MUSIC GENRES CLASSIFICATION USING DEEP NEURAL NETWORK

John Uchal, Haotian Zhou

Department of Mechanical Engineering,
Department of Electrical and Computer Engineering

ABSTRACT

This work investigates the effectiveness of using a deep learning neural network to categorize music based on genre. The goal is to create a classifier that can more accurately classify music compared to a simple SVM model baseline. A VGG-based neural network and Mel-spectrograms, MFCCs, and Chromagrams, processed as 2-D images, were each used to train and evaluate separate models. We also compare the performances of deep learning models against a baseline SVM model, and the performances of 3 different kinds of audio features. Each feature is presented as 2D images, and then the network will give the prediction based on the input features. Experimental results show that CNN-based model with Mel-spectrogram as features has a state-of-the-art performance over our version of Magnatune dataset.

***Index Terms*—** musical information retrieval, convolutional neural network, deep learning, music genres classification

1. INTRODUCTION

Genre classification plays an important role in how people consume music. Not only does genre classification help organize a persons music library, but it is also used in music streaming platforms such as Spotify, Apple Music, or Pandora to recommend new music and artists to users. In the ever growing world-music library, it is becoming increasingly difficult to find new artists/groups that a certain person would enjoy. A classifers ability to analyze a users audio history and make accurate recommendations for new music can greatly improve a users music consuming experience.

In Bahuleyan's study[1], the performances of two different classes of models were explored and tested. The first model uses a deep learning method based on a CNN model to identify the labels of genres of a given audio signal, and this CNN model only uses the spectrogram derived from the input files. The second model uses some other features which is calculated from the audio signals, of which some are time domain features and others are frequency domain features.

Inspired by this study, we want to explore the performance of the deep learning model on mel-spectrogram as well as some other features. This study focuses on a classification model based on a deep learning neural network and comparing its effectiveness to a simple SVM model. The neural network consists of two parts: a convolutional neural network and a feed-forward network. A VGG model was used to improve efficiency and reduce computation cost by replacing the fully-connected layers with layers created in this study. Mel-spectrograms, MFCCs, and Chromagrams were extracted from the raw audio files, presented as 2-D images, and fed into the VGG-based neural network. The SVM model used MFCCs to train and evaluate its effectiveness.

The rest of the paper is organized as follow: Section 2 introduces the methods we use and the architecture of our deep learning model; the process and result of our experiments will be presented in Section 3; Section 4 contains some discussion and our conclusions.

2. CLASSIFICATION METHOD

In this section, details of the data preprocessing work and the description of our proposed deep learning method are provided.

2.1. Data preprocessing

Since the computation capacity of our computers are limited, we have to concatenate the original sound files to 20-second long, and we also normalized the amplitude of all audio files with respect to their highest amplitude.

2.2. Feature Extraction

As stated above, we are using 3 different features for out study: Mel-spectrogram, Mel-Frequency Cepstral Coefficients, and Chromagram. The details of feature extraction process is show as follow:

2.2.1. Mel-spectrogram Generation

The spectrograms we derived is a 2D representation of input signals, having 2 axis, in which x-axis represents time domain (frames) and y-axis represents frequency domain (frequency bins). A colormap is used to quantify the magnitude of each pixels in the spectrogram, which is corresponding to a given frequency bin within a given time frame. In our study, all spectrograms of the audio signals are computed by Short Time Fourier Transform (STFT) based on Mel-frequency instead of normal frequency. The parameters used for generating the power spectrograms are listed below:

- Sampling rate (sr) = 16000
- Frame/Window size (n_fft) = 2048
- Time advance between frames (hop_size) = 512 (resulting in 75% overlap)
- Window Function: Hann Window
- Frequency Scale: Mel
- Number of Mel bins: 216
- Highest Frequency (f_max) = sr/2

2.2.2. *MFCC and Chromagram*

Using the similar way as deriving Mel-spectrogram, we also use the form of 3-channel 2D tensors to represent MFCC and chromagram.

Mel-Frequency Cepstral Coefcients (MFCC) was first introduced by Davis and Mermelstein in the early 1990s. MFCC has been serving as an effective feature for different kinds of audio processing tasks such as speech recognition, sound classification, etc. For our study, in the consideration of balancing efficiency and effectiveness, the Short-Time Fourier Transform (STFT) of the signal is taken with n_fft=1024 and hop size=4096 and a Hann window. Next, the power spectrum is computed, to which a series of triangular Mel filter bank is applied, which mimics the human perception of sound. After that, we take the discrete cosine transform (DCT) of the logarithm of all filterbank energies, and then the MFCC of a given audio signal is obtained . The parameter n_mfccs, which is corresponding to the number of Mel filter banks, was set to 32 in this study.

Chromagram is also a kind of feature that can represent music audio. The entire spectrum is projected onto bins representing different semitones (or chroma) of the musical octave. Regarding the fact that in music notes that are exactly 1 or 2 octave(s) apart are usually perceived as similar notes, obtaining the distribution of chroma even without the absolute frequency can give us useful information about the audio, and may reveal some musical similarity apparent in human perception but not apparent in the original spectra. For our study, the Short-Time Fourier Transform (STFT) of the signal is taken with n_fft=1024 and hop size=2048 and a Hann window, and the number of chroma is set to 32 in order to fit the CNN input requirement.

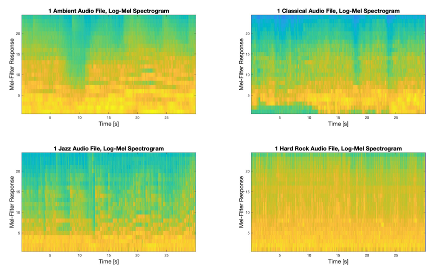


Fig. 1. Mel-spectrograms for 4 genres

2.3. Convolutional Neural Network

By using deep neural network, the task of music genres classification is achievable without the need for hand-crafted features. Convolutional neural networks (CNNs) have been widely used for the task of image classification[2] (Krizhevsky et al., 2012). The 3-channel (RGB) 2D tensor representation of an given image is fed into a CNN which is to be trained to predict the class of the image. In this study, the audio signals can be represented as a spectrogram, which has the form of 2D images and can be treated as such.[3, 4] (Nanni et al., 2016)(Lidy and Schindler, 2016). The purpose of the CNN is to utilize the spectrogram to predict the genres labels. From Figure 1, we can understand that there are some patterns in the spectrograms of the audio signals from different genres. Hence, spectrograms of different kinds of genres can be considered as images of different classes, and serve as input data into CNN, which has been proven effective on image classification tasks. Each block in a CNN consists of the following operations:

- **Convolution**: This step involves convolution with a 2D filter over the input image. We compute an element-wise multiplication between the filter and the overlapping area of the image, then we conduct a summation of each pixel in the area to return a feature value for this area. We may use several such filters , whose values are learned during the training of the neural network in the process of back propagation.
- **Pooling**: This step is to reduce the size of the 2D feature map obtained from the previous convolution step, which is can be viewed as the process of downsampling if we regard the 2D feature map as a kind of signal. For example, when doing max-pooling with 2x2 window size, we only take the maximum value among the 4 elements within the area of the feature map that are covered by the window. Similar to convolution process, we keep sliding the window across the feature map with a defined stride.
- Non-linear Activation: The convolution operation is

linear and in order to make the neural network more effective, non-linear activation function is needed. For this purpose, we need to apply an activation function such as ReLU on each element of the feature map.

In this study, we take advantage of a CNN architecture known as VGG-16, which was the top performing model in the ImageNet Challenge 2014[5](Simonyan and Zisserman,2014). This model consists of 5 convolutional blocks, each consists of convolution, pooling and activation layers, and they are linked in a sequential order. The last convolutional block is followed by a set of fully-connected layers, which map the output of the last convolutional block to a vector that contains the probabilities with respect to each of the possible classes. For our task of music genre classification we use the model architecture with pre-trained weights, and only keep the convolutional blocks. The output of the convolutional blocks is then fed to a set of fully-connected layers in order to predict the genre of the music. The whole architecture of our model is shown as Figure 2.

There are 2 ways of using the pre-trained model:

- **Transfer learning**: Weights in the convolutional blocks are untrainable but the weights in the fully-connected layers (circled by the red box in Figure 2) are able to be trained to fit the correct genre label.
- **Finetuning**: In this setting, weights in the convolutional blocks are trainable, we start with the pre-trained weights, but allow all the weights to be trained during the training process. The model may fit the data better in this way, but the computational cost is significantly higher than transfer learning.

In consideration of both efficiency and effectiveness, as well as the limitation of our devices, we use transfer learning in our study. The final layer of the whole neural network outputs a vector that contains the probabilities (calculated with the softmax activation function) for each class labels. Next, the cross-entropy loss L is computed as follows:

$$L = -\sum_{c=1}^{M} y_{o,c} * ln(p_{o,c}) \tag{1}$$

where M is the number of classes; $y_{o,c}$ is a binary indicator (0 or 1) if class label c is the correct classification for observation o; $p_{o,c}$ represents the models predicted probability observation o is of class c. This loss is used to backpropagate the error, compute the gradients and update the weights within the network. This is an iterative process which continues until the loss converges.

2.4. Implementation Details

As stated previously, we use VGG-16 model as the convolutional blocks. Implementing the whole structure from scratch could be very difficult, so we use *Keras*, a widely used deep learning platform, to implement our model. For the fully-connected layers, we use a 512-unit hidden layer followed by a 256-unit hidden layer, and the last layer is a softmax layer. In order to reduce the effect of over-fitting, we use a L2 regularizer with λ set to 0.01, for the first fully-connected layer and a dropout rate of 0.5 for the first and the second layer. The neural networks are implemented in Python using Keras 2.2.4 with tensorflow-gpu 1.13.1 as backend; an NVIDIA GeForce GTX 1060 with Max-Q Design GPU was used to accelerate the training process. All models were trained for 200 epochs with a batch size of 64, with the ADAM optimizer[] (Kingma and Ba, 2014), and the learning rate is set to 0.0002. The feature extraction process is implemented with a Python open-source library *librosa*[6] The dataset is split into train (90%), test (10%) sets, since we can only train our model on a limited amount of data, we use cross-validation to fully utilize our dataset.

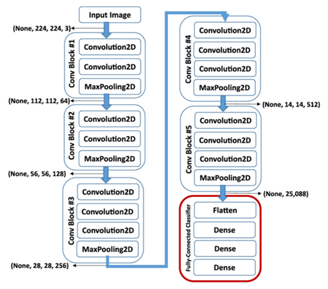


Fig. 2. Schematic of vgg16-based neural network

2.5. Baseline SVM Model

The baseline model used for comparison was a ECOC (error-correcting output codes) model with SVM (support vector machine) learners built in MATLAB. Initially Mel-Spectrograms and MFCCs were planned to both be used to train and evaluate the models, however MATLAB only uses the CPU when training its models and as such computation cost is relatively very high compared to other training systems which use the GPU. To reduce computation time, measures were taken to reduce the size of the training vectors. First, since Mel-Spectrograms and MFCCs contain similar data and MFCCs are more typically used, only MFCCs were used in the final model analysis. To further reduce computation time, only the first 10 seconds of each audio file was used in feature extraction. From there, a spectrogram was created using a Hamming window, a window size of 3200 samples (200 ms), a hop size of 1280 samples (80 ms). 24 logarithmically spaced Mel-filters were to create the Mel-Spectrogram and after the Discrete Cosine Transform of each frame was found to create the MFCC of an audio file, only the first 13 dimensions of the MFCC were kept. From here silent or low-volume frames were removed.

3. EXPERIMENTS

We conducted 4 experiments, 3 for our deep learning models, and 1 for baseline SVM model. All our experiments are conducted on a selection of Magnatune dataset. The selection contains approximately 5500 audio files in 9 genres: Alt Rock, Ambient, Classical, Electro Rock, Electronica, Hard Rock, Jazz, New Age, World.

3.1. Metrics and Result

We use 2 metrics to evaluate the performance our classification task: accuracy and f-score. The comparison of performance of the models is shown in Table 1.

Model	Accuracy	F-score
CNN+Mel-spectrogram	0.90	0.90
CNN+MFCC	0.75	0.74
CNN+Chromagram	0.74	0.72
Baseline SVM	0.32	0.31

Table 1. Comparison of performance of the models on the test set

3.2. Discussion

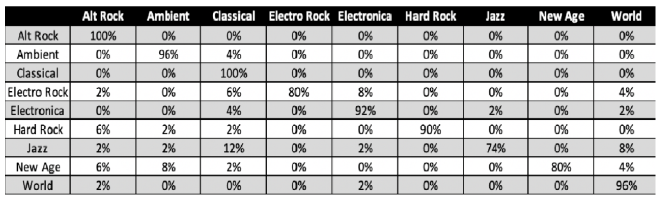
From the result we can see that the CNN+Mel-spectrogram model has the best performance over the dataset, and CNN models have significantly better performance. The baseline SVM model that uses the MFCC as input data performs poorly on the dataset. This shows that CNNs can signicantly improve the scores on such an image classification task.

Among 3 features we used for the experiments, Mel-spectrogram has the best performance, it could be attributed to that Mel-spectrogram contains features that can easily be identified and utilized by CNN structures. Though the MFCC and Chromagram model don't have the same level of performance as Mel-spectrogram, they still have acceptable performance over the dataset, so they can also be used for the classification task, but further improvement is needed.

The confusion matrix for our Mel-spectrogram model is shown as Figure 3, we can notice that several pairs of genres are more likely to be confused with each other: Classical and Jazz, Electro Rock and Electronica, New Age and Ambient. This observation is understandable because those pairs sound similar or have some patterns in common, e.g., similar instrument set, similar pitch distribution.

4. CONCLUSION

We succeeded in designing a classification schema for the music genres classification task. Our CNN-based music genres classification model achieved a state-of-the-art performance over the dataset we use. However, since the real-world music data can be very large and complex, this model still needs to be tested on other datasets if it is to be used for real-world applications. Besides, combination of other audio features and other machine learning methods with this model that may improve the performance still need to be explored in the future.



	Alt Rock	Ambient	Classical	Electro Rock	Electronica	Hard Rock	Jazz	New Age	World
Alt Rock	100%	0%	0%	0%	0%	0%	0%	0%	0%
Ambient	0%	96%	4%	0%	0%	0%	0%	0%	0%
Classical	0%	0%	100%	0%	0%	0%	0%	0%	0%
Electro Rock	2%	0%	6%	80%	8%	0%	0%	0%	4%
Electronica	0%	0%	4%	0%	92%	0%	2%	0%	2%
Hard Rock	6%	2%	2%	0%	0%	90%	0%	0%	0%
Jazz	2%	2%	12%	0%	2%	0%	74%	0%	8%
New Age	6%	8%	2%	0%	0%	0%	0%	80%	4%
World	2%	0%	0%	0%	2%	0%	0%	0%	96%

Fig. 3. Confusion matrix for Mel-spectrogram model

5. REFERENCES

[1] Hareesh Bahuleyan, "Music genre classification using machine learning techniques," *arXiv preprint* arXiv:1804.01149, 2018.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classication with deep convolutional neural networks," In *Advances in neural information processing systems*, pages 10971105, 2012.

[3] Loris Nanni, Yandre M.G. Costa, Alessandra Lumini, Moo Young Kim, and Seung Ryul Baek, "Combining visual and acoustic features for music genre classication," *Expert Systems with Applications*, 45:108117, 2016.

[4] Thomas Lidy and Alexander Schindler, "Parallel convolutional neural networks for music genre and mood classication," *MIREX2016*, 2016.

[5] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint* arXiv:1409.1556, 2014.

[6] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python." *Proceedings of the 14th python in science conference*, pp.18-25, 2015.