붙임 3

연구논문/작품 최종보고서

2018 학년도 제 2 학기

제목 : 머신러닝을 통한 드럼악보 추출

홍주경(2012311332), 박동준(2013311872)

2018 년 11 월 07 일

지도교수: 윤희용 교수님 ([illegible])

계획(10)	주제(20)	개념(20)	상세(30)	보고서(20)	총점(100)
6	17	16	21	15	75

붙임 3

연구논문/작품 최종보고서

2018 학년도 제 2 학기

제목 : 머신러닝을 통한 드럼악보 추출

홍주경(2012311332), 박동준(2013311872)

2018 년 11 월 07 일

지도교수: 윤희용 교수님 (서명)

계획(10)	주제(20)	개념(20)	상세(30)	보고서(20)	총점(100)

Contents

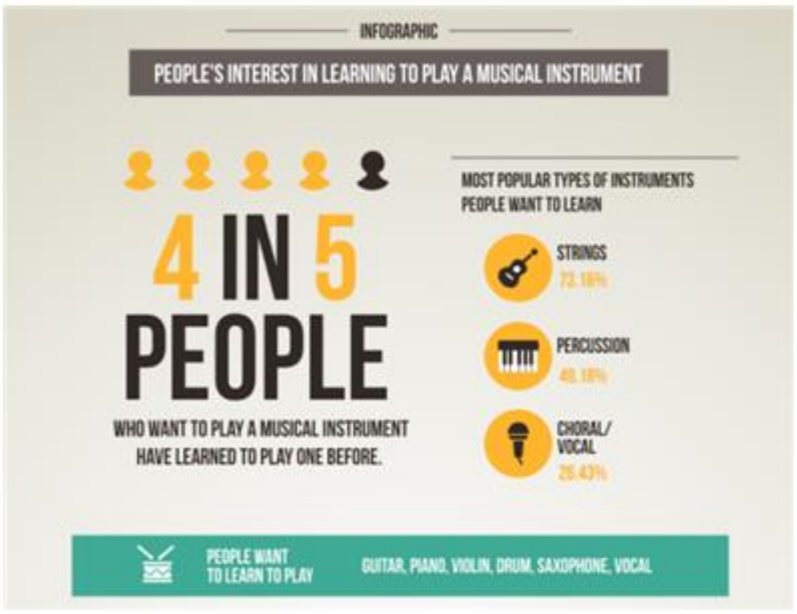
0. 요약

Recently, drum, the most popular and important percussion instrument in modern music, is an instrument that many people want to learn. However, many people are suffering from difficulties of making drum scores. Moreover, many people feel that there are not enough materials about drum score, and also people found that making drum scores are labor-intensive work. appeal for lack of materials to find or make music.

For those who are experiencing this difficulty, we want to create a product that extracts drum scores from voice files. The working principle of the work is as follows. First, from the voice file with drum sound, we classified each musical instruments' sound signal and extracted the data set of drum sound by using MIR(Music Information Retrieval) Technology After that, by applying one of machine learning algorithm and multiclass classification method, categorize classified drum sound by 7 kinds. Finally, visualize the categorized data with the drum score.

1. 서론

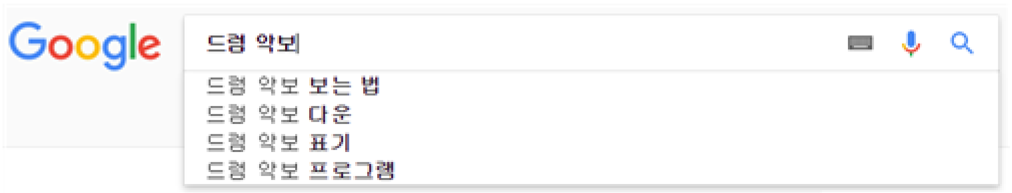
1.1. 제안배경 및 필요성



<Figure 1> Infographic of Survey Reportfrom JakPat:

People's Interest in Playing a Musical Instrument

Drums are the most popular and important percussion instrument in contemporary music. According to the <Figure 1>, a random survey of internet site JakPat, 49.18% of the survey participants replied that they have interest in percussion, including drums. As such, drums are not the only thing for specialized people, and many people are learning drums as a hobby.

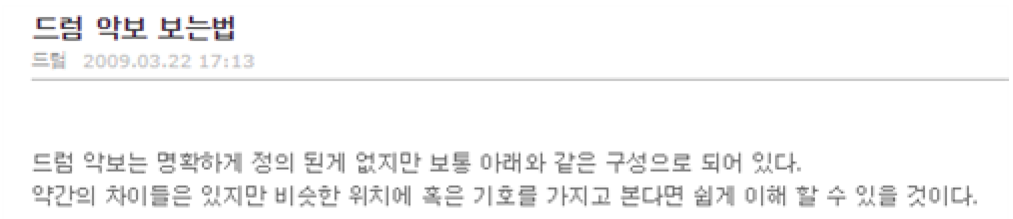


드럼 악보

드럼 악보 보는 법
드럼 악보 다운
드럼 악보 표기
드럼 악보 프로그램

<Figure 2> Related keyword of "drum score" search in Google Korea

Despite these popularizations of drums, however, drummers including hobbyists faced large obstacles. According to <Figure 2>, when we search drum score in Google Korea, the first thing in the auto-complete related query is "How to view drum score". Even in the lecture about "how to view drum score", as in <Figure 3>, says "There are no clearly defined method to make drum score.". As such, reading and making drum score is a quite cumbersome work, and so if you are not a major in drums, writing your own drum score is a very difficult and demanding work. So when people are looking for a piece of drum score that they want, or when they want to write a new piece of drum score, they suffer from the lack of materials.



드럼 악보 보는법

드럼 2009.03.22 17:13

드럼 악보는 명확하게 정의 된게 없지만 보통 아래와 같은 구성으로 되어 있다.
약간의 차이들은 있지만 비슷한 위치에 혹은 기호를 가지고 본다면 쉽게 이해 할 수 있을 것이다.

<Figure 3> Lecture about how to view drum score

1.2. 연구논문/작품의 목표

In order to make it easier for many people who are experiencing this difficulty, we want to create a program that can extract the drum music score by inputting the music file. And, with this work, we hope that many drummers will be able to find and make drum score easier, and hope to be able to lead a more convenient cultural life.

1.3 연구 논문/작품 전체 Overview

1.3.1. Sound Signal Extraction and Drum Signal Extraction

It is the process of getting the data sets of the voice file or the sound source prior to the machine learning. Refer to existing MIR-based technologies and understand how to extract sound signals from audio sources by using matlab program through internet research. In addition, the extracted signal undergoes secondary classification with 3 kinds for example, snare drum, hi-hat, Base Drum, according to the drum sound. Classification will depend on the frequency of the drum part.

1.3.2 IDMT-SMT-Drums Data Sets

Since it was almost impossible for us to collect tons of audio files about drum by ourselves, we found a compressed drum tracks on the internet. This dataset was made by Dittmar and Gärtner in 2014, consists of 95tracks with total 24 minutes. The data set is divided into 3 types of drum sounds which are Snare (SN), Kick (KK) and High Hat (HH). Our code will be using each 95 tracks to extract the audio signals and be trained to discriminate type of drum sounds.

1.3.3. Machine Learning

Before letting the computer to learn the actual dataset, understand the various machine learning modeling techniques first and find the appropriate technique to use. Moreover, by using Python and Matlab, we will study how to apply machine learning technique to the computer. In addition, when we input the real data set to let computer to learn, we will use an appropriate method and programming language. Finally, calculating the accuracy by applying various machine learning methods.

1.3.4. Visualize with Musical Score

According to the result from 3.3, visualize the data from result with open source software which help user to draw musical score. Prior to use that, we need to first study and understand the instructions of that software. Then we can finally draw a musical score by inputting data set with coding.

1.4 선행연구 및 기술현황

1.4.1. Hum On

There is an application called 'Hum On'. It is a simple concepted application that turns humming, or hum, into a musical note. However, this simple concept contains really complex technique. The sound of humming is different for person by person. In addition, even the same sound of the same person is different from the sound of other situations. The 'Hum On' is the application that transfers this to the musical score.

Machine running is the basis for making this difficult technology available. 'Hum On' analyzed humming 'Big Data' by machine learning technology and applied the technique of reading user intention. With this technology, you can turn humming into musical score as well as other sounds.

1.4.2. Machine Learning

Machine learning means let machine to learn something. In other words, even if a person does not explicitly direct logic to a computer, it means that the computer can 'learn' through a large number of data sets and automatically solve the problem through the learned 'experience'. Like that, machine learning is a field of research that develops algorithms that allow machines to "learn" from data and to perform actions that are not explicitly specified in code.

1.4.3. Artificial neural network

Artificial neural network, which is a statistical learning algorithm of machine learning inspired by neural networks of biology. An artificial neural network refers to the entire model that has artificial neurons (nodes) that form a network due to defects of the synapses, which change the binding strength of the synapses through learning and have problem solving ability.

Artificial neural networks include teacher learning that is optimized for problems by inputting teacher signals (answers), and companion learning that does not require teacher signals. In the case of this work, the actual music and its drum score can be used, so teacher learning will be used.

1.4.4. Decision Forest

Decision Forest is a forecasting model which connects observed value with desired value. It is a kind of forecasting modeling method using decision tree in machine learning. Classification tree is a tree model with finite target variable. In this tree structure, a leaf node is a class label and branch is a logical product of characteristics of class label.

Decision tree is visual and explicit. It is used to express decision making process and decision made or data itself. In this project, artificial neural network will be replaced if decision forest is more precise than artificial neural network.

1.4.5. MIR(Music Information Retrieval)

MIR is a technology which analyzes and extracts rhythmic information of audio contents automatically and visualize it. MIR is a kind of Digital signal process(DSP), and it is studied more these days. MIR analyzes specific vectors made by parameterizing audio signal, with techniques such as pattern matching. In plain language, MIR changes music sound to 'sound signal'. Music is just a gathering of some sounds. However, if the music is changed to electric signal, it becomes information. It can be sole signal or split signal, to be sent to someone. In now, especially in web or mobile, MIR is applied to common application.

1.4.6. Visualization

There are not many programs which can draw music score with computer. Moreover, few programs provide circumstance where we can use data from machine learning to draw music score.

In this project, we use a software named 'Lilypond'. Lilypond is complete open source software. It has all functions required to typing music score. Also, many users are improving it animatedly and can use it for free, because it is open source software.

2. 관련연구

First, [1] "Machine Learning for Audio, Image and Video Analysis: theory and applications" is about machine learning for audio, image and video analysis. In this paper, writers used Bayesian theory of decision, Gaussian density, expectation and maximization algorithm, K-Means, etc. for machine learning. They applied the methods to speech and handwriting r ecognition, automatic face recognition and video segmentation and keyframe extraction.

Second, [2] "Machine Learning for Audio, Image and Video Analysis" is similar to [1], but more theoretical than it.

Third, [3] "Application of Symbolic Machine Learning to Audio Signal Segmentation" is ab out machine learning to audio signal segmentation. In this paper, writers address a data-driven approach to the problem of automatic segmentation of speech. They put music in to phones and notes respectively that makes use of symbolic machine learning technique s. This paper divides whole segmentation process into four steps:

1. Series of non-linear transformations are used for building first-order features that allow easy detection of segmentation candidates.
2. Features that describe sound properties in the neighborhood of a segmentation candid ates are developed.
3. The set of segmentation candidates is transformed into machine learning data set by la beling candidates in accordance to the annotated speech corpus.
4. Supervised symbolic machine learning methods are applied resulting in segmentation r ules.

Fourth, [4] "DNN을 이용한 오디오 이벤트 성능 검출 비교" and [5] "CNN과 CRNN을 이용한

오디오 이벤트 검출 비교". In this project, we have to choose which neural network is bett er. There are three candidates: Deep Neural Network(DNN), Convolutional Neural Networ k(CNN), Convolutional Recurrent Neural Network(CRNN). We read this papers considering selecting one of these three candidates and reached the following conclusion: The simple is the best. We choose DNN to solve this problem.

Finally, [6] "Automatic Classification of Musical Audio Signals Employing Machine Learnin g Approach" is about classification of musical audio signals. In this project, classifying the voice datasets is a important issue. We have 7,476 sounds of datasets by 200 drums, so c lassification is an unavoidable and indispensable task. This paper can classify audio signal s, and we read to classify our datasets.

3. 제안 작품 소개

3.1 시스템 구성

In this part we are going to introduce our development environment. What OS, Program for coding and deep learning method did we chose.

3.1.1 Ubuntu 16.04 LTS

<Figure 4> Simple Screenshot for Ubuntu 16.04 LTS

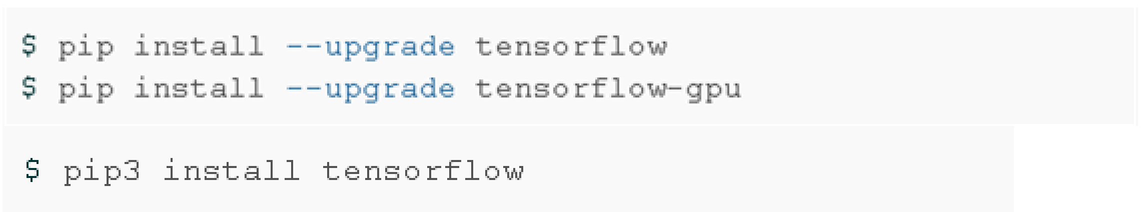
For OS, we selected Linux Ubuntu version 16.04 LTS. Figure 4 shows the simple screenshot of Ubuntu. To briefly introduce this environment, it's code name is Xenial Xerus and is Debian GNU/Linux based. Ubuntu has several important characteristics which are first, it is optimized for personal desktop environment. It provides comfortable user interface. Second, it is free software based. So we can use this OS for free. Also, there are several improvements from previous version which was Ubuntu 14.04 LTS. First, basic setting is now python v3. In previous version, python v2 was used. We will explain about python 3 at next stage. Furthermore, lots of applications are improved for user convenience.

3.1.2 Python v3

For programming language, we used Python 3 which has been growing in popularity over the last few years. This is an interpreted language so is passed straight to an interpreter that runs the code directly. This makes for a quicker development cycle because we just type in our code and run it, without the intermediate compilation step.

Recently Python is taking center stage as a machine learning language. This is because of its simplicity, it has machine learning library and it manages memory for user.

3.1.3 TensorFlow



```
$ pip install --upgrade tensorflow
$ pip install --upgrade tensorflow-gpu
```

```
$ pip3 install tensorflow
```

<Figure 5> Installing Tensorflow

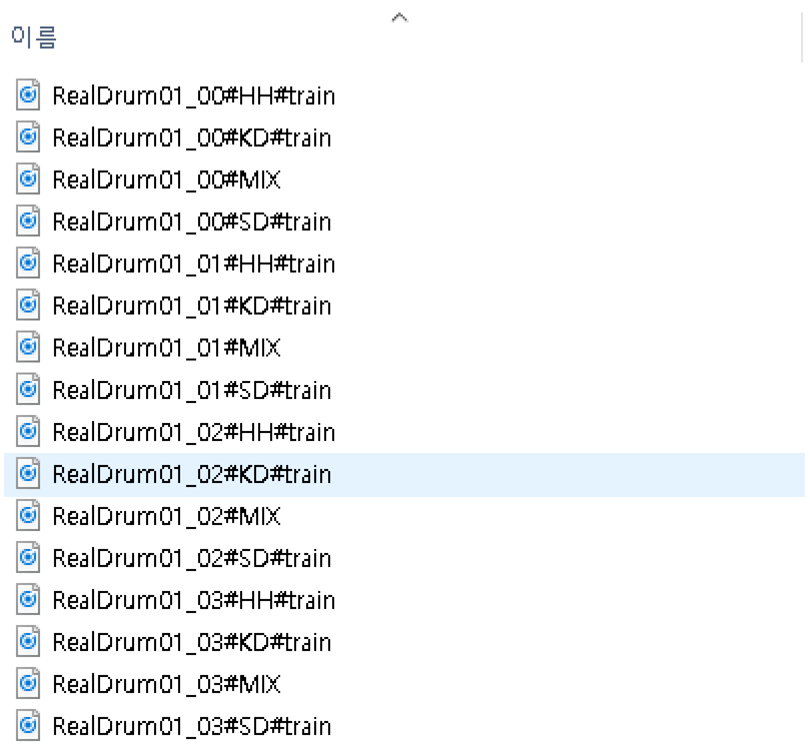
TensorFlow is an open source software library for numerical computation using data-flow graphs. It was originally developed by the Google Brain Team within Google's Machine Intelligence research organization for machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

TensorFlow is cross-platform. It runs on nearly everything: GPUs and CPUs—including mobile and embedded platforms etc.

We installed tensorflow to our desktop by following commands.

3.2 이론적 배경

3.2.1 DataSet for Drum sounds



이름

RealDrum01_00#HH#train
RealDrum01_00#KD#train
RealDrum01_00#MIX
RealDrum01_00#SD#train
RealDrum01_01#HH#train
RealDrum01_01#KD#train
RealDrum01_01#MIX
RealDrum01_01#SD#train
RealDrum01_02#HH#train
RealDrum01_02#KD#train
RealDrum01_02#MIX
RealDrum01_02#SD#train
RealDrum01_03#HH#train
RealDrum01_03#KD#train
RealDrum01_03#MIX
RealDrum01_03#SD#train

<Figure 6> How the data are arranged

For the first thing we do for this project, is to find various Drum audio datasets for deep learning method. The more audio source file we have, the more accurate result will be given.

However, it was impossible for us to get tons of drum audio files by self recording. So we found some opensource datasets in google. This dataset for drum machine contains 95 tracks for drum sounds which are from various drum producing company. It contains electrical, classic and vintage etc. Those drum sound data are again divided into numerous types which are snare, Hi-Hat and kicks. Figure 6 shows how the datasets are divided and arranged. Plus there are a mix tracks. This track will be used to check whether our code can properly classify the types of drum sounds. By those various kinds of drum sounds we can cover various type of audios.

3.2.2 LibROSA

LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

3.2.3 Scikit-learn

Scikit-learn is an open source tool that can be used to data mining and data analysis, built on NumPy, SciPy and Matplotlib.

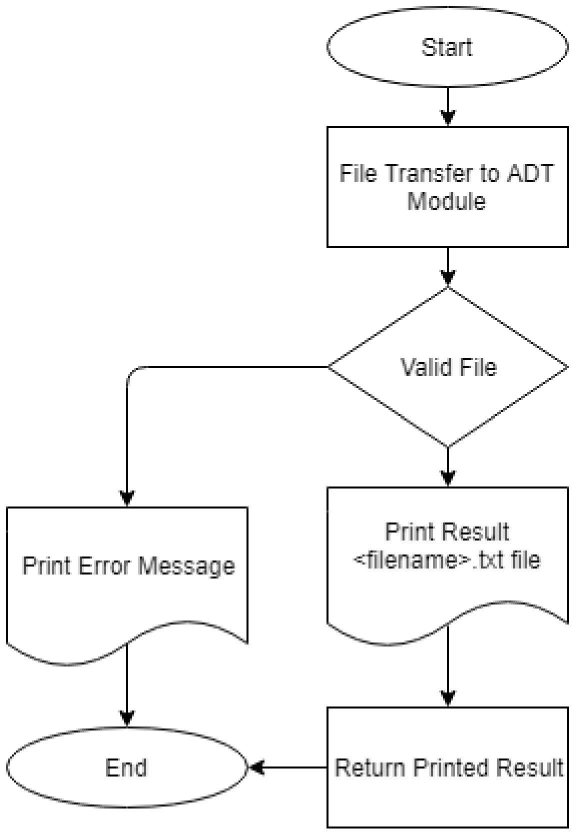
3.2.4 Lilypond

LilyPond is a music engraving program, devoted to producing the highest-quality sheet music possible. It brings the aesthetics of traditionally engraved music to computer printouts.

3.3 구현

In this section we will talk about what works do our code do with flow charts. Our code can be divided into 3 parts which are 3.3.1 ~ 3.3.3

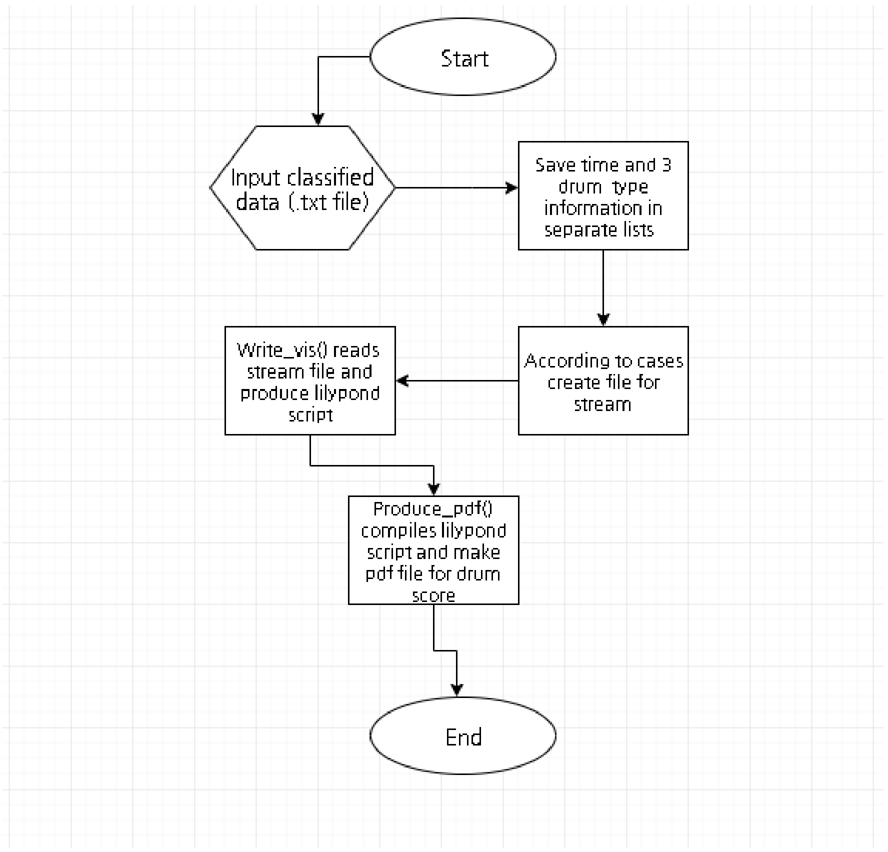
3.3.1 Machine Learning



<Figure 7> Flow Chart for Read_vis

First, user's input files go to ADT module. If that files are invalid, this program prints error message and it ends. On the other hand, if the files are valid, ADT module learns drum sounds from its database and it prints result <filename>.txt file. Then, the function read_vis returns the printed result.

3.3.2 Visualization


Start
Input classified data (.txt file)
Save time and 3 drum type information in separate lists
According to cases create file for stream
Write_vis() reads stream file and produce lilypond script
Produce_pdf() compiles lilypond script and make pdf file for drum score
End

<Figure 8> Flow Chart for Looper

In this stage, we input our classified data sets which contains drum audio file's segmented time data and classified drum types data into visualization.py code. In this code, we have 3 functions. First is read_vis(). This function receives the input file and saves segmented time data without overlapped data. Plus create a txt file with integrated classified drum data which sn, bd, hh, r symbols. Second is write_vis(). In this function we create a lilypond script based on the txt file created by read_vis(). Last, Produce_pdf() will compile .ly script file and create the pdf file for musical score of drum.

4. 구현 및 결과분석

4.1 Classification

We used a Python library named “ADT”. ADT stands for Automatic Drum Transdciption. This library receives parameters of data by machine learn and input audio file. Then, ADT notifies when drum sounds of the audio file are.

4.1.1 ADT(Automatic Drum Transcription)

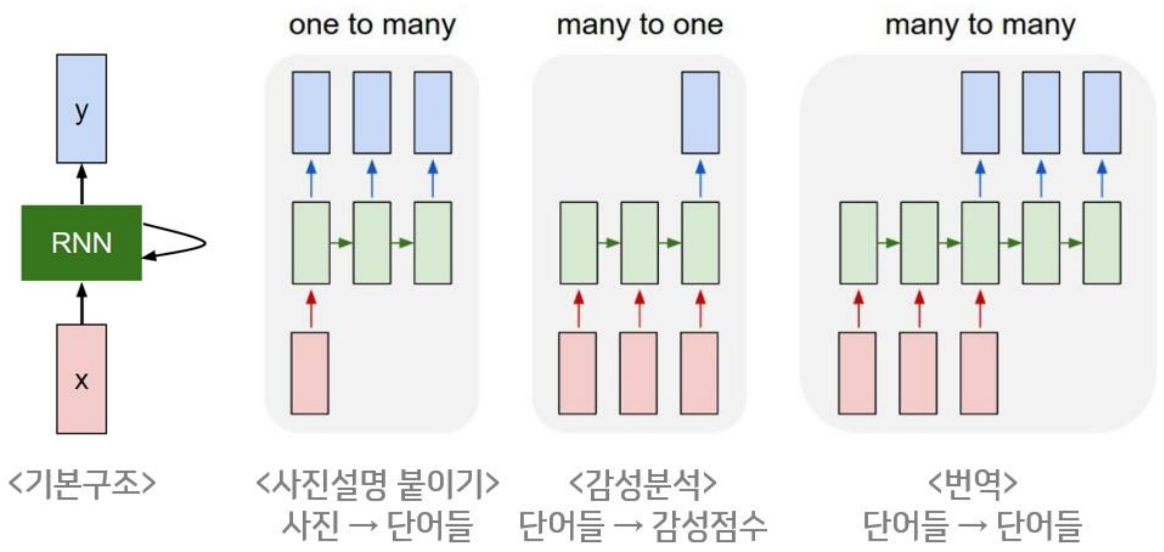
This library has BSD-2-Clause license. The License says: “Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution”. As far as this above is true, this library can be cloned, distributed and modified. So we put this library in our project and specified the license to the license file.
Also, ADT uses RNN and BDRNN, which is described in 4.3.2.

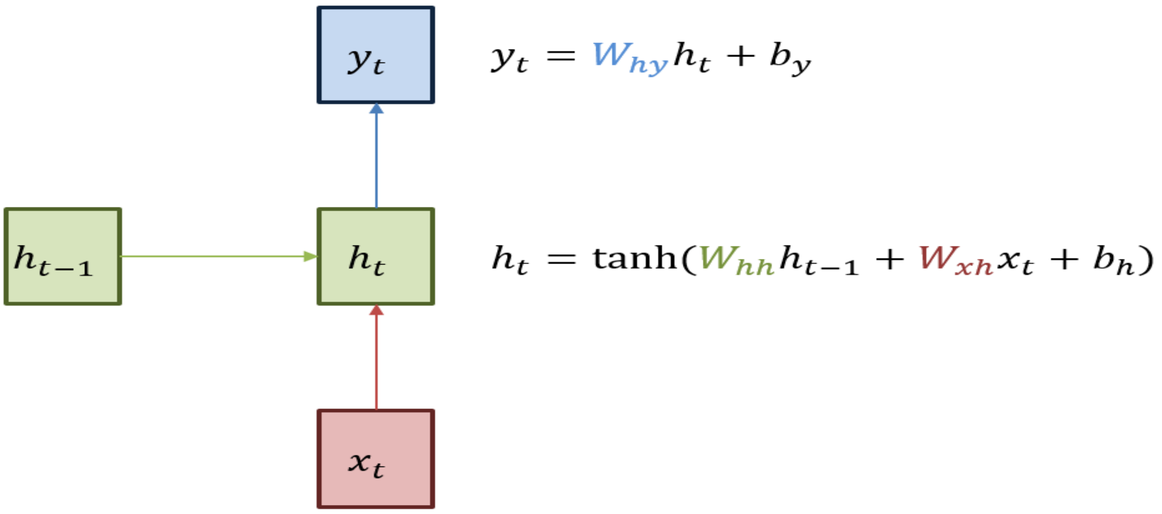
4.1.2 RNN(Recurrent Neural Networks)

RNN is a kind of artificial neural networks. In RNN, hidden nodes is connected with directed edges which makes directed cycle. This model is suitable model for audios, letters, etc.



<Figure 9> Basic structure of RNN

According to <Figure 9>, because RNN is network architecture that accepts input and output regardless of sequence length, RNN's greatest advantage is that it can create structures with various flexibility as needed.



<Figure 10> Basic structure of RNN

The basic structure of RNN can be seen <Figure 10>. Green box means hidden state. Red box is input x and blue box is output y. Current hidden state h_t is renewed when receives last hidden state h_{t-1}. The output y_t in the current state is a structure that is updated with h_t. As can be seen in the formula of <Figure 10>, the activation function of the hidden state is non-linear function, hyperbolic tangent(tanh). The reason for using non-linear functions is to utilize multi-layer networks.

ADT especially uses the BDRNN(Bi-directional Recurrent Neural Networks). BDRNN is a kind of RNN, which uses bi-directed edges for bi-directed cycle. Also, BDRNN shows higher Mean F-measure values in drum transcription RNN.

4.1.3 Bpm-tools

Bpm-tools is a linux tool which outputs bpm when user put an audio file to its parameter. ADT library finds drum sounds with split the music per time, basically. However, music score is more efficient when it is split per beat. Therefore, we used bpm-tools to compute BPM.

4.1.4 ADT-BPM

We used ADT with our own custom. Bpm-tools was used to compute BPM. Then, we used it to split output of ADT per beat. Beats can be calculated by multiplying BPM and minute of audio file. So we can draw the drum music score with the number of beats.

Also, ADT evaluates drum sounds lazily. For example, it can’t catch drum sounds which is split into 16 beats or more, but can catch drum sounds which is split into 8 beats or fewer. However, we wanted more precision. So we edited ADT to evaluate drum sounds more often.

4.2 Visualization

We basically used Lilypond package to draw the actual Drum musical score. But to write proper lilypond script, we need several data preparation. Mainly we need BPM value, audio play time with seconds, classified drum transcription data etc. We will now in this section explain each of data, how we gained and saved those. First, we converted the wav file to mp3. This process is needed because the packages we will use in future are only supporting the .mp3 format. So for conversion we used 'Lame' package that can installed by command 1

- sudo apt-get install lame

<Command 1> install lame package

After all, we can produce .mp3 file for our original .wav file with following command.

- lame <file_name>

<Command 2> converting .wav to .mp3

Second we used 'mp3info' package to save our audio files play length information with following command. This package can be easily installed and used with following commands.

- sudo apt-get install mp3info

<Command 3> install mp3info package

The result will be saved as .txt file for future use.

Third we used ‘bpm-tool’ package for automatically calculate our audio files BPM (Beat per Minute). We can install and use our package with following command.

The result will be saved as .txt file for future use and below is the screenshot of our result.

Now our work for preparation is done. Next step is use those data for calculating the segmentation base time. This calculation is one of the key information for drawing music score because it helps us to properly divide the audio file.

First, we need to convert BPM in to the second based unit. So we divide it in to 60 seconds.

$$BPS = \frac{BEATS}{1\,MINUTE} \times \frac{1\,MINUTES}{60\,SECONDS}$$

<Expression 1> Calculating BPS

Next, we use this BPS to compute total number of segments.

$$\text{Total segments} = \text{CEIL}(\frac{Play\ length}{BPS} + 0.1)$$

<Expression 2> Calculating Total Segments

The reason why we add 0.1 was to reduce error, and since number of segments should be integer value we used CEIL() function to always produce integer value.

Now we still need to calculate one more information which is segmentation base time. This will be the base time unit for whole segment calculation.

$$Base\ unit = \frac{Play\ length}{Total\ segments\ X\ 16}$$

<Expression 3> Calculating Base unit

Here 16 means we used 16 beat based music score.

After calculation, next step is to consider how we can actually draw a musical score. In our code, 2 functions are in charge of this process. We will now explain in detail about each of those function

First function is Read_vis(). The main job of this function is to read ADT result .txt file and produce rough version of drum score. This function also produces several txt files. 'time_list.txt' which contains all time information of Since we are dividing drum sounds in to 3 types ('snare - sn', 'base drum - bd', 'hihat - hh'), the possible cases are 8 cases. Plus, we use 'r' expression for showing no sn or bd or hh appeared

$$\text{Possible cases} = 2\ \left(\text{contain snare or not}\right) \text{X } 2\ \left(\text{contain base drum or not}\right)\ \text{X } 2\ (\text{contain hihat or not})$$

<Expression 4> Calculating possible cases

Those cases are significantly used to write rough script for lilypond. Cases in details are expressed in following <Table 1>

	Hihat	**Snare**	**Base drum**	**Result**
Case 1	o	o	o	**hh sn bd**

Case 2	o	o	x	**hh sn r**
Case 3	o	x	o	**hh r bd**
Case 4	o	x	x	**hh r r**
Case 5	x	o	o	**r sn bd**
Case 6	x	o	x	**r sn bd**
Case 7	x	x	o	**r r bd**
Case 8	x	x	x	**r r r**

<Table 1> Divided Cases

The result for this function is saved as dataset_<filename>.txt format.

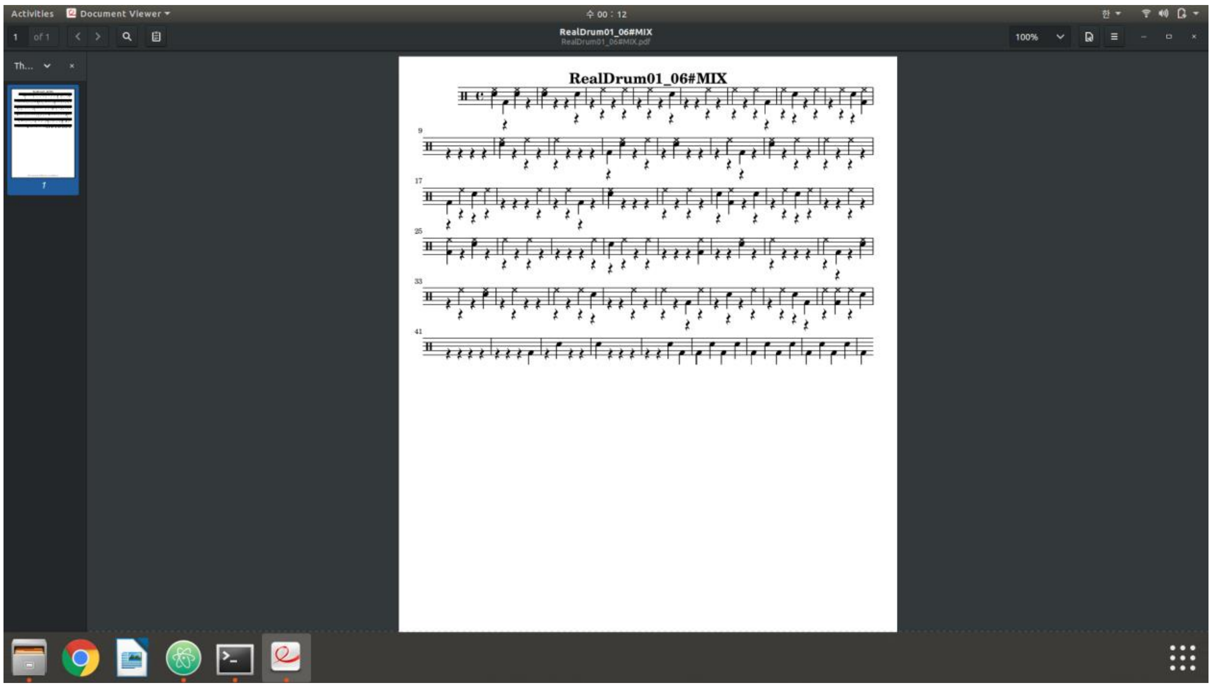
Second function is write_vis(). This function mainly produces complete lilypond script based on rough script written by read_vis(). We have fixed lilypond syntax outline and by adding rough script from read_vis() we can get final .ly lilypond script. In other words, ready to compile.

The last step of this project is to produce .pdf file for drum score. This job is done by a function called produce_pdf(). This function accomplishes a ubuntu terminal command below.

- lilypond <filename.ly>

<Command 4> Produce Drum Score with lilypond package

This command will produce our final drum musical score with .pdf file. The example result is shown in <figure 12>



〈Figure 12〉 Result screen (.pdf)

5. 결론 및 소감

5.1. Works for future

For the future we will use TensorFlow to let computer learn sampled data (the drum signals) from audio file. Then we will make a code to classify 4 different drum sounds.

For this work, we might increase the amount of drum sound types. At the initial perspective, we decided to divide drum sounds for 3 different types snare, hihat, and kicks. However, now we realized that there are more sounds that we need to focus. At last we will make a code to match drum score and specific drum signal. This is also called visualization step. The major thing in this step is to show the specific drum signal to the musical score.

5.2. Conclusion and Perspective

By advancing this graduation work, we have experienced many things that we cannot experience in general undergraduate courses.
First of all, with my team member, we got together in a place and made a lot of technical discussion. This made us to increase major knowledge and learn how to solve various problems.

Moreover, since this was the first time for us to do this big project, we systematically divided the project into parts and solved them step by step. This made us to solve problem fluently.

Finally, we used pair programming method to code our project. By using this method, we shared our major knowledge and anguished specific problem together. This made us to more concentrate on our work. Plus, we can also immediately proceed the code review and help to find each other's errors.

6. 참고 문헌

[1] 4명외박승민. (2010). “VCM과 Beat Tracking을 이용한 음악의 명암 분류 기법 개발.” 한국지능시스템학회 논문지.
[2] 최수환 (2009), Music Information Retrieval(MIR)을 활용한 음악적 리듬의 시각화 연구 - Onset 검출(Onset Detection) 알고리즘에 의한 시각화 어플리케이션, 한국HCI학회
[3] 남언정 (2005), Music Technology의 현 동향, 한국전자음악협회
[4] 유진희, 박상현 (2007), 허밍 질의 처리 시스템의 성능 향상을 위한 효율적인 빈번 멜로디 인덱싱 방법
[5] Francesco Camastra, Alessandro Vinciarelli. “Machine learning for audio, image a nd video analysis : theory and applications”, London : Springer, 2015.
[6] Yu, J. “Machine Learning for Audio, Image and Video Analysis Francesco Camastr a and Alessandro Vinciarelli”, INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING (SPIE) 2009. Berlin; Great Britain; Springer, 2005.
[7] Raskinis, A., Raskinis, G. “Application of Symbolic Machine Learning to Audio Sign al Segmentation” Berlin; Great Britain; Springer, 2005.
[8] 정석환, 정용주, “DNN을 이용한 오디오 이벤트 검출 성능 비교” 한국전자통신학회, 2018.
[9] 정석환, 정용주, “CNN과 CRNN을 이용한 오디오 이벤트 검출 성능 비교”, 한국정보기술학회, 2018.
[10] Kostek, B. , Kupryjanow, A. , Zwan, P. “Automatic Classification of Musical Audi o Signals Employing Machine Learning Approach” New York; Audio Engineering Soci ety, 2011.
[11] Davis, S. Mermelstein, P. (1980) Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. In IEEE Transacti ons on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4, pp. 357-366
[12] X. Huang, A. Acero, and H. Hon. Spoken Language Processing: A guide to theor y, algorithm, and system development. Prentice Hall, 2001.
[13] Southall, C., R. Stables, J. Hockman, Automatic Drum Transcription Using Bi-dire ctional Recurrent Neural Networks, Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR), 2016.

[14] Southall, C., R. Stables, J. Hockman, Automatic Drum Transcription For Polypho nic Recordings Using Soft Attention Mechanisms and Convolutional Neural Networks, Proceedings of the 18th International Society for Music Information Retrieval Confere nce (ISMIR), 2017.

[15] Southall, C., N. Jillings, R. Stables, J. Hockman, ADTWeb: An Open Source Brow ser Based Automatic Drum Transcription System. Proceedings of the 18th Internation al Society for Music Information Retrieval Conference (ISMIR), 2017.