**Music Genre Classification Based on   
Spectrogram Image Analysis**

David Chen

Harvard University  
51 Brattle St.  
Cambridge, MA 02138  
+1 (647) 998-7802

yuc995@g.harvard.edu

Cindy Liu

Harvard University  
51 Brattle St.  
Cambridge, MA 02138  
+1 (617) 237-0548

cindyliu@post.harvard.edu

**ABSTRACT**

In this paper, we introduce a methodology for the classification of audio data that leverages the many classification tools that currently exist for non-audio data. The classification of images and text data is well-established in the field of machine learning and many libraries and frameworks have been developed for such analysis. Audio classification, on the other hand, has lagged behind its atemporal counterparts, largely due to inherent difficulties in working with audio data. Our methodology allows researchers to apply an image classification tool to audio data by analyzing spectrograms, opening up many avenues in audio analysis and, we hope, accelerating progress in the development of tools for audio classification and analysis.

**Keywords**

Music genre classification; spectrogram; convolutional neural networks; image classification; feature extraction; MFCC; FFT; deep learning; machine learning.

# INTRODUCTION

Music in today’s world is ubiquitous. It can remind us of specific products and brands, thoughts and ideas, and past and imagined events, and can affect our mood significantly. When we hear a piece of music, we almost unconsciously categorize it in our minds – a pop song on the radio, electronic dance music in a nightclub, classical masterpieces at the symphony – and in the vast majority of cases, we have accomplished this labeling within the first 30 seconds of hearing the music. Therefore, there must be certain recognizable characteristics of different musical styles or genres themselves that allow us to classify what we hear so quickly, rather than through analysis of the complete musical work after experiencing it in its entirety.

Machine learning offers a way of giving us some insight into the problem. If a model can predict the genre of a piece of music with only data from the first 30 seconds, it will have achieved the same feat – and in an automatable way. The automation of music genre classification has become a busy area of research in recent years due to the demands of the music entertainment industry. To expand, a music service/business must amass a growing collection of musical works, and the manual annotation and organization of these ever larger and larger music collections has quickly become a wholly unscalable process. The search for efficient ways of automating the classification of music by genre (and many other labels) is thus quickly becoming an essential task in the music industry.

Unfortunately, in the field of machine learning, audio analysis has reached what some researchers call the “bottleneck of music genre classification” [3]. This is partially due to the lack of established conventions surrounding the format of audio data. Currently available audio datasets sometimes consist of raw audio files in varying formats, but more frequently researchers have chosen to reduce the size of the problem by training their models on a data set of derived features or annotations of the raw data rather than the raw audio itself. This has achieved some success – for example, Tzanetakis, *et. al.*, was able to reach a ~60% accuracy rate using standard classifiers and feature sets based on timbre (musical texture), rhythm, and pitch [4] – but many of these feature sets are determined based on previous research in speech recognition, which does not necessarily translate well to the much more complex feature space of musical data. Primarily, these feature sets fail to capture *musical intent*, that is, the inherent patterns underlying the composition of the music, rather than qualities of a particular recording or sample. Musical intent is a key characteristic of music that is not found in speech (the closest example might be spoken word performances), and which is not capturable by investigating pitch, rhythm, or timbre alone.

While much of the research on music genre classification has gone in the direction of feature set analysis, some intrepid researchers have set forth to do battle with raw audio data. Li, *et. al.*, showed promising results when applying CNN methodology directly to audio data (84% accuracy) using a fascinating modification to the image-based CNN in which they took windows of adjacent, overlapping audio clips along the duration of the full piece (paralleling the windows and stride parameters in image analysis) [3]. However, they only achieved 30% accuracy on their test dataset, which they acknowledge is due to overfitting. They additionally note that their method is highly sensitive to variations in musical timbre and other features (an artifact of their feature extraction method, MFCC), and performs less well when trained on data containing more than 4 labels and/or falling into genres that are generally considered to be similar to each other.

Thus a major problem remains of reliably classifying large music datasets that contain numerous labels of varying degrees of “similarity”. The size of the datasets renders it infeasible to employ manual classification to annotate each track; but significantly, it also means that manual extraction of features from the data (currently the only way of capturing musical intent) is also impractical. Classification techniques that do not require manual intervention are therefore at a high premium. We propose a method of spectrogram analysis that not only does not require manual intervention, but also preserves musical intent, generates a reduced feature space of manageable dimensions, and opens up musical analysis to the wide world of available image processing tools, allowing researchers to work around the overfitting problems experienced by previous researchers seeking to utilize image-processing tools directly on audio files.

# METHODOLOGY

## Dataset Selection

The first hurdle in machine learning research is always selecting an appropriate dataset for one’s research needs. While the availability of music datasets is nowhere near that of text, image, or even speech datasets, several options currently exist.

### Million Song Dataset

The largest of these options by far is the Million Song Dataset (MDS) [1], containing metadata and derived features for 1 million Western contemporary music tracks (i.e., it lacks world music and is underrepresented in the classical genre and somewhat overrepresented in the pop, rock, and indie genres [2]). The MDS genre labels are sourced from a website called MusicBrainz.org, which allows users to tag songs – each track can then have multiple tags based on users’ subjective opinions. The majority of the other features come from The Echo Nest, a self-described “music intelligence platform” [5]. The size of the full dataset is 280GB (even without audio files), but they thoughtfully provide a 10,000-song subset (1.8GB) for pilot testing. This is an excellent example of a feature-based dataset, as it does not actually include any audio data. Thus, researchers using this dataset are limited to the features they provide, which include some fairly enigmatic ones such as “song\_hotttnesss,” and are also subject to human “error” (as far as that is meaningful for a subjective label). This dataset proved not to be appropriate for our study, as it contains no raw audio, and additionally does not provide clear genre labels for each song. However, we include it here as it may be extremely valuable for other studies.

### MusicNet

The newest music dataset on the scene, MusicNet, is a collection of 330 recordings of classical music labeled by composer and instrumentation, and additionally containing an incredibly detailed set of over 1 million annotations describing the exact temporal events in each piece, hand-curated by trained musicians [6]. The dataset also includes thousands of MIDI files, some of which were aligned to the recordings using dynamic time warping in order to produce the temporal annotations. However, the creators of MusicNet warn that the dataset has an overrepresentation of Beethoven’s works, as well as of solo piano recordings, simply due to their greater availability. This is a good example of a dataset containing both raw audio files and derived features; it seems that Thickstun, *et. al.*, simply made all of their data publicly available to encourage others to expand on the dataset. Despite the impressiveness of their effort, however, their dataset was designed not for genre classification at all, but for composer/instrumentation classification and note prediction; it was thus also not appropriate for our data needs. However, it has extremely exciting implications for the future of stylistically-accurate machine-generated musical compositions, which may prove to be immensely valuable in reconstructing historical works which were lost or are incomplete (currently the domain of musicologist-composers).

### GTZAN Dataset

In contrast with MusicNet, the GTZAN dataset is one of the earliest available music datasets, having been made publicly available with Tzanetakis’s landmark paper in 2002 [4]. The dataset consists of 1,000 30-second clips of recordings across 10 genres (equally distributed), collected from studio recordings (personal CDs), microphone recordings, and radio broadcasts. According to Sturm [6], Tzanetakis created his dataset for his own research and had no intention for it to be “a benchmark for genre recognition,” and thus there was very little in the way of quality control on the final dataset. (Sturm’s own work seems to be the only comprehensive assessment of the GTZAN data, and it finds a number of shortcomings, casting shadows on the decade’s worth of music genre classification research that had been done using the dataset.) However, for the specific purpose of genre classification it is still one of the best options, as it is of a reasonable size (1.2GB), and is structurally simple, containing only audio files and their corresponding genre labels. We have found it to be an optimal dataset for the scope and purpose of our study.

## Feature Selection

Once we had obtained our dataset, the next challenge was the problem of how to whittle down the enormous number of features present in audio data. For example, for a single 30-second mono-channel track sampled at a rate of 22.05kHz (such as those found in the GTZAN dataset), there are already 22.05kHz × 30s = 661,500 features. This immense feature space cannot be handled in any meaningful way as is, so selecting a method for feature extraction is an extremely important step in audio analysis.

The structure of raw audio data, essentially a series of frequency spectrums over time, means that it can be reduced in two ways: by sampling across the frequencies represented, or over the time duration of the piece. We assessed both options: by calculating the Mel-frequency ceptrum coefficients (MFCC) for each piece, and by looking at the spectrograms for 3-second intervals of the pieces.

### MFCC

MFCC analysis was developed in the mid-20th century as part of early speech analysis research. Originating with the development of the Mel frequency scale, which attempts to map hertz frequencies to “perceptual” pitches in a nonlinear fashion (the current standard is a logarithmic relationship), the Mel frequency cepstrum is generated by taking the log power spectrum of a sound’s mel frequencies, and then applying a linear or discrete cosine transform to the data. This gives us a representation of a sound’s power spectrum based on its perceptual component pitches, and is a more accurate approximation of how a human would perceive the sound (than non-mel-based transforms). MFCC has become one of the standard feature extraction techniques in music genre classification, as it preserves the *timbre* of a sound, a characteristic which is extremely important in music and yet is not measurable in any way as pitch or rhythm are. Musical timbre is usually described not metrically but with some kind of descriptive adjective, like “brassy” or “pure”, and is not only obviously highly subjective, but often ambiguous. For example, as a human listening to a piece of music, our judgment of its genre may be heavily dependent on whether we hear the “twang” timbre of a banjo (folk genre), as opposed to the “twang” timbre of a harpsichord (classical genre). In both cases a timbre description of “twang” is appropriate, but the actual genres differ greatly. The ability to capture timbre in any objective sense, as MFCC seems to do, adds a lot of important features to the available feature sets for analysis.

### Spectrogram

A spectrogram is a visual representation of a duration of sound, and can be seen as a piece of music’s “fingerprint”. It is a highly efficient way of visualizing audio, compacting complex, multidimensional data into a single image that nonetheless offers a quickly interpretable format. The conversion is accomplished with a Fast Fourier Transform (FFT) of the raw data, converting the audio signals into frequency spectrums.

Unlike MFCC, FFT uses the standard linear hertz frequency scale. Though this does not give us the timbre or “perceptual” features of the music, for the purpose of training a (non-human) classification model, it does not seem essential to preserve those features.



Figure 1. Three-dimensional spectrogram  
created using the sound editor Snd [9].

In Figure 1, we see a 3-dimensional version of a spectrogram, clearly illustrating the changes in the sampled frequency spectrums over time. More commonly, however, spectrograms are presented as 2-dimensional images, so that the frequency (Y) axis and time (X) axis are clear, with the third dimension of the frequency amplitudes being encoded as colors along a spectrum, similar to a heatmap, as shown in Figure 2.

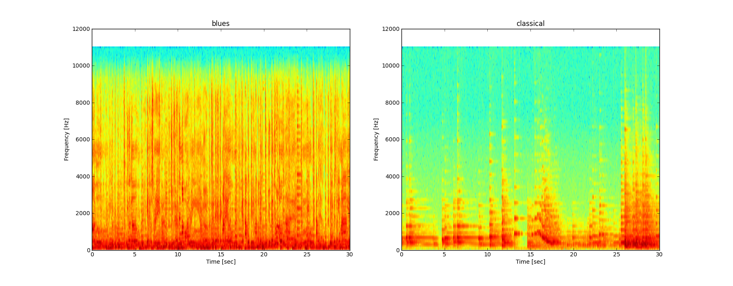


Figure 2. Two-dimensional spectrograms for 3-second clips  
of music recordings from the “blues” and “classical” genres.

The spectrograms in Figure 2 were created from 3-second clips of the 30-second samples in the GTZAN dataset. For our classification model, however, we simplified the data even further by using only 1 color channel, so that our results are based on spectrograms like the one in Figure 3.

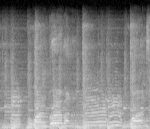


Figure 3. 1-channel spectrogram from the “blues” genre.

# RESULTS

## DDD

Please

## DDD

The

## DDD

Please

## DDD

For

Table . Table captions should be placed above the table

|  |  |  |  |
| --- | --- | --- | --- |
| **Graphics** | **Top** | **In-between** | **Bottom** |
| Tables | End | Last | First |
| Figures | Good | Similar | Very well |

# CONCLUSIONS

Footnotes

## DDD

Do

# SUMMARY & FUTURE WORK

subsection

## Subsections

The

### Subsubsections

The

#### Subsubsections

The

#### Subsubsections

The

# ACKNOWLEDGMENTS

Our

# REFERENCES

1. Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., and Lamere, P. 2011. The Million Song Dataset. In *Proceedings of the 12th International Society of Music Information Retrieval Conference* (ISMIR 2011). http://labrosa.ee.columbia.edu/millionsong/
2. Dawen, L., Gu, H., and O’Connor, B. 2011. *Music Genre Classification with the Million Song Dataset*. 15-826 Final Report. Carnegie Mellon University. http://www.academia.edu/2821380/Music\_Genre\_Classification\_with\_the\_Million\_Song\_Dataset
3. Li, T. L. H., Chan, A. B., and Chun, A. H. W. 2010. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*  (IMECS 2010, Hong Kong, March 17-19, 2010). http://www.iaeng.org/publication/IMECS2010/IMECS2010\_pp546-550.pdf
4. Tzanetakis, G., and Cook, P. 2002. Music Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*. Vol 10, No 5 (July 2002). https://www.researchgate.net/publication/3333877  
   Dataset can be downloaded from http://marsyasweb.appspot.com/download/data\_sets/
5. The Echo Nest. http://the.echonest.com/
6. Thickstun, J., Harchaoui, Z., and Kakade, S. 2016. Learning Features of Music from Scratch. *Under review as a conference paper at ICLR 2017*. https://arxiv.org/pdf/1611.09827v1.pdf
7. Sturm, B. L. 2012. An Analysis of the GTZAN Music Genre Dataset. In *Proceedings of the* *Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies* (Nara, Japan, Nov. 2, 2012). ACM, NY. https://pdfs.semanticscholar.org/61c3/2434b768c388c7babd5d7a0093d81f5a9531.pdf
8. Ioffe, S., and Szegedy, C. 2015. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Technical paper. Google Inc., Mar 2, 2015. https://arxiv.org/pdf/1502.03167v3.pdf
9. Debianux (Wikipedia user). 2007. *3D surface spectrogram of a part from a music piece*. User’s own work, uploaded to Wikipedia Aug. 3, 2007. https://en.wikipedia.org/wiki/Spectrogram#/media/File:Spectrogram.png
10. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= <http://doi.acm.org/10.1145/161468.16147>.
11. Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.
12. Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= <http://doi.acm.org/10.1145/332040.332491>.
13. Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
14. Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
15. Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.
16. Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= <http://doi.acm.org/10.1145/964696.964697>.
17. Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI= <http://dx.doi.org/10.1016/j.jss.2005.05.030>.
18. Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= <http://doi.acm.org/10.1145/90417.90738>.

Columns on Last Page Should Be Made As Close As Possible to Equal Length