Musicology: Validating the Effectiveness of Convolutional Neural Networks for Song Genre Classification

Gabriel Mersy
mersy006@umn.edu

Kevin Lyk
lykxx003@umn.edu

Abstract

We propose a novel experimental technique that supports the argument for the use of convolutional neural networks to classify songs into discrete music genres. Our conclusion is reached by comparing CNN performance on two musicologically separable categories of genre classification tasks: cross-genre similarity and cross-genre dissimilarity. In addition to the experimental results presented in this paper, we expand upon the importance of a broad understanding of musicology when applying artificial intelligence techniques to music informatics problems.

1 Introduction to the Problem

1.1 Signal Processing

A contemporary song is composed of a digital layering of audio signals. These signals are passed through a variety of digital manipulation processes. Each individual component of a song, such as an instrument or vocal, has a unique natural frequency response that is captured by the microphone when recording. In the mixing process, audio signal processing techniques are applied to these tracks to highlight the desirable auditory facets within the individual instruments and vocals (hence the term mixing). Next, in mastering, the signal is glued together to obtain a finished, commercially viable master track that is composed of a singular audio file. Quantitatively speaking, music is reliant on inherent patterns that are fundamentally a function of time and frequency response of the signal that can be readily manipulated through the use of digital and analog techniques.

1.2 Musicology and Music Informatics

The field of music informatics deals with the extraction of information from music through the use of technology. Musicology, on the other hand, is the comprehensive, domain-based analysis of the qualitative and quantitative aspects of music. As we will show throughout this paper, the interplay between these two fields is not to be underestimated.

Characteristics such as tempo and rhythm govern the time domain of a song. For example, the smash hit *Old Town Road* by Lil Nas X has a time signature of 4:4, which implies that the song has 4 quarter-note beats per measure at its tempo, or speed, of 136 beats per minute. By dividing its tempo by its time signature, we can conclude that *Old Town Road* has 34 measures in a one minute time period. Furthermore, a song's structure depicts the framework for the specific elements that are included in a set number of measures within the song. A high energy chorus of 8 measures might lead into a rap verse of 16 measures where there is a completely different set of instruments. A concept known as switching up one's flow refers to the rapper's rhythmic style of delivery, their delivery may differ in each structural aspect of the song.

The other fundamental configuration present in music is the frequency response of the instruments and vocal recordings that make up a song. Music uses an organized collection of notes that are stratified by scale and key. Scales concern a subset of the complete set of musical notes. Each musical note has a natural frequency that defines the note. As an example, the bass note C1 has a frequency of 32.7 hertz. A chord is a layering of two or more musical notes from a scale corresponding to a particular key.

Ultimately, patterns across time and frequency give us an accurate mathematical depiction of the song itself. Elements such as key, frequency response, rhythm, and structure all are important in constructing musicological understanding of the particular song. Additionally, this highlights the fact that a complete knowledge representation of a song cannot be purely reduced to quantitative aspects. As such, qualitative understanding of the music domain is of utmost importance in the application of artificial intelligence techniques.

1.3 Traditional CNNs

Convolutional neural networks (CNNs) are most commonly used in image classification. Image classification expects an input of some image that the computer interprets as a $a \times b \times 3$ array, where a and b are integers representing pixel size, and the 3 renders the RGB color channels. Each individual pixel within the array is then assigned a value from 0 to 255, which allows the computer to interpret facets such as color or shape. This input layer is connected to an output layer by what are known as hidden layers. The term neural network is a reference to the notion of the biological neural networks connecting the dendrites in the brain to facilitate cognition. Each layer is connected to the next layer through the mathematical operation of tensor multiplication.

The convolution layers of a CNN is where most of the image recognition work is done [3]. The first filter, or kernel, of the CNN is applied at these layers and takes a small portion of the image, applies a weight to it, and then sums up the computed values. This modified tensor is then passed on to the next layer. Each filter must fully convolve the entire image before the CNN as a whole can move on. If a filter were $5 \times 5 \times 3$, then it would looks at all of the possible pixel partitions of size $5 \times 5 \times 3$ with respect to the input image. Pooling layers are also important, as their job is to decrease the spatial size of the tensor. In max pooling, the maximum value is taken from each pool to reduce the tensor dimension. This improves the computational time and space complexity when training the network.

The manner in which CNNs see sound is through pictorial representations such as spectrograms. Spectrograms have the ability to be used by many different machine learning algorithms. In our project, we decided to focus on implementing the CNN using Keras, which is a high level API for developing neural networks that was recently adopted by TensorFlow. Keras is very user-friendly, exceptionally fast, and can be run on top of many other libraries. As such, Keras was the perfect fit for the problem at hand.

2 Related Work

2.1 CNNs for Sound Classification

Deep convolutional neural networks have long been used for image classification tasks as is reflected in the literature. However, the work concerning their application to audio classification tasks is still in its infancy. We will start by examining papers centered around the general subject of audio classification using deep CNNs, and then we will proceed to review the literature specifically oriented on music genre classification.

In 2016, Justin Salamon and Juan Pablo Bello proposed a 5 layer CNN for environmental sound classification [9]. They provided detailed insight into the mathematical framework behind their CNN. The network learned to map an input audio spectrogram X to the output sound class Z such that

$$Z = F(X|\Theta) = f_L(...f_1(X|\theta_1)|\theta_L) \tag{1}$$

where each function f_l refers to the layer of the network, and Θ is the set of parameters in the network.

The first three layers are convolutional given by

$$f_l(X_l|\theta_l) = h(WX_l + b), \theta_l = [w, b] \tag{2}$$

where W is a filter that extracts patterns from the audio, $h(\cdot)$ is an activation function such as ReLU, b is the vector bias, and X_l is an input tensor (roughly equivalent to a matrix).

The next two layers are fully connected and consist of an activation function and a matrix product instead of convolutions. A technique called dropout is applied to the last two layers with $p = 0.5$ which inhibits learning between neurons and the ensuing over fitting that occurs along with this phenomenon. Ultimately, the paper reports an accuracy of 0.74, or 74 percent.

In another article, Jordi Pons and Xavier carry out a comparative study of the architecture of CNNs for audio classification [8]. Holding the data set constant, they show that the more model learn parameters a CNN has, the better the performance. This is an important consideration, and shows that in order to truly learn from audio, you need a very deep network which therefore contains a significant amount of parameters.

2.2 CNNs for Music Classification

Now turning to the specific task of classifying music, in 2010 Tom Li, et al. proposed a unique architecture for music classification [5]. In order to process the audio signal, they recommended a 190 × 13 input layer, where each row consists of a frame of 13 mel frequency cepstral coefficients (MFCCs). This implies that the network will be expecting a $\mathbb{R}^{190 \times 13}$ tensor as the input. The convolutional base consists of 3 layers, each reducing the tensor size until it is flattened into the class output format. The authors conclude that CNN is a viable alternative to traditional music classification methods. Additionally, they provide evidence to the claim that MFCC allows for the extraction of patterns in an audio sample with a similar effect to an image-based approach, such as the aforementioned spectrogram.

One of the most common practices for applying CNNs to music informatics tasks is the use of 2D time frequency visualization as input data. This way, the CNN can actually "see" the patterns present in the sound. In Choi, Fazekas, and Sandler's work, they propose a different way to look at these visualizations [1]. They assert that a CNN analyzing a spectrogram is not necessarily the same as the observation of an image. The solution that the authors offer is a technique called auralisation. Auralisation requires another methodological phase in the CNN that allows for a more accurate time-domain signal.

In their 5 layer CNN, the researchers focused on 4 different genre-specific inputs: Bach, Park, Toy, and Eminem. As each layer in the CNN got extensively more detailed, the differences in the visualizations of each song type became significantly more apparent. The first convolution layer had an input tensor size of 257 by 73, while the last convolution layer had a tensor size of 8 by 5. Their first run-through obtained an accuracy of 75 percent. They also proved that characteristics such as musical key, instrument type, and chord progression affected each layer differently. As evidence to their claim, they provided a correlation graph that showed that key had the highest correlation through each of the 5 layers. On the other hand, the inclusion of instruments had the lowest correlation. However, in the final layer all 3 variables had relatively the same correlation, showing that their CNN architecture ultimately worked well. In the end, this technique can reduce computational time and prove to be an accurate technique for categorizing music.

Another paper by Park and Lee [7] depicts how CNNs can be used to extract information from music. They rely on a 4 layer CNN with max pooling throughout all of the convolution layers. Their CNN was created to distinguish music sound bites from random noise. This implies that their task was aimed at binary classification into the categories of either noise or music. By including 5 different signal-to-noise ratios (SNR), they experimentally evaluated how well the CNN performed given differing amounts of random noise. Confirming our intuitive understanding, higher SNR values were associated with more accurate results. The most accurate outcome occurred when the SNR was clean. As such, Park and Lee were able to conclude that given a clean SNR, one only needs a temporal sound bite length of 0.8 seconds in order to accurately determine whether the input was noise or music.

Similarly, since CNNs are largely black box in nature, especially when it comes to audio, Zieler et al. developed a method to deconvolve the convolution filters (kernels) so that humans can understand the patterns that are being extracted [12]. Furthermore, they proved the efficacy of CNNs for audio classification using an ablation study, which is a way to examine the contribution of each hidden layer. The method showed that CNNs home in on musical aspects such as a main melody, bass, and vocal textures. This is an important contribution to the topic at hand, because CNNs for audio are less intuitive than their image counterparts.

Another consideration to keep in mind the vital importance of high quality audio files: an imperative outlined by Uhlich, Giron, and Mitsufuji [11]. In their paper, the authors trained neural networks to be able to extract a single instrument out of a song or piece of music. They used the following representation for their deep neural network (DNN):

$$x_k + 1 = \max(W_k * x_k + b_k, 0), k = 1, ..., K \tag{3}$$

where x_k is the input to the kth layer and W_k, b_k are the networks weights and bias respectively at that particular layer. Notice that each hidden layer has ReLU activation. The authors found that their error depended on the instrument that was trying to be extracted. For example, the violin was much easier to extract than the horn or piano. This makes it clear that data needs to have higher quality, as the team went on to make note that some data samples were lower quality than others. In the future, they planned on only using the data from the same set that was *proven* to be high quality. In addition to this, the results suggest that high data quality is a precursor for prediction accuracy.

However, unlike many other types of neural networks, data *quantity* may not be of chief importance in music informatics. Lee and Nam recount how their CNN "failed to outperform the best of the previous works" on their specific data set [6]. Although they were able to incrementally improve their results by using alternative data sets, their CNN still lacked the best-in-class accuracy that they had hoped for. Their CNN aimed at categorizing songs into different music genres. Lee and Nam found that the more songs they included in the data set, the worse their CNN performed. The only data set that showed improvement when compared to their baseline had 900 songs, as opposed to the ones that had well over 200,000 songs. Their results bring up an important consideration; namely, that machine learning classification algorithms are difficult to apply to music, because it lacks standardization. Music instead focuses on promoting characteristics such as creativity and novelty. Hence, data set selection can prove to be the most important aspect to the field of music informatics. Additionally, a unique condition for music genre classification is summed up by the notion of song *quality* over *quantity*, which is an observation that is in sharp contrast to the consensus on the importance of big data for deep learning.

3 The Data Set

We used the GTZAN data set for our genre classification task [10]. This data set is one of the standards for music genre classification. The version of GTZAN we utilized consists of 8000 music files divided into the following 10 genres: Blues, Classical, Country, Disco, Hip hop, Jazz, Metal, Pop, Reggae, and Rock. Each class has 800 music snippets that are 30 seconds in length. Each song has a sample rate of 22,050 hertz. An important consideration to note is that this is exactly one half of the standardized sample rate for commercial music (44,100 hz). This implies that the each audio file was either sampled or re-sampled at the rate of $\frac{1 \text{ sample}}{4.54\times10^{-5} \text{ seconds}}$ which is a lower rate than the standard for commercial audio. Therefore, the audio quality of this data set is much lower than what would be expected.

A spectrogram is defined as a visual representation of the loudness (decibels) across the entire spectrum of frequencies with respect to the audio file's time domain. Table 1 contains audio spectrograms for 4 randomly-selected songs within 4 of the classes of genres. We can make inferences from the patterns contained in a spectrogram in the same manner that a convolutional neural network would. For example, in Figure 4 around the 20 second mark, there was a gradual reduction in the loudness across the frequency spectrum as shown by the black region indicating a very quiet region of -80 dB. As musicians, this tells us that the high-frequency instruments like trumpets or violins stopped playing, and then the lower-frequency instruments stopped playing soon after that point. This reduction in volume leads to a few rest measures where it is likely that none of the instruments were playing. Thus, this highlights the fact that a large number of patterns can be extracted from a simple spectrogram. Spectograms are invaluable pieces of information for a convolutional neural network.

3.1 Data Pre-processing

We employed the use of the pre-processing techniques of Huang, et al., in which they selected a 2 second spectrogram to represent each audio file as to reduce the network training time [4]. The training of CNNs require vast computational resources due to the high number of learnable parameters present in these models. In the interest of reducing the algorithmic run-time and space requirements,

Table 1: Mel-frequency spectrograms

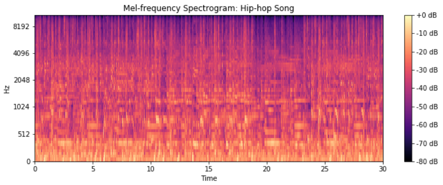

Figure 1: Hip-Hop

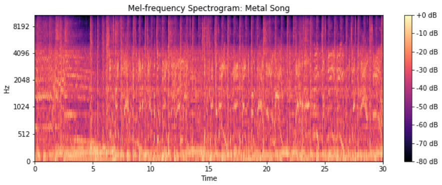

Figure 2: Metal

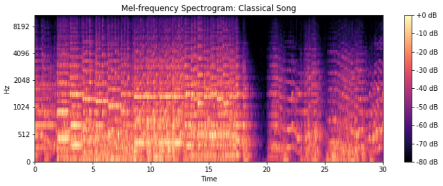

Figure 3: Classical

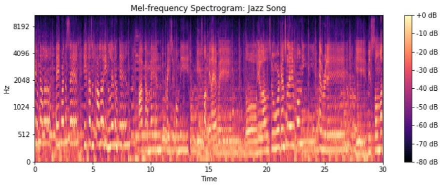

Figure 4: Jazz

it is best practice to convert the data into an efficient format. The authors used Librosa, which is arguably the most popular Python library for audio signal processing. Each song is represented as a mel spectrogram with 64 windows occuring over a 2 second time interval to obtain a $\mathbb{R}^{64\times 173}$ tensor. With 800 songs for each of the 10 classes, the dimensionality of the complete feature set is $X \in \mathbb{R}^{8000\times 64\times 173}$. Each song tensor $X_i \in X$ has a corresponding numerical genre label $y_i \in \{0, 1, ..., 9\}$ such that the label vector has 8000 elements.

4 Methodology

4.1 Problem Representation

Our overall objective is to analyze the performance of a single convolutional neural network leveraged on a set of binary classification tasks selected with regard to characteristics of one genre u in relation to another genre v. In other words, we are interested in highlighting the variability in prediction accuracy with respect to specific genre classification tasks. We are reproducing the CNN used by Costa, et al. that achieved a very high prediction accuracy [2]. Our neural network is given by a composite function $\hat{F} : \mathcal{X} \to \mathcal{Y}$ that maps the feature song tensor to the probability of the class label y_1 where $\hat{p} = P(\hat{y} = y_1)$ and $\hat{p}^C = 1 - P(\hat{y} = y_1) = P(\hat{y} = y_0)$ such that

$$\hat{p} = \hat{F}(X|\Theta) = \hat{f}_5(...\hat{f}_1(X|\theta_1)|\theta_5) \tag{4}$$

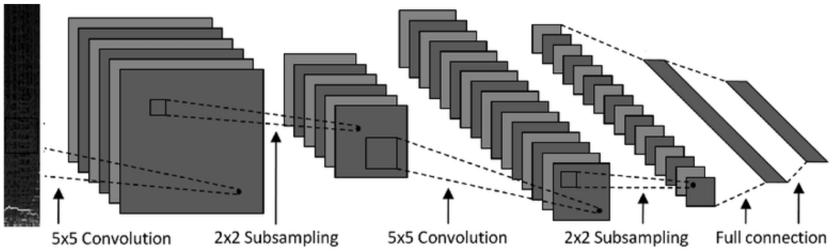

Figure 5: General CNN architecture [2]

with the parameters of layer i given as $\theta_i \in \Theta$. We use the loss function binary cross entropy and compute the trivially-defined accuracy metric after each forward and backward pass of the data to assess training performance. Binary cross entropy is given by:

$$\mathcal{L}(y, \hat{p}) = -(y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})) \tag{5}$$

where $\hat{p}$ is the vector of probabilities and y is the vector of actual class values.

Our set of tasks T with $|T| = 4$ consists of elements $t_i \in T$ where t_{i0} is the first data set, and t_{i1} is the second data set. Our modified architecture of the 5-layer Costa CNN is as follows (see Figure 5 for visualization):

1. Convolution layer l_1 with a 5×5 filter (kernel), 1×1 stride, and ReLU activation
2. Max pooling layer l_2 with a 2×2 pool and a 2×2 stride
3. Convolution layer l_3 with a 5×5 filter (kernel), 1×1 stride, and ReLU activation
4. Max pooling layer l_4 with a 2×2 pool and a 2×2 stride
5. Fully-connected output layer l_5 with sigmoid activation

Note that the softmax activation function in the original Costa output layer was converted to a sigmoid activation function which is more appropriate for binary classification. The neural network was implemented in Keras using a TensorFlow backend. We divided the specific data sets of 1600 samples into training and validation subsets of 1120 samples and 480 samples respectively for each task (70 percent/30 percent split). We used the Adam stochastic optimization algorithm with the parameters set as $\epsilon = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$[1].

4.2 Experimental Design

We created a total of 4 classification tasks for our experiment. The first two tasks consisted of genres that, in terms of musicological components, are qualitatively distinct. Conversely, the remaining two tasks consisted of genres that are qualitatively similar (interrelated):

[1] You can find our code here: https://github.com/gmersy/song-genre-classifcation

1. Hip hop and Classical (distinct)
2. Heavy metal and Jazz (distinct)
3. Pop and Hip hop (similar)
4. Rock and Country (similar)

For example, Hip hop music often has loud, digitized drums with a simple rhythm, a heavy baseline, and loud unmelodious rapping, while classical has a broader blend of orchestral instruments following a complex rhythm and almost always lacks vocals. This implies that we would hypothesize that the two genres are markedly distinct in terms of their spectrograms on average and therefore can be easily differentiated by the CNN. This is in sharp contrast to Rock and Country, since they use very similar instrumentation with a significant focus on guitar and piano. Thus, we would expect that these two genres would have relatively similar spectrograms on average and would then lead to a higher degree of difficulty on the part of the CNN in differentiating the genres. The exact same CNN was used for each task with the same parameter values. The two experimental variable that were initially changed was the classification task; that is, the data sets used. Following the conclusion of this experiment, a change in epoch number was was evaluated to extend upon the results of our initial experiment.

5 Results

Our results are given in Table 2 with the validation set accuracies for each task across the two epochs $e_1,\ e_2$. $\Delta_{1,2}$ is the marginal change in accuracy when comparing the two training times. $\Delta_{1,2}$ is defined as $a_{e_2} - a_{e_1}$, where a is the validation accuracy for the epoch e_i.

Model	**Accuracy**	**f1**	$\Delta_{1,2}$
Hip hop and Classical	1.000	0.994	*-0.006*
Heavy metal and Jazz	1.000	0.990	*-0.010*
Pop and Hip hop	0.048	0.925	*0.877*
Rock and Country	0.277	0.446	*0.169*

Table 2: Validation set accuracies for the 4 classification tasks

6 Conclusion

The results provide evidence to the claim of performance differences between the qualitatively similar genre classification tasks and the qualitatively distinct classification tasks. Notably, the two tasks that had a high degree of musicological distinction between the two genres under consideration had a validation set accuracy of 100 percent at the end of the first epoch. To contrast this, the two tasks with inherent musicological similarities presented accuracies of less than 30 percent at the end of the first epoch. The Hip hop and Pop task then increased to above 90 percent accuracy for the second epoch, while the Rock and Country data set only increased modestly. Hence, this suggests that the CNN presented a higher magnitude of difficulty when classifying similar genres and needs further training to improve its performance. Conversely, the CNN was almost instantly able to classify the musicologically distinct genres as shown by the reduced accuracy on the second epoch for both of these tasks.

Furthermore, these results are largely explainable with respect to the data set. Pop and Hip hop from the decade of 2000, which is the decade during which the data set was released, subjectively displayed a larger degree of musicological divergence when compared to the convergence the genres have underwent in recent years. The recent introduction of trap Hip hop to popular music has proven to merge the two genres to a greater extent than that which existed in the previous decades. One of the main limitations to our conclusion centers around the notion that we relied on 2 second snippets of music to train our CNN due to our lack of computational resources. Our results could be extended by using better computers and processors to train the network on a larger portion of the each audio snippet.

In the future, we hope to extend our study by including other genres of music. Given the results that we obtained, it would be best to work on applying musicological techniques to differentiate genres that are inherently similar to each other. Since our poorest result was between the genres of Pop and Hip hop, this would be a great starting point. Looking back at Table 1, it also is apparent that the spectrograms of Hip-Hop and Metal are very similar. Even though the two genres don't sound the same to a human ear, testing our CNN on similar visuals may yield more meaningful results to improve classification accuracy in the long run. We think that we may have uncovered something notable here with our research, and we accordingly would like to extended our study to produce a full conference paper within the next year.

References

[1] Keunwoo Choi, George Fazekas, and Mark Sandler. Explaining deep convolutional neural networks on music classification. *arXiv:1607.02444*, 2016.

[2] Yandre M.G. Costa, Luiz S. Oliveira, and Carlos N. Silla. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied Soft Computing*, 52:28 – 38, 2017.

[3] Adit Deshpande. A beginner's guide to understanding convolutional neural networks.

[4] D. Huang, E. Pugh, and A. Serafini. Music genre classification. `https://github.com/derekahuang/Music-Classification`, 2017.

[5] Tom Li, Antoni Chan, and Andy Chun. Automatic musical pattern feature extraction using convolutional neural network. *Lecture Notes in Engineering and Computer Science*, 2180, 03 2010.

[6] Juhan Nam and Jongpil Lee. Multi-level and multi-scale feature aggregation using sample-level deep convolutional neural networks for music classification. *arXiv:1706.06810*, page 3, 06 2017.

[7] Taejin Park and Taejin Lee. Music-noise segmentation in spectrotemporal domain using convolutional neural networks. *Electronics and Telecommunications Research Institute*, page 2, 2015.

[8] J. Pons and X. Serra. Designing efficient architectures for modeling temporal features with convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2472–2476, 2017.

[9] Justin Salamon and Juan Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, PP, 01 2017.

[10] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[11] Stefan Uhlich, Franck Giron, and Yuki Mitsufuji. Deep neural network based instrument extraction from music. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 04 2015.

[12] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.