Analysis and Process of Music Signals to Generate Two-Dimensional Tabular Data and a New Music

Oakyoung Han[1] and Jaehyoun Kim[2, *]

Abstract: The processing of sound signals is significantly improved recently. Technique for sound signal processing focusing on music beyond speech area is getting attention due to the development of deep learning techniques. This study is for analysis and process of music signals to generate tow-dimensional tabular data and a new music. For analysis and process part, we represented normalized waveforms for each of input data via frequency domain signals. Then we looked into shorted segment to see the difference wave pattern for different singers. Fourier transform is applied to get spectrogram of the music signals. Filterbank is applied to represent the spectrogram based on the human ear instead of the distance on the frequency dimension, and the final spectrogram has been plotted by Mel scale. For generating part, we created two-dimensional tabular data for data manipulation. With the 2D data, any kind of analysis can be done since it has digit values for the music signals. Then, we generated a new music by applying LSTM toward the song audience preferred more. As the result, it has been proved that the created music showed the similar waveforms with the original music. This study made a step forward for music signal processing. If this study expands further, it can find the pattern that listeners like so music can be generated within favorite singer's voice in the way that the listener prefers.

Keywords: Frequency domain signals, spectrogram, fourier transform, filterbank, long short-term memory.

1 Introduction

In recent years, deep learning techniques have been successfully applied to many signal processing tasks. Especially in speech signal processing, neural networks with deep structures have been introduced to the speech generation techniques [Ling, Ai, Gu et al. (2018)]. Google Duplex uses WaveNet, which is a Deep Neural Network (DNN) model to generate raw audio waveforms. This Text-To-Speech (TTS) system achieves high performance in terms of speech conversion, and provides great improvements in reproducing natural speech [Lieto, Moro, Devoti et al. (2019)].

[1] Sungkyun Software Education Institute, Sungkyunkwan University, Seoul, 03063, Korea.

[2] Department of Computer Education, Sungkyunkwan University, Seoul, 03063, Korea.

[*] Corresponding Author: Jaehyoun Kim. Email: jaekim@skku.edu.

Received: 06 December 2019; Accepted: 06 February 2020.

CMC. doi:10.32604/cmc.2020.09362

The next challenge for sound signal processing is probably music because music extends beyond speech in human hearing range as shown in Fig.1.

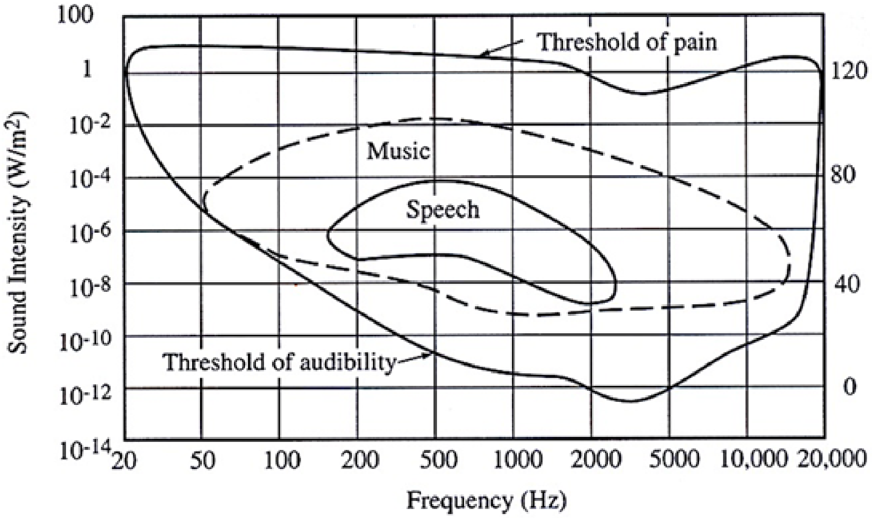


Figure 1: Human hearing range [Dull, Metcalfe and Brooks (1995)]

Generative adversarial network (GAN) receives attention for image generation [Hong, Kim and Kang (2019)], and also is proved to be able to generate symbolic music [Guan, Yu and Yang (2019)] since GAN is very useful for digital signal handling [Tu, Lin, Wang et al. (2018)]. Tab. 1 shows existing music generation system [Li, Jang and Sung (2019)]. Continuous recurrent neural networks based on the GAN (C-RNN-GAN) is a GAN model that is configured as a long short-term memory (LSTM) neural network to generate note-based melodies [Mogren (2019)].

Table 1: Music generation systems

	C-RNN-GAN	MidiNet	Muse GAN	Enhanced GAN
Model	GAN	Conditional GAN	GAN	GAN

Neural Network	LSTM;
	Bi-LSTM
		CNN	CNN	LSTM;
				Bi-LSTM;
				CNN

MidiNet is a conditional GAN model configured by a convolutional neural network (CNN) that generates a bar-based melody based on a given chord. However, it cannot distinguish between long notes and continuous tones with the same pitch when encoding is performed [Yang, Chou and Yang (2017)]. MuseGAN is similar in structure to MidiNet, as both use GAN and CNN to generate music. However, to compensate for the lack of continuity caused by using CNNs to generate bars, it uses two subnetworks for generator section which are a bar generator and a temporal structure generator. The time continuity of the bar sequence generated by the bar generator is handled by the temporal structure generator [Dong, Hsiao, Yang et al. (2018)]. Enhanced GAN generates a melody with two discriminators based on RNN and CNN to ensure the correlation between bars and the rationality of the node structure.

To select input music for the study, famous tunes in classical music have been reviewed and Tab. 2 lists the result of the rank. The most famous music was listed as “Eine kleine Nachtmusik” composed by Mozart, and the type of instruments for the music is the strings. The next famous music was “Für Elise” composed by Beethoven and piano is the type of the music. The third famous music was the opera “O mio babbino caro” composed by Puccini [Rizzi (2018)].

Table 2: Famous tunes in classical music

Rank	Composer	Title	Type
1	Mozart	Eine kleine Nachtmusik	Strings
2	Beethoven	Für Elise	Piano
3	Puccini	O mio babbino caro	Opera

Opera was selected as an input music type since the opera only includes vocal sound that can generate the difference by different singers. Three different performances of “O mio babbino caro” were adopted for the music analysis. Three opera singers for “O mio babbino caro” are André Rieu, Jackie Evancho and Maria Callas. André Rieu and Jackie Evancho are young girls while Maria Callas is adult. Since there is an obvious age gap, the analysis result must be distinguished from each other.

The concept of the steps of the study are suggested in Fig. 2. The first goal is to draw waveform of the input music file. The frequency domain of music signals is represented as normalized waveforms. The next step is to figure out the spectrogram from the waveforms by applying Fourier transform. Then filterbank in Mel scale has been studied to get the final spectrogram which fit into the human hearing scale. For the next step, two different research carried out to generate two-dimensional tabular data and to generate a new music file. With the two-dimensional tabular data, data manipulation can be processed for the necessary purpose. On the other hand, the computer can generate a new music based on the preferred music by applying LSTM neural network.

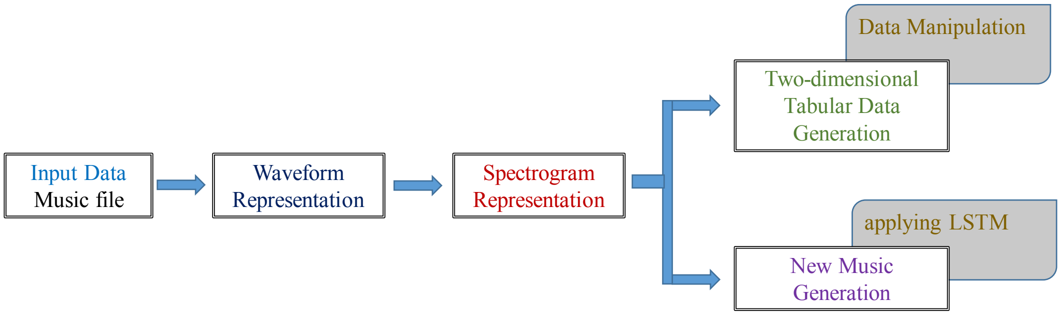
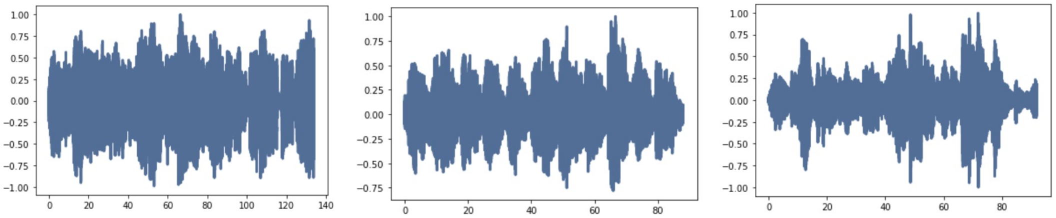


Figure 2: Process of the study

2 Analysis and process

2.1 Frequency domain signals

ThinkDSP, the digital signal processing module in Python has been used to generate waveforms for wave files of the three opera data. To get the waveforms, the impulse response needs to be computed by multiplying spectrum of the impulse and the filter, and then converting the result from a spectrum to a wave [Downey (2014)]. The normalized wave files within time domain for the three selected data are shown in Fig. 3. The normalized waveforms of the youngest opera singer Andre showed the average range of the amplitude between 0.5 and -0.5, and it showed the widest in amplitude among three. The normalized waveforms of the second youngest opera singer Jackie showed the average range of the amplitude between 0.4 and -0.4, and it showed the narrower in amplitude than Andre. The normalized waveforms of the oldest opera singer Maria showed the average range of the amplitude between 0.25 and -0.25, but it showed the various amplitude values compare with other singers.

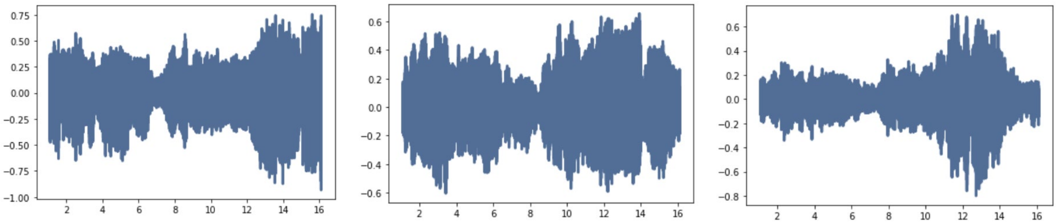


(1) Andre (2) Jackie (3) Maria

Figure 3: Normalized waveforms

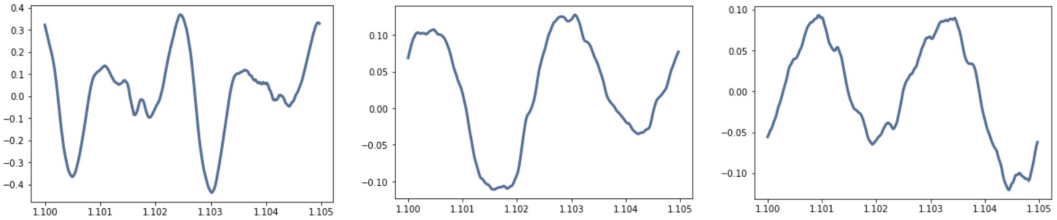
For more accurate waveforms, a segment is selected at the start position 1.1 for duration 15 as shown in Fig. 4. The result of Jackie cannot be characterized, but the clarity of Andrew's vocalization and emotionally deep Mary's amplitude can be seen in the segment display. Predicting the amplitude of listeners' favorite opera through machine learning and recommending other opera music with similar wave amplitude can be a useful application of audio signals.

For magnifications of the waveform, an even shorter segment is displayed in Fig. 5 with the start position of 1.1 for duration of 0.005. The difference in each singer can be clearly identified with the waveform pattern. Andre curves the waveforms vividly without shaking, while Maria expresses the most wavering waveforms.



(1) Andre (2) Jackie (3) Maria

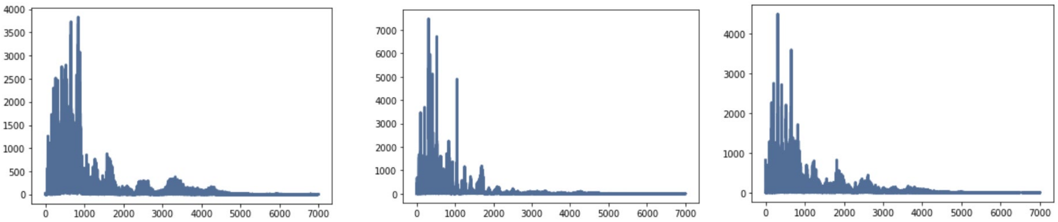
Figure 4: Segment with a constant pitch



(1) Andre (2) Jackie (3) Maria

Figure 5: Waveform with the shorter segment

The final spectrum for the segment with the value of high as 7000 are represented in Fig. 6. Since it has lots of frequency components, zooming in on the fundamental and dominant frequencies is performed in Fig. 7.

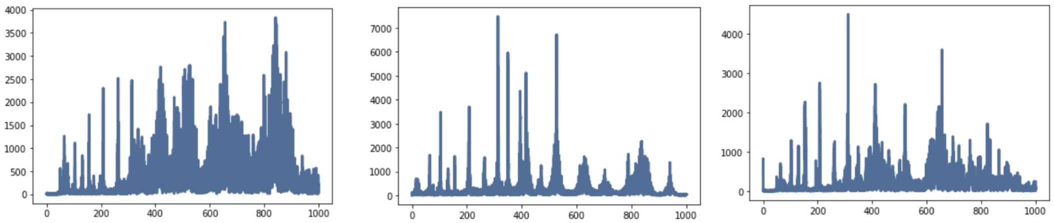


(1) Andre (2) Jackie (3) Maria

Figure 6: Spectrum for the segment

2.2 Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. Spectrogram image representation efficiently extracts audio variations present in the music signal [Birajdar and Patil (2019)]. A spectrogram is created from a time-domain signal that is represented by normalized waveform in one of two ways: calculated from the time signal using the Fourier transform, or approximated as a filterbank that results from a series of band-pass filters which was the only way before the advent of modern digital signal processing.

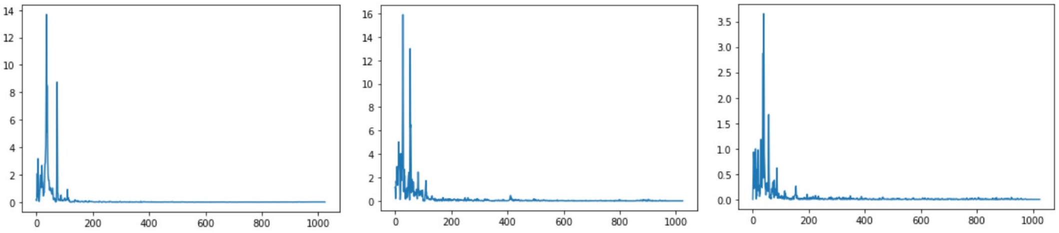


(1) Andre (2) Jackie (3) Maria

Figure 7: The fundamental and dominant frequencies

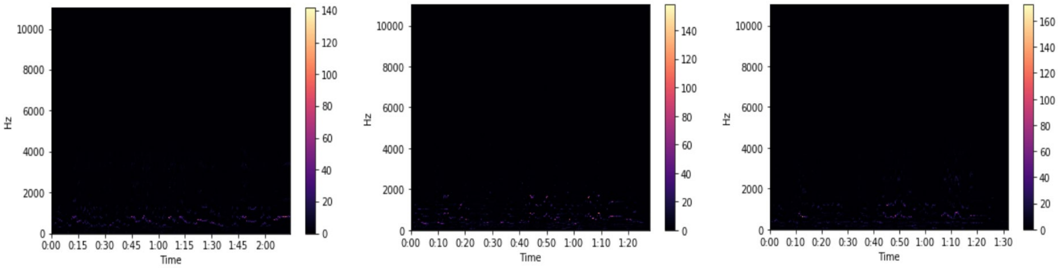
2.2.1 Fourier transform

The Fourier transform (FT) decomposes a function of a signal into its constituent frequencies. The term Fourier transform refers to the frequency domain representation. The discrete Fourier transform (DFT) takes a signal and produces its spectrum while the spectrum is the set of sinusoids that add up to produce the signal. The Fast Fourier transform (FFT) is an efficient way to compute the DFT [Takahashi (2019)]. The Short-time Fourier transform (STFT) represents a signal in the time-frequency domain by computing DFT over short overlapping windows. To process STFT, LibROSA which is a python package for music and audio analysis is used. It provides the building blocks necessary to create music information retrieval systems [McFee, Raffel, Liang et al. (2015)]. LibROSA is used to get spectrogram of the same three opera sound. The method librosa.effects.trim() has been applied with proper parameters to trim leading and trailing silence from an audio signal. After processing librosa.effects.trim(), librosa.stft() method is executed. The results of the STFT with 2048 for the default frame are show in Fig. 8. Spectrograms with 512 frames of the signal are displayed in Fig. 9. The y-axis of the basic spectrogram is linear, and the perception of the image is very low. The improved spectrograms by cutting down the range of frequency from 10000 to 8192 and by using log value for the y-axis are shown in Fig. 10. The equation Eq. (1) is to convert the time domain waveform into the frequency domain waveform using FFT.



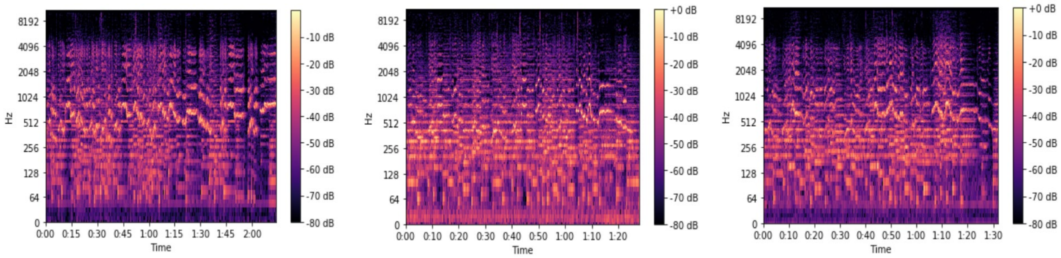
(1) Andre (2) Jackie (3) Maria

Figure 8: Visualization of STFT



(1) Andre (2) Jackie (3) Maria

Figure 9: Basic spectrogram



(1) Andre (2) Jackie (3) Maria

Figure 10: Improved visualization of spectrogram

$$\boldsymbol{F(k) = \int_{-\infty}^{\infty} f(x)e^{-2\pi ikx}dx} \quad (1)$$

There is difference among three spectrograms. It may help to check the certain pattern that a user preferred, so a recommendation for other music with the similar pattern can be provided by analyzing the music signal.

2.2.2 Filterbank

For producing high fidelity audio and the reduction of information loss, MelNet can be applied [Vasquez and Lewis (2019)]. However, in this study, the Mel scale is applied for the result of some non-linear transformation of the frequency scale since Mel scale filterbank features can extract information from consecutive frames [Lim, Lee, Park et al. (2018)]. This Mel scale is constructed such that sounds of equal distance from each other on the Mel scale. The graph of Mel scale is shown in Fig. 11 while the equation for Mel scale of the frequency is shown in Eq. (2), and it can be converted as the equation Eq. (3). The Mel-spaced filterbank for 26 filters is shown in Fig. 12.

$$M(f) = 1125 \log (1 + f / 700) \quad (2)$$

$$M^{-1}(m) = 700 (\exp(m/1125) - 1) \quad (3)$$

In Fig. 10, the difference between 500 and 1000 Hz is obvious, whereas the difference between 7500 and 8000 Hz is barely noticeable. For better spectrogram, filterbank can be applied as Fig. 13.

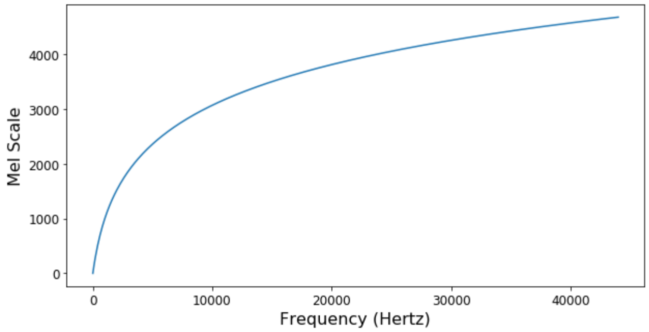


Figure 11: Converting hertz to Mel scale

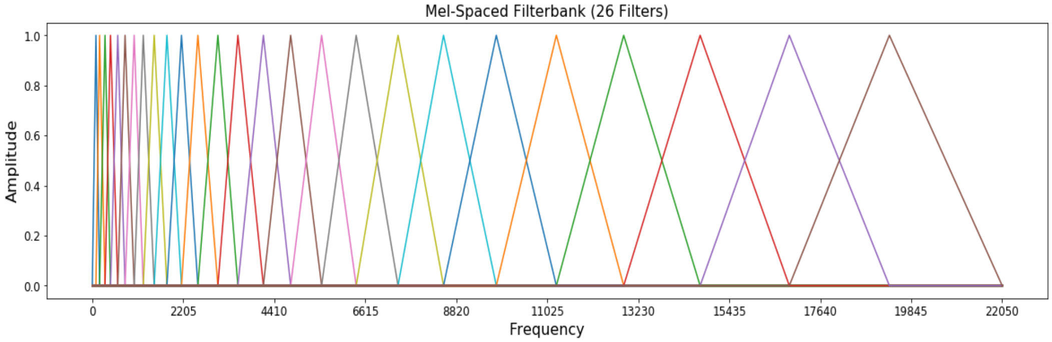
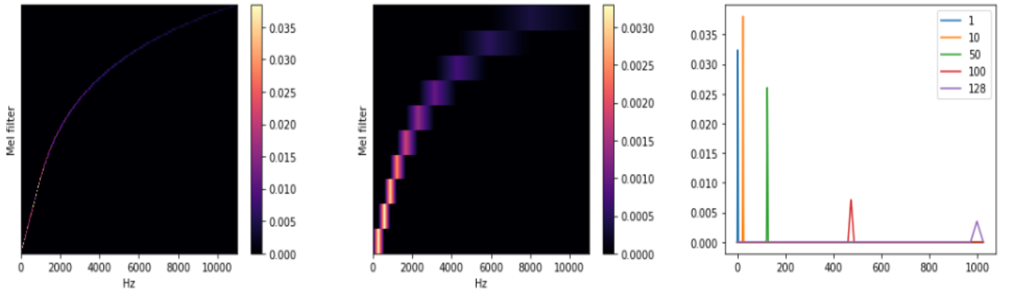


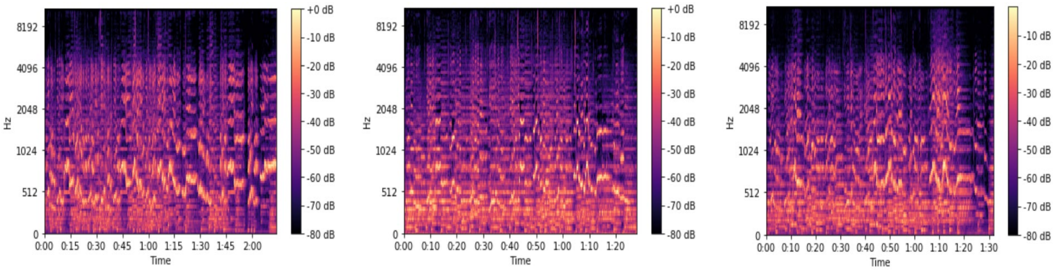
Figure 12: Filterbank of 26 filters



filterbank for converting from Hz to mels 10 mels only for better visualization Plotting some triangular filters separately

Figure 13: Mel scale

The final spectrogram of three different music signals is shown in Fig. 14. The value of this analysis is that analysis is focused on not the distance on the frequency dimension, but distance as it is heard by the human ear by applying filterbank in Mel scale.



(1) Andre (2) Jackie (3) Maria

Figure 14: The final spectrogram using Mel scale

3 Generation

3.1 Two-dimensional tabular data generation

A matrix profile of music in quadratic space has been researched for music analysis and exploration [Silva, Yeh, Batista et al. (2019)]. In this study, we tried to generate tow-dimensional tabular data from spectrogram. LibROSA python package is a tool for music information retrieval system [Raguraman, Mohan and Vijayan (2019)]. With the final spectrogram, an array can be created by executing a method librosa.feature.melspectrogram(). It generated 128 by 5771 array data, then a method librosa.power_to_db() is executed to generate a database based on the sound signal of human ear. After turning the spectrogram into database, pandas for python system is imported to create data frame. The Python library pandas provides easy-to-use data structures and data analysis tools. In particular, it can be used to easily read and modify CSV files. When reading a CSV file, the content is stored in a two-dimensional tabular data structure with labeled axes as rows and columns called DataFrame.

For example, spectrogram and waveform representations of the audio signal can be shown as 4 Steps like in Fig. 15. The waveform spans nearly 100,000 timesteps whereas the temporal axis of the spectrogram spans roughly 400. Complex structure is nested within the temporal axis of the waveform at various timescales, whereas the spectrogram has structure which is smoothly spread across the time-frequency plane.

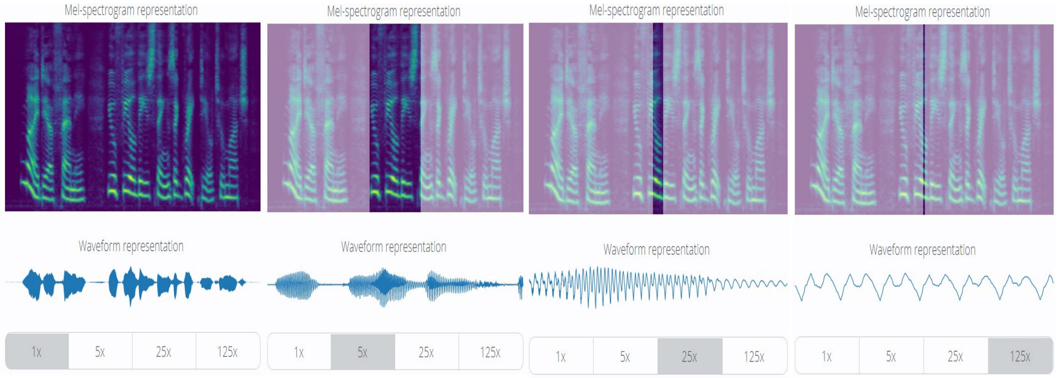


Figure 15: Spectrogram and waveform representations

To match spectrogram and waveform data is available with the two-dimensional tabular data and it is shown in Fig. 16. It means actual matrix values for the music signal is possible to access due to this study.

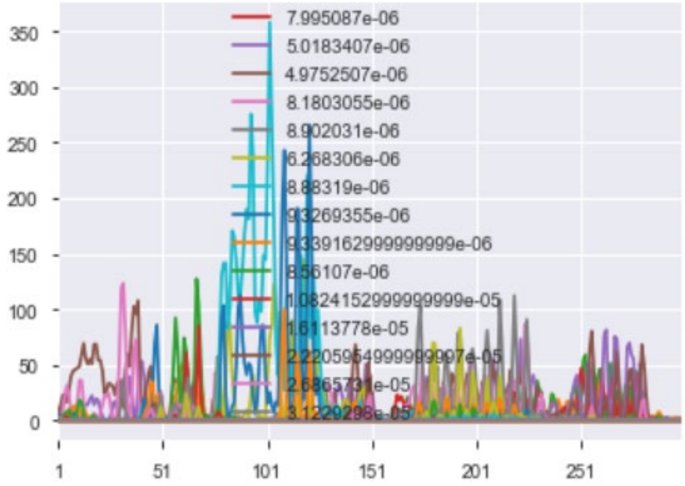
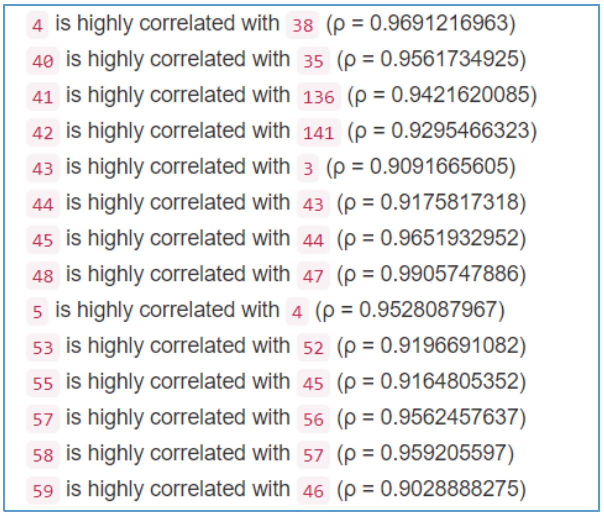


Figure 16: 2D data value plot via timestep

In addition, sample manipulation within dataset can be checking the correlation of data to figure out the pattern in the dataset, and it is shown in Fig. 17. Checking the correlation can be accomplished several research; for example, which data pattern is appropriate for representing sadness or happiness.



```
4 is highly correlated with 38 (ρ = 0.9691216963)
40 is highly correlated with 35 (ρ = 0.9561734925)
41 is highly correlated with 136 (ρ = 0.9421620085)
42 is highly correlated with 141 (ρ = 0.9295466323)
43 is highly correlated with 3 (ρ = 0.9091665605)
44 is highly correlated with 43 (ρ = 0.9175817318)
45 is highly correlated with 44 (ρ = 0.9651932952)
48 is highly correlated with 47 (ρ = 0.9905747886)
5 is highly correlated with 4 (ρ = 0.9528087967)
53 is highly correlated with 52 (ρ = 0.9196691082)
55 is highly correlated with 45 (ρ = 0.9164805352)
57 is highly correlated with 56 (ρ = 0.9562457637)
58 is highly correlated with 57 (ρ = 0.959205597)
59 is highly correlated with 46 (ρ = 0.9028888275)
```

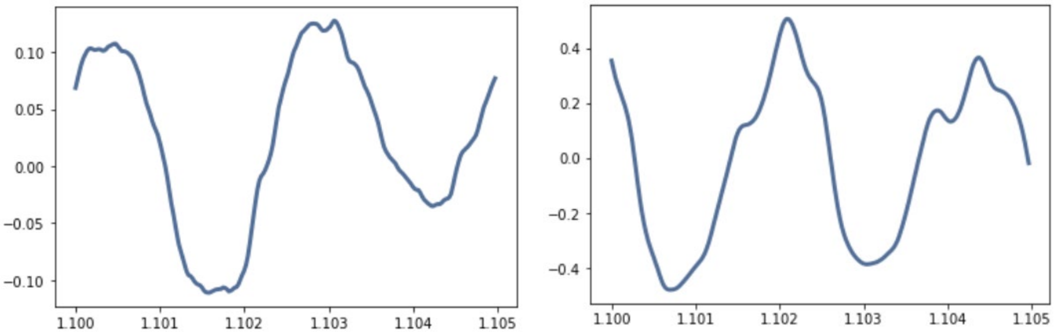
Figure 17: Correlation of dataset

3.2 Music generation

Another research direction of the study is to generate a music using time-domain signals that we have generated. Generating realistic music is one of the spotlight tasks in the field of deep learning. Various models have been suggested and the models used Generative Adversarial Network (GANs), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) to generate music [Kulkarni, Gaikwad, Sugandhi et al. (2019)]. Some researchers are focused on generation of a music with an architecture for the creation of emotionally congruent music using machine learning aided sound synthesis [Williams, Hodge, Gega et al. (2019)]. Other researchers have studied for melody extraction and encoding method to generate a music automatically [Li, Jang and Sung (2019)]. Deep learning architectures are autistic automata which generate music autonomously without human user interaction, far from the objective of interactively assisting musicians to compose and refine music [Jhamtani and Berg-Kirkpatrick (2019)]. Furthermore, GAN is more preferred for music generation. Nevertheless, if the time-domain signals are obtained, LSTM neural network can generate music signals based on the spectrogram. LSTM can useful for many fields since LSTM network is utilized to discover long-term temporal dependencies and final prediction [Arif, Wang, Fei et al. (2019)].

To solve the melody generation problem, it needs to explore the effect of explicit architectural encoding of musical structure via comparing two sequential generative models: LSTM which is a type of RNN and WaveNet which is dilated temporal-CNN [Chen, Zhang, Dubnov et al. (2019)]. Hewahi, AlSaigal and AlJanahi (2019)] have studied to explore the usage of long short-term memory neural network (LSTM-NN) in generating music pieces [Hewahi, AlSaigal and AlJanahi (2019)]. WaveNet is a deep neural network for generating raw audio waveforms. The model is probabilistic and autoregressive, with the predictive distribution for each audio sample. When applied to text-to-speech, it yields state-of-the-art performance, with human listeners rating it as significantly more natural sounding than the best parametric and concatenative systems [Oord, Dieleman, Zen et al. (2016)]. However, it may have some limitation on music which is more expand region on sound.

We chose the spectrogram of Jackie's performance as the sampling data, and generated a new wave file by prediction using LSTM to provide similar chords with Jackie's. The comparison of the original waveform with the waveform of a generated music can be checked in Fig. 18. As you can see, the wave pattern is more like Andre's waveform, clear and straight forward with less vibration, but the waveform is very close to the Jackie's waveform.



Jackie's waveform

Waveform of the created music

Figure 18: Comparisons of the original waveform and created one

4 Conclusion

In this paper, music signal representations have been studied for frequency-domain signals as waveforms and time-domain signals as spectrogram. If we looked into the waveform with a short segment, the difference in wavelength is obvious even if the same song is song by other singers. By applying filterbank, we have proved that spectrogram can be plotted focused on the human hearing. It can find out the reason why people prefer bands with different bandwidths. People prefer different wavelengths and are more satisfied when they listen to their favorite range of music signals.

We successfully generated two-dimensional tabular data from the spectrogram, and it means we can analysis or manipulate the music signal data as we want to study. Also by using LSTM, we created a new music that showed similar wave pattern with the original music. However, the created music was not good enough to enjoy as a classical opera music due to the high pitch.

In the further work, wide usage of the two-dimensional tabular data needs to be researched. In addition, we need to improve the quality of the created music so people can generate any music that they want to listen with the voice they preferred to fully enjoy the music via assistant of artificial intelligent.

Acknowledgement: The authors thank Dr. Jeong at Hansung University for his helpful advice. The authors are also grateful to the very thorough reviewers.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Arif, S.; Wang, J.; Fei, Z.; Hussain, F. (2019): Video representation via fusion of static and motion features applied to human activity recognition. *KSII Transactions on Internet & Information Systems*, vol. 13, no. 7, pp. 3599-3619.

Birajdar, G. K.; Patil, M. D. (2019): Speech and music classification using spectrogram based statistical descriptors and extreme learning machine. *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15141-15168.

Chen, K.; Zhang, W.; Dubnov, S.; Xia, G.; Li, W. (2019): The effect of explicit structure encoding of deep neural networks for symbolic music generation. *2019 International Workshop on Multilayer Music Representation and Processing*, pp. 77-84.

Dong, H. W.; Hsiao, W. Y.; Yang, L. C.; Yang, Y. H. (2018): MuseGAN: multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. *Thirty-Second Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, vol. 32, pp. 34-41.

Downey, A. B. (2014): *Think DSP: Digital Signal Processing in Python*, Version 1.0. 9. Green Tea Press.

Dull, C. E.; Metcalfe, H. C.; Brooks, W. O. (1955): *Modern Physics*. The holt science program, New York.

Guan, F.; Yu, C.; Yang, S. (2019): A GAN model with self-attention mechanism to generate multi-instruments symbolic music. *2019 International Joint Conference on Neural Networks*, pp. 1-6.

Hewahi, N.; AlSaigal, S.; AlJanahi, S. (2019): Generation of music pieces using machine learning: long short-term memory neural networks approach. *Arab Journal of Basic and Applied Sciences*, vol. 26, no. 1, pp. 397-413.

Hong, S.; Kim, S.; Kang, S. (2019): Game sprite generator using a multi discriminator GAN. *KSII Transactions on Internet & Information Systems*, vol. 13, no. 8, pp. 4255-4269.

Jhamtani, H.; Berg-Kirkpatrick, T. (2019): Modeling self-repetition in music generation using generative adversarial networks. *Machine Learning for Music Discovery Workshop, Proceedings of the 36th International Conference on Machine Learning*, vol. 36, pp. 1-7.

Kulkarni, R.; Gaikwad, R.; Sugandhi, R.; Kulkarni, P.; Kone, S. (2019): Survey on deep learning in music using GAN. *International Journal of Engineering Research & Technology*, vol. 8, no. 9, pp. 646-648.

Li, S.; Jang, S.; Sung, Y. (2019): Automatic melody composition using enhanced GAN. *Mathematics*, vol. 7, no. 10, pp. 883-895.

Lieto, A.; Moro, D.; Devoti, F.; Parera, C.; Lipari, V. et al. (2019): "Hello? Who am I talking to?" A shallow CNN approach for human *vs*. bot speech classification. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 44, pp. 2577-2581.

Lim, M.; Lee, D.; Park, H.; Kang, Y.; Oh, J. et al. (2018): Convolutional neural network based audio event classification. *KSII Transactions on Internet & Information Systems*, vol. 12, no. 6, pp. 2748-2760.

Ling, Z. H.; Ai, Y.; Gu, Y.; Dai, L. R. (2018): Waveform modeling and generation using hierarchical recurrent neural networks for speech bandwidth extension. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 883-894.

McFee, B.; Raffel, C.; Liang, D.; Ellis, D. P.; McVicar, M. et al. (2015). librosa: audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference*, vol. 8, pp. 18-25.

Mogren, O. (2016): C-RNN-GAN: continuous recurrent neural networks with adversarial training. *Constructive Machine Learning Workshop (Neural Information Processing Systems*, pp. 1-6. https://arxiv.org/abs/1611.09904.

Oord, A. V. D.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O. et al. (2016): Wavenet: a generative model for raw audio. *arXiv:1609.03499*, pp. 1-15.

https://arxiv.org/abs/1609.03499.

Raguraman, P.; Mohan, R.; Vijayan, M. (2019): LibROSA based assessment tool for music information retrieval systems. *In 2019 IEEE Conference on Multimedia Information Processing and Retrieval*, pp. 109-114.

Rizzi, S. (2018): The 15 most famous tunes in classical music.

https://www.classicfm.com/discover-music/famous-classical-music-tunes/.

Silva, D. F.; Yeh, C. C. M.; Zhu, Y.; Batista, G. E.; Keogh, E. (2018): Fast similarity matrix profile for music analysis and exploration. *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 29-38.

Takahashi, D. (2019): Fast fourier transform. *Fast Fourier Transform Algorithms for Parallel Computers*, pp. 5-13.

Tu, Y.; Lin, Y.; Wang, J.; Kim, J. U. (2018): Semi-supervised learning with generative adversarial networks on digital signal modulation classification. *Computers, Materials & Continua*, vol. 55, no. 2, pp. 243-254.

Vasquez, S.; Lewis, M. (2019): MelNet: A generative model for audio in the frequency domain. *arXiv:1906.01083,* pp. 1-14.

https://arxiv.org/pdf/1906.01083.pdf.

Williams, D.; Hodge, V.; Gega, L.; Murphy, D.; Cowling, P. et al. (2019): AI and automatic music generation for mindfulness. *Audio Engineering Society International Conference on Immersive and Interactive Audio. Audio Engineering Society*, pp. 1-10.

Yang, L. C.; Chou, S. Y.; Yang, Y. H. (2017): MidiNet: a convolutional generative adversarial network for symbolic-domain music generation. *Proceedings of the 2017 International Society of Music Information Retrieval Conference*, pp. 1-8.

https://arxiv.org/abs/1703.10847.