Using Machine Learning Techniques to Improve Computer-Based Assessment of Musical Performances

Grant Proposal

Sumanth Sura

Massachusetts Academy of Math and Science @ WPI

Worcester, Massachusetts

Author Note

Special thanks to Joseph Yu for offering information on Mel Frequency Cepstral Coefficients (MFCCs).

This information helped filter signals affected by noise and proved useful in recognizing timbre.

Abstract (RQ) or Executive Summary (Eng)

Music is deeply imbedded in international culture and has brought joy to people for tens of thousands of years since its inception. Most people in modern society have played music before. Some musicians dedicate their lives to this practice, and therefore are constantly looking for ways to hone their skills. One essential part of becoming a better musician is receiving quality feedback on their performances. Musicians typically look for criticism regarding tone (or, more generally, timbre), rhythm, articulation, intonation, and style. Historically, aspiring musicians have sought the help of a teacher or mentor who is an expert in the performer's instrument. Such mentors provide useful advice since humans have an intuitive understanding of musical quality (e.g. the difference between a "good" and "bad" sound, how different articulations sound, what emotions the performer is trying to convey, etc.). However, with the rise of technology, computer programs and applications have been created to analyze a user's musical performance. The usage of these apps has become increasingly popular during the COVID-19 pandemic, when musicians had limited access to travel and gatherings with teachers. Music teachers also experienced increased burnout during the pandemic, making them less available to students (Sarikaya, 2021). The main problem with these programs is they are only able to detect the note accuracy and rhythm of the performance and return a score to the user based on those two metrics. As a consequence, many musicians feel they do not get adequate feedback from such programs.

To improve a computer's analysis of musical performance, we aim to improve two aspects: recognition (of the different features of music) and understanding (of how these features are). The first major gap in knowledge exists in detecting different instruments. This is essential in assessing techniques during performances. For example, playing a staccato passage (short notes with brief pauses) would sound different on a flute and a violin because a

flute's sound would flow more while the violin's would sound more choppy. Since modern musical analyzers only detect rhythm and note accuracy, they cannot make this distinction. To solve this problem, we can analyze the frequencies emanated by different instruments when playing the same note. When playing a particular note on a given instrument, most frequencies of the audio coming from it are near or at the frequency of the note. But there are also "other frequencies that give each instrument its particular qualities" (Cuff, 2016). Therefore, a major goal of this project is creating an algorithm that can recognize these different frequencies for each instrument. Generating the constituent frequencies from the time-domain representation of the raw audio is not difficult, as one can convert the time-domain representation into a frequency-domain representation using a Fast Fourier Transform (FFT). My strategy is to identify the frequencies with the highest magnitude that are not within a certain range of the desired frequency (the note). As with any ML algorithm, the values of which frequencies correspond to which instruments will improve over time. A second major flaw in musical analyzers is their inability to recognize different musical techniques. For example, modern programs cannot detect a crescendo, or recognize the difference between pizzicato (plucking a violin string) and arco (pulling on a violin string with a bow). These techniques can require gaps between different notes, and the change in dynamics of certain frequencies at a given point in time. To solve these problems, we can transform the time-domain representation of the raw audio files into a 2D spectrogram, which gives us the frequency and magnitude at any given time. We can then use an image processing software, with weights and biases altered to recognize gaps (for staccato vs. legato) and changes in brightness, or magnitude (for dynamic changes). A similar method has been proven to be effective in the past for recognizing different sounds (Thornton et. al., 2019). Lastly, computers have, as of yet, not been able to determine what elements are more important than others for performing a specific piece of music. While they do have the score for any given piece, they do not change their algorithm depending on

which piece is being played. My project intends to, for the first time, be able to prioritize certain elements of music before others and change its algorithm accordingly. This means that for rhythmically complex pieces, intonation and tone are less relevant than they would be in a slow, rhythmically simpler piece. Developing such an algorithm would also allow the individual playing styles of performers to shine through. In a piece where the style is "aggressive", it might be important to exaggerate sforzandos/accents (sudden loud notes) and pauses between notes. It can even be applied to recordings of old pieces, with a clear distinction in the "manner of execution" of certain rhythms for historically accurate performances (musicians who want to emulate the performance during the time period it was written in) modernized performances (performances for our generation's audience) (Liebman et. al., 2012). An algorithm for weighting rhythm against the other elements is to count the number of significant rhythm changes, and then create a machine learning algorithm that takes the input of a user playing that recording to update which spots (i.e. time slots) the user is having trouble with rhythm.

Section II: Specific Aims

The main goal of this project is to give a musician more powerful feedback on their playing, which takes into account tone quality and different musical techniques (staccato vs. legato, or long notes vs. short notes). The final product would ideally also weigh importance to different aspects of playing based on certain elements in the music; i.e. be able to understand how important rhythm vs. intonation is for a given piece.

Specific Aim 1: Distinguish between good tone quality and bad tone quality, and provide the user information about what time frame (or measure number & beat) they played the wrong tone, and understand what was wrong with the tone (too "crunchy", too "airy" or light, etc.) A long-term goal for this aim would be for the computer to understand what constitutes a rich tone on the level of an experienced human musician.

Specific Aim 2: Give the user feedback on the musical techniques in the music. Current music analysis tools do little to nothing to tell the user how well they are performing dynamic changes like crescendos and decrescendos, articulation markings (staccato, marcato, vibrato, etc.), and even different ways of playing an instrument, like (arco vs. pizzicato – playing with a bow vs. plucking the string). The goal is to analyze the shape of the spectrogram to distinguish between different styles of playing.

Specific Aim 3:

Lastly, we aim for the computer to be able to weight different elements of a piece based on the given contents of the piece. This part is crucial because this is the final test that the analyzed data is run through. Giving equal weight to all categories might actually skew the results in the user's favor or against the user's favor.

Section III: Project Goals and Methodology

Relevance/Significance

I am pursuing this project in the hope of harnessing the untapped potential of technology in the field of music. Most musicians have a negative view of technology's effect on the music industry as a whole, but this project could revolutionize the way musicians use technology to improve their playing. It will also be a breakthrough for the development of artificial intelligence as a whole, because historically, humans have viewed music a uniquely human trait. Only we can analyze it, understand it, and enjoy it. This project's success will show that is not necessarily true, and that computers can understand some of the more "human" characteristics of music like tone quality and timbre**.**

Innovation

2D spectrograms are graphs that show different colors to represent amplitude. The main algorithm of this project is a machine learning algorithm that can interpret what the brightness of every pixel means

for the tone quality of the user. However, the computer is plugging in random weights and biases (the main components of the algorithm), but has no understanding of what these numbers mean. The innovation of this project would be understanding the shape of the graph, and combining these elements to get an interpretation of what the computer is actually listening to. To phrase this in other terms, say that a human was listening to a recording of a musician playing short, accented notes. The computer, with solely the machine learning algorithm, will be able to understand how closely the graph resembles what it should be. Implementing a **shape detection** algorithm is important for the computer to understand the "shape" of an accent, or a short note, etc.

Methodology

Specific Aim #1: To find out how closely a given recording resembles "good tone", we plan on first applying random weights and biases to the model, and then tweak the model based on test trials until it gives us the results that we want. This simple set-up for the model will ensure that the biases and weights are the proper values. The output of this model would be probabilities that the audio file matches certain qualities of the recording.

Justification and Feasibility. This method has been proven effective in the past. In 2019, B. Z. Leitner and S. Thorton tested the performance of a machine learning model that edited the layers of a previous model. A machine learning model is built in layers. Each group of neurons (cells that hold numbers), has a set of operations, defined by the weights and biases, that maps it to the next layer. The neurons in the middle have no meaning to humans, but they can eventually be traced to have some vague meaning based on what we want the model to do. Changing the core layers of a machine learning model, a few layers after the input is given, fundamentally changes the model's inner workings. It is like replacing a human brain with the brain of a different species. However, changing the last few layers of a machine learning model will yield a similar model, but can be specialized for certain cases. This is more

like replacing the brain of a human painter with that of a human musician. Leitner and Thorton tried to do exactly that and created three machine learning models: one built from the ground up, one edited version (by changing the final 10 or so layers) of an image processing algorithm, and an unedited version of the image processing algorithm. The image processing algorithm was specialized for spectrograms to classify different types of audio. The edited image processing algorithm had an accuracy rate of 88.5%, which was very close to the tailored ground-up model which had an accuracy of 88.9%. (The control algorithm, the unedited image processing algorithm, had an accuracy rate of 82.9%, significantly worse than the other two models). My project will build off of something similar and use shape detection algorithms on top to get the computer to understand what it is analyzing.

Summary of Preliminary Data.

Three different variations of a given recording were sampled to manually observe differences in the spectrograms before implementing a machine learning algorithm to sort through the data, so that it is clear what parameters should be set for the computer. (This is because the human perception is meant to be the “ideal” output.) Below are three graphs. The first represents an octave scale, played with short bow strokes (staccato). The lowest tones, or the “note” picked up by the human ear, is circled in red.

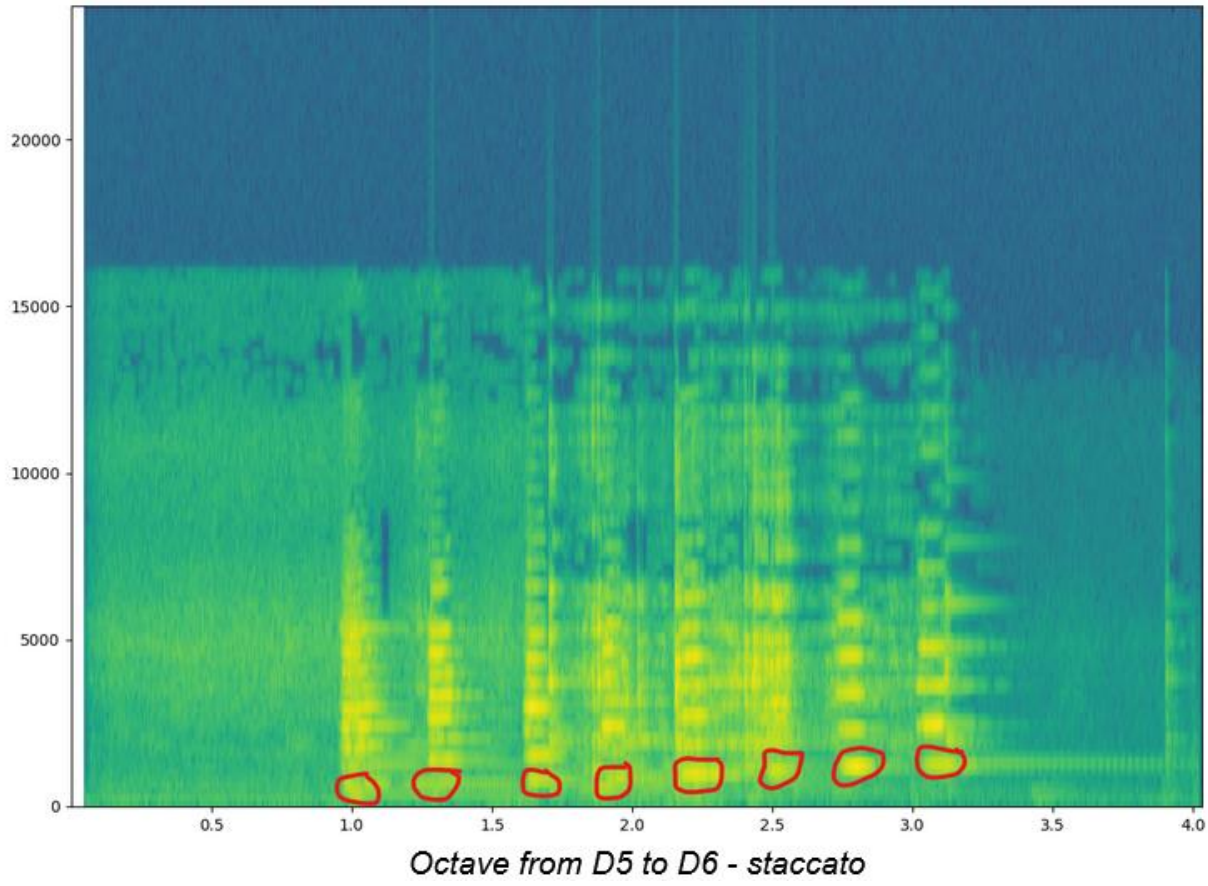


Figure 1. Spectrogram analyzing a 587Hz – 1174Hz scale with staccato notes

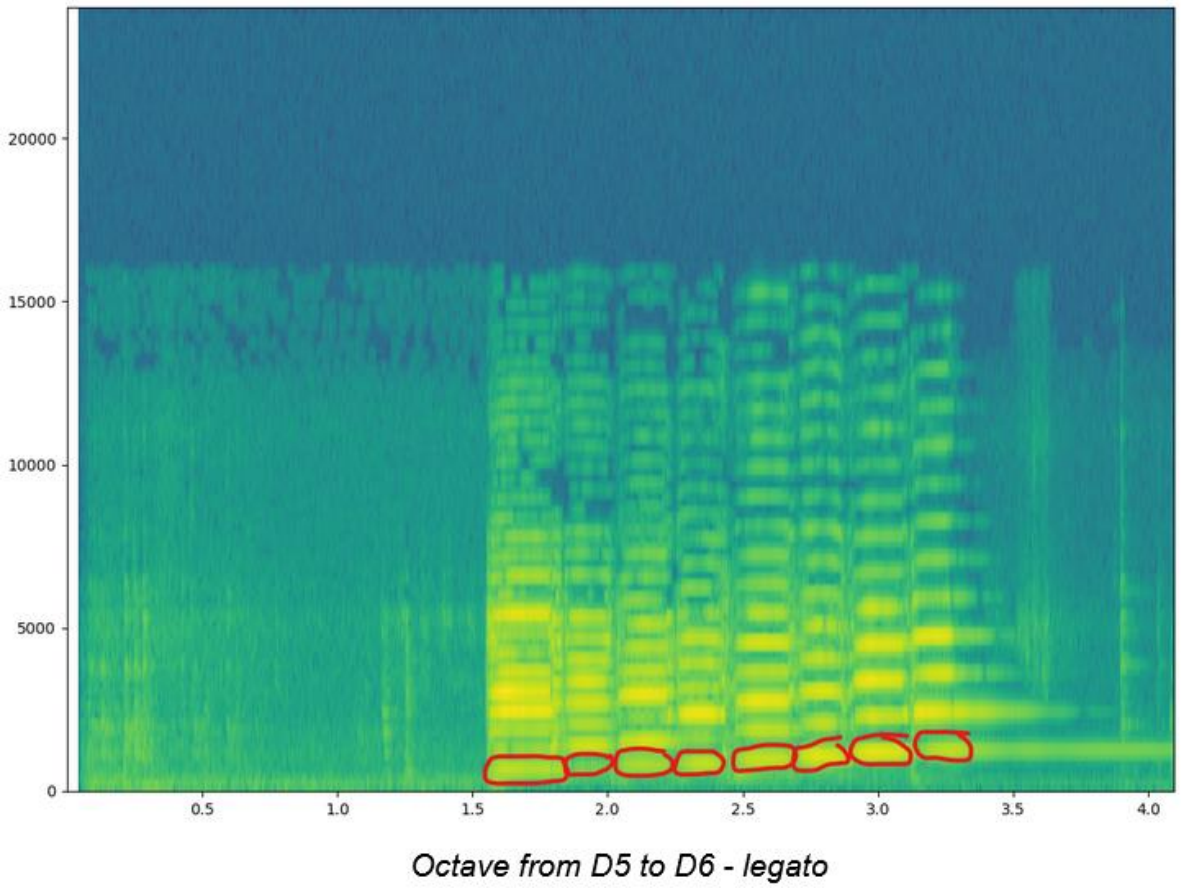


Figure 2. Spectrogram analyzing a 587Hz - 1174Hz scale with legato notes

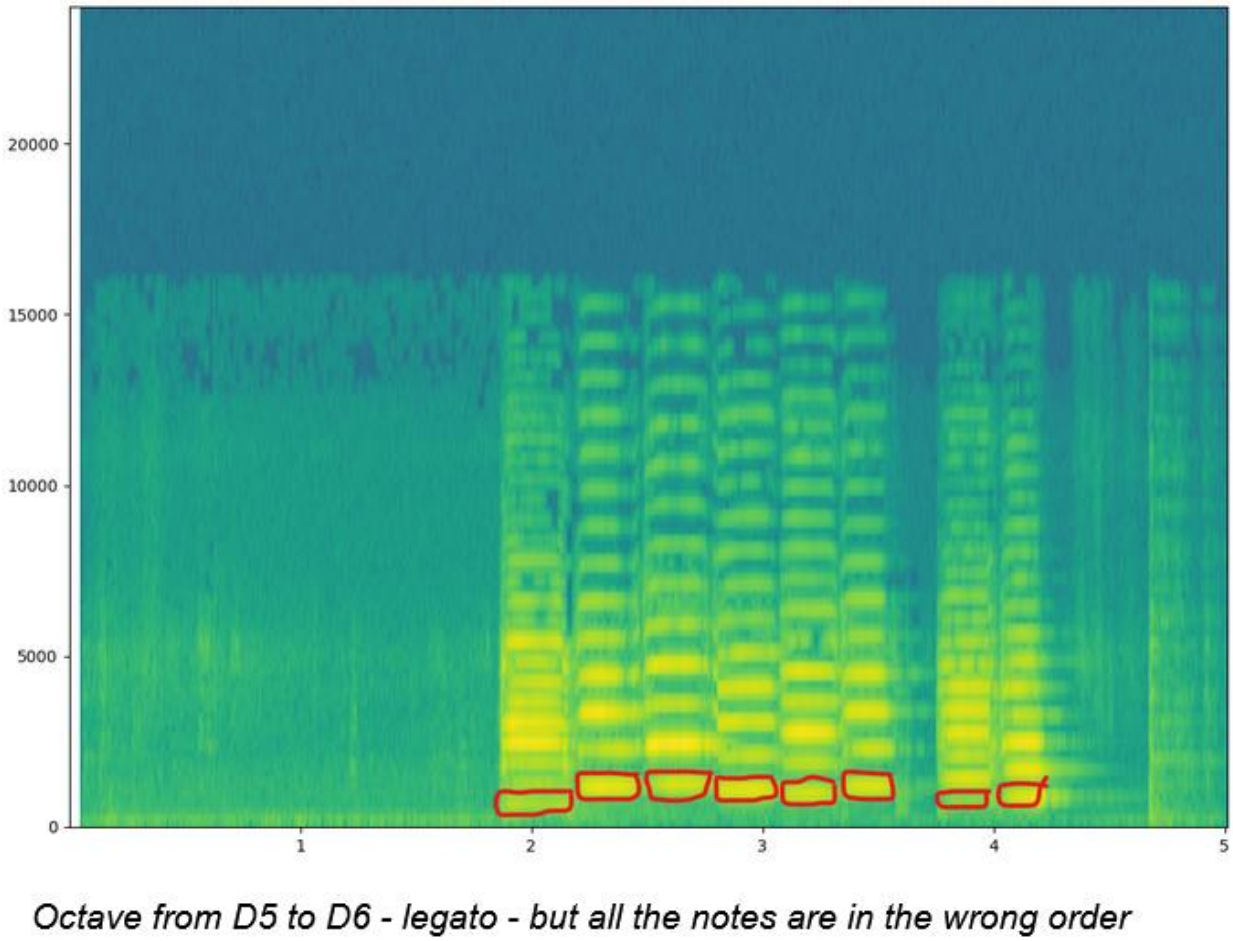


*Figure 3. Spectrogram analyzing a 587Hz – 1174Hz scale with **scrambled** legato notes*

Expected Outcomes. The overall outcome of this aim is to develop an algorithm that will allow computers to understand tone quality in music. This could be extended in a multitude of ways, like understanding how noises in engines compare to one another. However, this algorithm was chosen to run for musical performances because music has, obviously, the greatest variation in timbre and tone.

Potential Pitfalls and Alternative Strategies. We are aware of the possibility that running thousands of spectrograms might be inefficient. So, an alternative method would be to take the Fast Fourier Transform (FFT), which gives frequency data, and utilizes this algorithm at every time frame. This way, we could manually create a spectrogram and analyze the data points without the image

processing. However, this could be a significant hurdle because much more is known about image processing than audio processing.

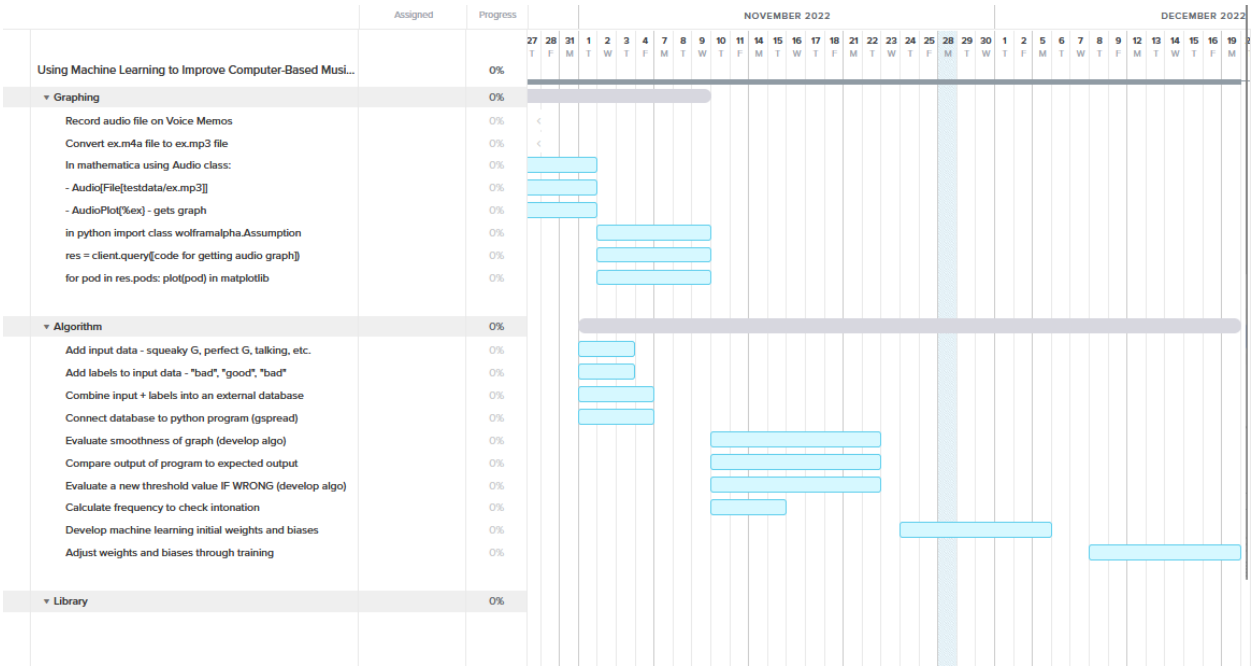
Section III: Resources/Equipment

The only resources required for this project are a musical instrument to record on (currently using violin and voice) and a computer to run code. I am using Python on the IDE (Integrated Development Environment) PyCharm and the text editor VSCode (Visual Studio Code).

Section V: Ethical Considerations

We observe no possible violations to human safety or the environment through this project.

Section VI: Timeline



Section VII: Appendix

Section VIII: References

[1] S. Chu, S. Narayanan and C. J. Kuo, "Environmental Sound Recognition with Time–Frequency Audio Features," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 17, no. 6, pp. 1142-1158, Aug. 2009.

[2] "The Nature of Sound." The Physics Hypertextbook.

[3] Kilshore Prahallad, "Spectrogram, Cepstrum and MelFrequency Analysis," Carnegie Mellon University.

[4] Homburg, Helge, et al. "A Benchmark Dataset for Audio Classification and Clustering." ISMIR. Vol. 2005. 2005.

[5] Piczak, Karol J. "Environmental sound classification with convolutional neural networks." 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2015.

[6] Hershey, Shawn, et al. "CNN architectures for large-scale audio classification." 2017 ieee international conference on acoustics, speech and signal processing (icassp). IEEE, 2017.

[7] Salamon, Justin, and Juan Pablo Bello. "Deep convolutional neural networks and data augmentation for environmental sound classification." IEEE Signal Processing Letters 24.3 (2017): 279-283.

[8] Becker, Sören, et al. "Interpreting and explaining deep neural networks for classification of audio signals." arXiv preprint arXiv:1807.03418 (2018).

[9] Eduardo Fonseca, Jordi Pons, Xavier Favory, Frederic Font, Dmitry Bogdanov, Andres Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra. "Freesound Datasets: A Platform for the Creation of Open Audio Datasets." In Proceedings of

the International Conference on Music Information

Retrieval, 2017.

[10] Bart Thomee, David A. Shamma, Gerald Friedland,

Benjamin Elizalde, Karl Ni, Douglas Poland, Damian

Borth, and Li-Jia Li, YFCC100M: The New Data in

Multimedia Research, Commun. ACM, 59(2):64–73,

January 2016

[11] Kaggle Freeaudio Tagging 2019,

https://www.kaggle.com/c/freesound-audio-tagging2019/overview/evaluation