Neural Network Music Genre Classification
Classification des genres de musique par réseau neuronal

Nikki Pelchat and Craig M. Gelowitz

***Abstract*— Music genre classification utilizing neural networks (NNs) has achieved some limited success in recent years. Differences in song libraries, machine learning techniques, input formats, and types of NNs implemented have all had varying levels of success. This article reviews some of the machine learning techniques utilized in this area. It also presents research work on music genre classification. The research uses images of spectrograms generated from timeslices of songs as the input into an NN to classify the songs into their respective musical genres.**

***Résumé*— La classification des genres musicaux utilisant des réseaux de neurones (NNs) a connu un succès limité ces dernières années. Les différences dans les bibliothèques de chansons, les techniques d'apprentissage machine, les formats d'entrée et les types de NNs mis en œuvre ont toutes eu des niveaux de succès variables. Cet article passe en revue certaines des techniques d'apprentissage machine utilisées dans ce domaine. Il présente également des travaux de recherche sur la classification des genres musicaux. La recherche utilise des images de spectrogrammes générées à partir d'intervalles de temps de chansons comme entrée dans un NN pour classer les chansons dans leur genre musical respectif.**

***Index Terms*— Classification, convolutional neural network (CNN), deep learning, music, music genre, neural network (NN), spectrogram.**

I. Introduction

A. Machine Learning and Neural Networks

MACHINE learning has become very popular in recent years. Depending on the type of application and the data set available, certain types of machine learning techniques are more appropriate than others for different applications. There are generally four main types of learning algorithms in machine learning. The main types of learning algorithms include supervised learning, unsupervised learning, semisupervised learning, and reinforcement learning [1]. Supervised learning utilizes a fully labeled data set to build a mathematical model whereas unsupervised learning attempts to extract useful features from an unlabeled data set without any specific target in mind. Correspondingly, semisupervised learning utilizes a data set that contains both labeled and unlabeled data. Reinforcement learning takes a different approach through the use of a feedback mechanism. In reinforcement learning, the learning is accomplished through a process of being rewarded for correct actions or predictions. For example, reinforcement learning is often used in games where the intent is to minimize risk and maximize reward.

A neural network (NN) is a technique of machine learning that is generally effective at extracting critical features from complex data sets and deriving a function or model that expresses those features [2]. The NN utilizes a training data set to first train a model. After the model is trained, the NN can then be applied to new or previously unseen data points and classify the data based on the previously trained model.

A convolutional NN (CNN) is a type of neutral network that is intended to process multidimensional vectors such as images [2]. A CNN can be used for both binary classification and multiclassification tasks where these classifiers differ only in the number of output classes. For example, a data set of animal images can be utilized to train an image classifier. The CNN is provided a vector of pixel values from an image that is accompanied by the correct output class label the vector described (cat, dog, bird, etc.). When an image is provided as an input to the CNN for training, it will attempt to classify the image with an output class. For every training image, the classification is compared with the expected output class label. The weights throughout the network are then iteratively updated through a backpropagation process in an attempt to improve the accuracy of the CNN during training. Each iteration helps form the CNN model which defines the features of training data sets and its ability to correctly classify images.

There are several other types of NNs that have been developed for different problem sets and application types [3]. For example, generative adversarial networks (GANs) have been used to generate a brand-new image content that has similar characteristics to a given training data set [4]. An example of this might be producing a new realistic image of a cat or even a human face.

Autoencoders can be used to reduce the dimensionality of data into a smaller latent space while preserving imperative features. Training an autoencoder includes shrinking input down to a compressed representation and then decompressing the data with the aim of recovering the original data [5]. Once trained, autoencoders have been used to predict missing or corrupted values of images [3].

Manuscript received June 5, 2019; accepted January 8, 2020. Date of current version August 11, 2020. *(Corresponding author: Nikki Pelchat.)*

The authors are with the Software Systems Engineering, University of Regina, Regina, SK S4S 0A2, Canada (e-mail: pelchat.nikki@gmail.com; craig.gelowitz@uregina.ca).

Associate Editor managing this article's review: Mohamed Shehata.
Digital Object Identifier 10.1109/CJECE.2020.2970144

Another common NN is the recurrent NN (RNN). RNNs are used when the sequential context is important. This means the output from a previous iteration may influence the current output [6]. Common application examples of RNNs are for text data and speech data because the order of words is important with respect to the meaning of the sentence.

B. Music Classification

Humans are particularly good at listening to short samples of songs with the ability to distinguish the artist, the song title, and even the genre. Emulating these abilities has been attempted through a number of NN approaches, and they have shown varying levels of success [7]. A popular application example for automatically determining both artist and song title from music data is the mobile device application Shazam. Shazam is primarily known for being able to determine the song title and the artist from only a few seconds of a song. Shazam has also been working on being able to classify different aspects of music data including genre, instruments used, mood description of the composition, and whether a given user might like a given song [8]. Shazam uses a technique they refer to as a song's signature. Shazam defines a song's signature as the large peaks in amplitude taken from the song's spectrogram. "It's like collecting the locations of the highest mountain peaks in a region; instead of (*latitude, longitude, altitude*), we have (*time, frequency, amplitude*) for these prominent peaks" [8]. Tim O'Brien from Shazam puts together an NN which consisted of two fully connected layers with a final classification output layer of genre labels. "This resembles a very 'vanilla' multiclass classifier model" [8]. He achieved test accuracy in the low 90% range. Combining his NN with Sharath Pingula's (another Shazam employee) track-level collaborative filtering features, he was able to slightly improve the model.

Others have also had success in music genre recognition using spectrograms. Costa *et al*. [9] worked on music genre classification using different types of inputs including spectrograms. Using 900 songs from the Latin Music Database, they divided them into 10 music genres equally. From each of the 900 songs, they extracted three 30 s segments from the beginning, middle, and end of the songs. From the spectrograms they mathematically extracted several features where each 30 s spectrogram was represented as ten 28-dimensional feature vectors. This ultimately resulted in 30 vectors for a single music piece. With those vectors as inputs, they trained a NN and then compared their work with Lopes *et al*. [10]. Lopes had previously presented a method to limit the number of training data points to only include the ones that had shown better discrimination with respect to the output class. The final accuracy Lopes achieved was 60% whereas Costa *et al.* saw an improvement of 7% having achieved a 67% recognition rate.

Despois [11] also used spectrograms as input into a CNN. Despois used a music library which consisted of 2000 labeled songs but reduced the number of genres to Hardcore, Dubstep, Electro, Classical, Soundtrack, and Rap. The subgenres were included in the main genre class. He then converted the songs into spectrograms and sliced the spectrograms up into 128 × 128 pixel images which equated to 2.56 s of the song. He used these spectrogram snippets as input into his CNN. The architecture of the CNN consisted of four CNN layers, a fully connected layer, and a softmax function to classify the results into the genre classes specified. The test set accuracy was not provided, however, the reported validation set accuracy was 90%. Validation accuracy is a measure of the NN's accuracy after each epoch using data the NN has not seen before. He also mentioned some future research and tweaks to the code that could be done which might vary the accuracy of the NN. It included changes to the function that divides the data set into training, validation, and test sections, which could potentially bring down the accuracy.

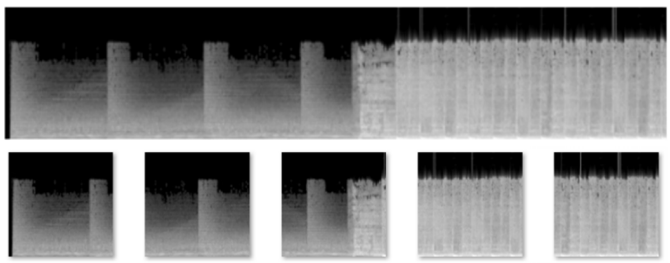

Fig. 1. (Top) 13 s spectrogram of Pop music and (bottom) divided into 2.56 s segments.

II. Implementation Work and Results

A. Implementation

An NN similar to that mentioned by Despois [11] has been implemented in this article with several alterations. The differences include an increase in the number of music genre classes, an increase in the number of training spectrogram slices per genre used for training the NN, as well as the use of different music data sets. The work included a change to the activation function, the use of different digital signal processing window functions when converting the mp3 files to spectrograms, as well as an increase in the number of CNN layers.

In this article, two music libraries were used, both consisting of 1880 songs categorized into ten genres. The preparation of the music data sets included transforming the stereo channels into one mono channel and utilizing the SoX (Sound eXchange) [12] command-line music application utility to convert the music data into a spectrogram. An example of a spectrogram of Pop music is shown at the top of Fig. 1.

The next step in preparing the data sets included slicing up the larger spectrograms into 128 pixel wide PNGs each representing 2.56 s of a given song. A representative example of the 128 × 128 pixel image spectrogram slices is shown at the bottom of Fig. 1.

Both data sets consisted of 1880 different songs, each of 3 min duration. The songs were divided into 2.56 s segment spectrograms to make approximately 132 000 labeled spectrogram snippets. The labeled spectrogram inputs of the data sets were split into 70% training data, 20% validation data, and 10% test data. This transformation is illustrated in Fig. 2.

The NN implemented was a deep CNN using TensorFlow. The final CNN architecture is illustrated in Fig. 3. All the weights were initialized using Xavier initialization and the input vector was a 128 × 128 pixel spectrogram. The first six layers are convolutional layers with a kernel size of 2 × 2 with a stride of two and a max-pooling layer implemented after each successive layer. After the first six layers, there is a fully connected layer where each output of the previous layer

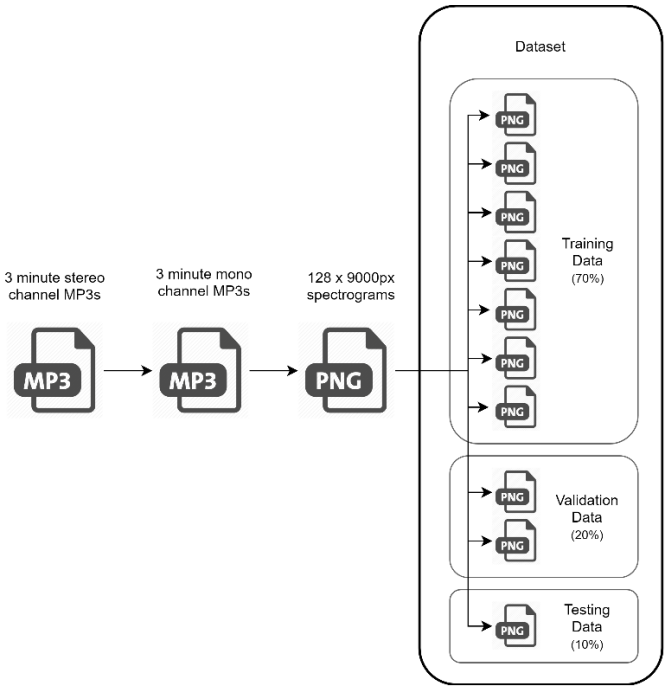

Fig. 2. Data set preprocessing stages.

is fed into each input of the fully connected layer. This technique results in an array of length 4096. A softmax layer is then applied to determine 10 outputs each representing a specific genre.

The activation function being used throughout the network is a rectified linear unit (ReLU). The optimizer used was Adam optimizer. To curb overfitting, a dropout with probability of 0.5 was implemented during training.

B. Experimental Results

The alterations made to an NN similar to that studied by Despois [11], which have been implemented in this article, are covered in this section. When the CNN was initially trained using 27 genres across 995 songs with no changes in architecture from Despois [11], a test accuracy of 47% was observed. It appeared that the implementation of the CNN was overfit as the accuracy for the validation data was 95% versus the test data at only 47%.

Following the initial results, modifications were made to limit the number of genres to 7 (Hip Hop/Rap, Electronic, Rock, Pop, R&B Soul, Alternative, and Country). This was done by consolidating numerous subgenres into the main seven genres. In addition, the data set was almost doubled to 1880 songs, equating to 132 000 total spectrogram slices. With these changes, test accuracy was calculated to be 62%.

Next, the activation function used in the convolutional layers was changed from an exponential linear unit (ELU) to an ReLU resulting in an increase of 5% accuracy. The choice of activation function came from testing out numerous functions including Tanh, ELU, ReLU, and Leaky ReLU. After training the CNN using the ReLU activation function, the test accuracy for this article improved to 67%, which was previously 62%.

Subsequent adjustments involved edits to the width of the slices being used as inputs for the CNN. By default, the input width of spectrograms for the CNN was 128 pixels equating to 2.56 s of a song. Using a slice size of 1.25 s, the accuracy decreased by 2%. It is possible to say that 1.25 s is not long enough to classify a song into a genre. In contrast, increasing the slice size to 5 s decreased accuracy by 3%. By increasing the slice size to 5 s, the number of labeled spectrogram samples in the data set went from 132 000 to 66 000. The decrease in the number of slices appears to affect the accuracy which may indicate that the network requires additional training data to be optimal. To test these hypotheses, more songs are necessary.

The next adjustment involved changes to the digital signal processing (DSP) window function which is used when converting the mp3 file to a spectrogram. The DSP window default was the Hann window, "which has good all-round frequency-resolution and dynamic-range properties" [12]. After assessing the Hann, Bartlett, Hamming, Rectangular, and Kaiser windows it was discovered that the Hamming window provided the most accuracy for this article. Using the Hamming window, a test accuracy of 71% was achieved.

With changes to the activation function and DSP window function, edits to the number of genres were changed again to understand the effects. Broadening the number of genres from 7 to 10 diminished the accuracy by 2%. The decrease in accuracy was minimal compared to the benefits gained in the network's overall objective. The decision was made to continue classifying ten genres (Hip Hop, Rap, Electronic, Dance, Rock, Pop, R&B Soul, Alternative, Classical, and Country) instead of seven. An accuracy of 69% was obtained with these changes.

Modifications to the CNN architecture included adding and removing layers of the CNN. Originally there were four CNN layers, by removing two layers an accuracy of 59% was attained. By adding two more layers to the CNN architecture, a total of six CNN layers, accuracy was improved from 69% to 72%.

To ensure the accuracy of the network persists across different data sets, the same NN architecture was trained on a secondary data set. The second data set used in this article consists of the same number of genres as the initial data set but comprises new songs within those 10 genres. While assembling the new data set, the same number of songs per genre was used as the first data set. This was important because in the initial data set the number of songs per genre was unequal, ranging from 630 songs in Pop to 27 in Electronic. Table I shows the ten genres with the respective number of songs per genre. Using the newly composed data set to training the NN, with the same settings and configurations as before, a test accuracy of 79% was observed.

The next adjustment was to the number of songs per genre for the second data set. As mentioned above, the number of songs in each genre was unequal in the first data set and this was maintained in the second data set for consistency and for comparing test accuracy. The second data set had enough songs in the library to form an equal representation of songs across all genres. This was done by adding or removing songs from certain genres with the goal of having approximately 200 songs in each of the 10 genres. This equates to about

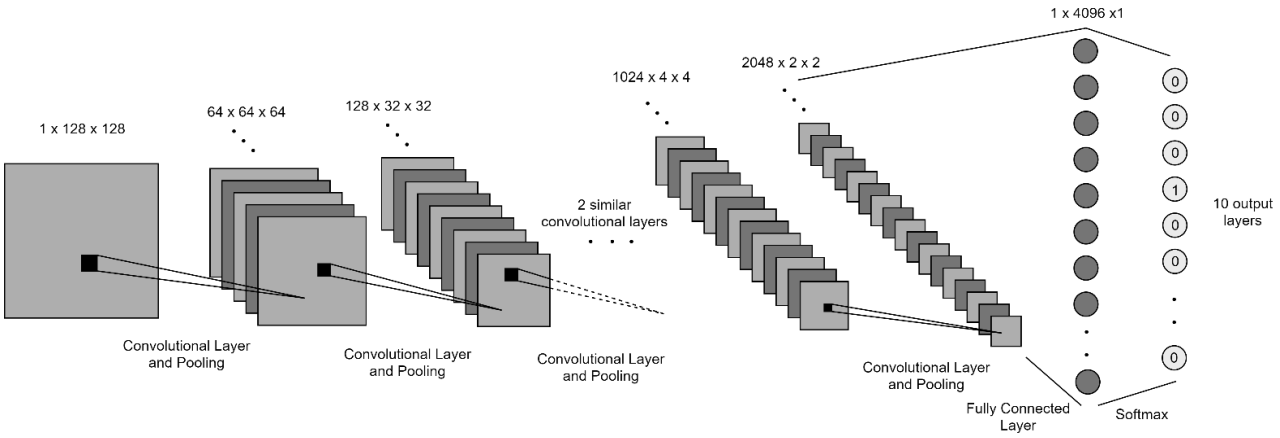

Fig. 3. Implemented CNN architecture.

TABLE I
THE NUMBER OF SONGS PER GENRE WITHIN DATA SET 1

Pop	630 songs	R&B Soul	107 songs
Country	411 songs	Dance	71 songs
Rap	214 songs	Hip Hop	46 songs
Rock	198 songs	Classical	42 songs
Alternative	134 songs	Electronic	27 songs

15 000 spectrogram slices in each genre. After balancing out the number of songs per genre, a test accuracy of 85% was attained. It appears having an equal number of songs in each genre for the training data set is beneficial because the test accuracy tended to increase. This is possibly due to a negative bias being introduced when the number of songs was uneven.

III. CONCLUSION

This article contains an outline of the research and implementation work done in machine learning with respect to musical genre classification. The research implementation involved taking songs, converting them into short-time segments and representing the time segments by their respective spectrogram images. Each of these spectrograms was labeled by music genre and then used as inputs into a CNN. The NN had six convolutional layers followed by a fully connected layer and then a softmax function at the end to calculate the probability of each genre detected and then returned a one-hot array of genre classifications. The results obtained were 85% accurate on the test data.

Future research work with respect to modifications to the algorithms and feature engineering includes changes to the initialization of the weights and experimenting with different filters while converting the mp3 file to a spectrogram. For example, applying a high pass or low pass filter to better understand the effects of each of these modifications. The current implementation has only been trained using grayscale spectrogram images. Future work may include additional features embedded into a colored spectrum.

For testing purposes, the current accuracy is being verified on 2.56 s of the song only. Future work may include different approaches to testing that may include classifying all the slices for a song and selecting the genre class which had the most slices categorized into it. For a 3 min song, all 70 slices would be classified in order to make a final determination. A pooling of individual results may be an effective method for the removal of false-positives.

Other considerations that may apply to this type of work include binary classifiers for determining whether a user might like a given song based on their personal song library or determined listening preferences. RNNs could also be considered using a song library of Musical Instrument Digital Interface (MIDI) data for the consideration of sequential features for song and genre data fed into the network.

REFERENCES

[1] S. Gollapudi, *Practial Machine Learning*. Birmingham, U.K.: Packt, 2016.
[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning.(Report)," *Nature*, vol. 521, no. 7553, p. 436, May 2015, 2015.
[3] J. Shah. (2017). *Neural Networks for Beginners: Popular Types and Applications*. Accessed: Jan. 12, 2019. [Online]. Available: https://blog.statsbot.co/neural-networks-for-beginners-d99f2235efca
[4] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," *CoRR*, vol. abs/1611.09904, Nov. 2016. Accessed: Jan. 12, 2019. [Online]. Available: https://arxiv.org/abs/1611.09904
[5] Y. Chen, "Towards explaining neural networks," M.S. thesis, Fac. Sci., Utrecht Univ., Utrecht, The Netherlands, 2017. Accessed: Jan. 12, 2019. [Online]. Available: https://dspace.library.uu.nl/handle/1874/353056
[6] T. Mikolov *et al.*, "Recurrent neural network based language model," in *Proc. INTERSPEECH*, vol. 2, 2010, p. 4.
[7] Y. M. Costa, L. S. Oliveira, and C. N. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Appl. Soft Comput.*, vol. 52, pp. 28–38, Mar. 2017. Accessed: Dec. 16, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494616306421
[8] T. O'Brien. (2017). *Learning to Understand Music From Shazam*. Accessed: Dec. 19, 2018. [Online]. Available: https://blog.shazam.com/learning-to-understand-music-from-shazam-56a60788b62f
[9] Y. M. G. Costa *et al.*, "Music genre recognition using spectrograms," in *Proc. 18th Int. Conf. Syst., Signals Image Process.*, 2011, pp. 1–4.
[10] M. Lopes, F. Gouyon, A. L. Koerich, and L. E. S. Oliveira, "Selection of training instances for music genre classification," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 4569–4572.
[11] J. Despois. *Finding the Genre of a Song With Deep Learning—A.I. Odyssey Part. 1*. Accessed: Dec. 27, 2018. [Online]. Available: https://hackernoon.com/finding-the-genre-of-a-song-with-deep-learning-da8f59a61194
[12] L. Norskog. (2015). *SoX—Sound Exchange*. [Online]. Available: https://linux.die.net/man/1/rec