp.2020.3038524.

[2]"Voice Conversion Challenge 2020." *Voice Conversion Challenge 2020*, http://www.vc-challenge.org/#:~:text=Voice%20conversion%20(VC)%20refers%20to,the%20original%20speaker%20(source).

[3] Toda, Tomoki, et al. "Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, 2007, pp. 2222–2235., https://doi.org/10.1109/tasl.2007.907344.

[4]Luan, Yi, et al. "Semi-Supervised Noise Dictionary Adaptation for Exemplar-Based Noise Robust Speech Recognition." *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, https://doi.org/10.1109/icassp.2014.6853897.

[5]Kaneko, Takuhiro, et al. "Sequence-to-Sequence Voice Conversion with Similarity Metric Learned Using Generative Adversarial Networks." *Interspeech 2017*, 2017, https://doi.org/10.21437/interspeech.2017-970.

[6] Daher, Rema, et al. "Change Your Singer: A Transfer Learning Generative Adversarial Framework for Song to Song Conversion." *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, https://doi.org/10.1109/ijcnn48605.2020.9206878.

[7] Brownlee, Jason. "How to Develop a Cyclegan for Image-to-Image Translation with Keras." *Machine Learning Mastery*, 1 Sept. 2020, https://machinelearningmastery.com/cyclegan-tutorial-with-keras/#:~:text=The%20benefit%20of%20the%20CycleGAN,the%20day%20and%20at%20night.

[8] Tsurumeso. "Tsurumeso/Vocal-Remover: Vocal Remover Using Deep Neural Networks." GitHub, https://github.com/tsurumeso/vocal-remover

[9] Jansson, Andreas et al. "Singing Voice Separation with Deep U-Net Convolutional Networks." *ISMIR (2017)*. URL: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/171_Paper.pdf

[10]T. Kaneko, H. Kameoka, K. Tanaka and N. Hojo, "Cyclegan-VC2: Improved Cyclegan-based Non-parallel Voice Conversion," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6820-6824, https://doi.org/10.1109/ICASSP.2019.8682897.

[11] jackaduma. "jackaduma/CycleGAN-VC2." GitHub, https://github.com/jackaduma/CycleGAN-VC2

[12]T. Kaneko, H, Takuhiro and H. Kameoka. "Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks." *arXiv:1711.11293, Nov.* 2017 (EUSIPCO, 2018) https://doi.org/10.48550/arxiv.1711.11293

8 Appendix

8.1 Final Project Audio Files

Examples of training data and converted vocal files, as well as the synthesised final converted song for both male-to-male singer conversion and female-to-male singer conversion: https://drive.google.com/drive/folders/1jhXc_Dzn3-A8HPGi6kjDq0s7eM0319rc?usp=sharing

8.2 Codes

"yimengq/CS230." GitHub, https://github.com/yimengq/CS230