WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

MUSIC GENRE CLASSIFICATION

Tejas Dalvi, Suchet Kamble, Saurabh Yadav

Department of computer Engineering, padmabhushan vasantdada patil pratisthan's college of engineering, sion-mumbai

ABSTRACT

We discuss the application of convolutional neural networks and convolutional recurrent neural networks for the task of music genre classification. We focus in the case of a low- computational and data budget where we cannot afford to train with a large dataset. We start using a well-known architecture in the field and we use transfer learning techniques to adapt it to our task. Different strategies for fine-tuning, initializations and optimizers will be discussed to see how to obtain the model that fits better in the music genre classification. Moreover, we introduce a multiframe approach with an average stage in order to analyze in detail almost the full song. It is used at training time to generate more samples and at test time to achieve an overview of the whole song. Finally, we evaluate its performance both in a handmade dataset and in the GTZAN dataset, used in a lot of works, in order to com- pare the performance of our approach with the state of the art

Keywords: spectrograms, neural networks, feature extraction, categorization, music information retrieval

1. INTRODUCTION

Sound is represented via an audio signal, which includes properties such as frequency, decibel, and bandwidth. The amplitude and time of a typical audio signal can be stated mathematically. The computer can interpret and analyse these audio signals because they come in a variety of formats. The mp3 format, WMA (Windows Media Audio), and wav (Waveform Audio File) formats are only a few examples.

Music analysis is based on a song's digital signatures for a variety of elements like as acoustics, danceability, tempo, energy, and so on, in order to discover the types of music that a person would want to listen to.

Music is differentiated by categorized classifications known as genres. Humans are the ones who come up with these genres. A music genre is defined by the features that its members have in common. These features are usually linked to the music's rhythmic structure, instrumentation, and harmonic content.

Genre classification may be quite useful in explaining certain intriguing challenges, such as developing song references, searching down related songs, and identifying societies that will enjoy that particular music. It can also be utilised for survey reasons.

Automatic musical genre classification can help or even replace people in this process, making it a very useful addition to music information retrieval systems. Furthermore, automatic genre classification of music can provide a foundation for the generation and evaluation of features for any type of content-based musical signal analysis.

Because of the rapid growth of the digital entertainment sector, the concept of automatic music genre classification has been highly popular in recent years.

Although categorizing music into genres is arbitrary, there are perceptual characteristics such as instrumentation, rhythmic structure, and texture that can help define a genre. Until now, digitally downloadable music had to be classified by hand. Automatic genre categorization algorithms would thus be a helpful addition to the development of audio information retrieval systems for music.

Problem Statement

Music has a significant impact on people's lives. Music brings individuals together who share similar interests and serves as the glue that holds communities together. Communities can be identified by the type of music they write or listen to. Various societies and groups listen to various types of music. The genre of music is a key quality that distinguishes one type of music from another.

The aim of this project is:

1. Create a machine learning model that categorises music into its appropriate genre.

2. To evaluate the accuracy of this machine learning model to that of previous models and derive the appropriate conclusions.

Objectives

1. Developing a machine learning model that classifies music into genres shows that there exists a solution which automatically classifies music into its genres based on various different features, instead of manually entering the genre.

2. Another objective is to reach a good accuracy so that the model classifies new music into its genre correctly

3. This model should be better than at least a few pre- existing models.

2. DATASET

GTZAN is a dataset created by Tzanetakis et al[[19]]. It is compounded of 1000 music excerpts of 30 seconds dura- tion with 100 examples in each of 10 different music genres: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popu- lar, Reggae, and Rock. The files were collected in 2000-2001 from a variety of sources including personal CDs, radio, mi- crophone recordings, in order to represent a variety of record- ing conditions. And they are all 22050Hz Mono 16-bit audio files in .wav format.

However, regarding our approach it has one important limitation. The duration of each music excerpt (30s) makes that only one frame per song can be extracted and without discarding anything neither at the begging nor in the end of the song. Therefore, although this dataset has been useful to compare the transfer learning performance with other works, it cannot be applied to evaluate the multiframe approach, as in this case more than one frame per song is needed.

In addition, the dataset includes metadata that allows users to experiment without having to deal with feature extraction. The librosa Python library, version 0.5.0, was able to extract all of these features. Each feature set (excluding zero-crossing rate) is calculated on 100-sample windows separated by 512-sample hops. We have divided the dataset into train subset, 100 songs per genre.

3. METHODOLOGY

The details of data pre-processing are described in this section, followed by a description of the recommended strategy to this classification challenge.

Deep Neural Networks

We can classify music genres without using hand-crafted attributes thanks to deep learning techniques. Convolutional neural networks (CNNs) are an excellent choice for picture classification. A CNN is given the 3-channel (R-G-B) matrix of a picture, which it uses to train itself on those images. The sound wave is represented as a spectrogram in this study, which can be treated as an image. The CNN's objective is to predict the genre label using the spectrogram (one of eight classes).

Spectrogram Generation

A spectrogram is a 2D representation of a signal, having time on the x-axis and frequency on the y-axis. In this study, each audio signal was converted into a MEL spectrogram (having MEL frequency bins on the y-axis). The parameters used to generate the power spectrogram using STFT are listed below:

- Sampling rate (sr) = 22050
- Window size (n_fft) = 2048
- Hop length = 512
- X_axis: time
- Y_axis: MEL
- Highest Frequency (f_max) = 8000

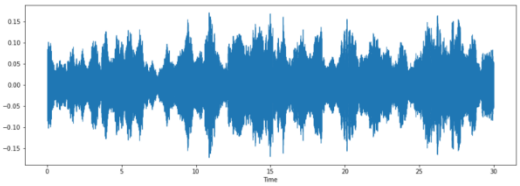


Fig- 1: Sample spectrograms for 1 audio track from eachmusic genre

The spectrograms from each genre are shown in Figure 1.

Convolutional Neural Networks

Figure 2 show that the spectrograms of audio signals belonging to various classes have some distinguishing patterns. As a result, spectrograms can be thought of as "pictures" and fed into a CNN.

Feed Forward Network

A CNN is a feed-forward network, which means that input samples are fed into the network and transformed into an output; in supervised learning, the output is a label, which is a name given to the input. They connect raw data to categories, recognizing patterns that would indicate that an input image should be classified "folk" or "experimental," for example. A feed forward network is trained on tagged images until it can guess their categories with the least amount of error. The network uses the trained set of parameters (or weights, collectively known as a model) to categories input it has never seen before. A trained feed forward network can be subjected to any random collection of photos, and how it classifies the first photograph will not necessarily affect how it classifies the second. Seeing a spectrogram of a folk song does not cause the internet to see a spectrogram of an experimental music. A feed forward network, in other words, has no concept of time order and just regards the current example it has been exposed to as input. Feed forward networks have amnesia about their recent history, recalling only the early times of their training nostalgically.

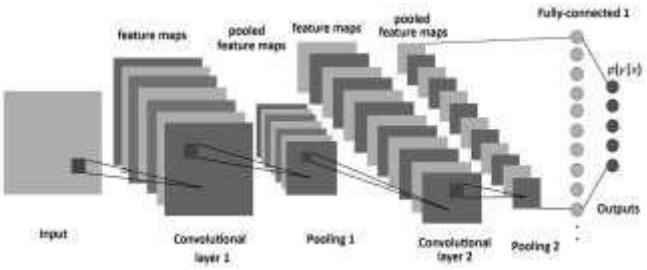


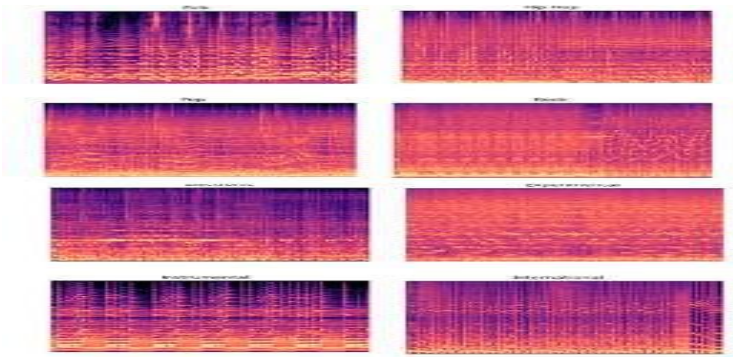
Fig -2: CNN Framework

Operations Of CNN

Each block in a CNN consists of the following operations:

- Convolution: This stage entails moving a matrix filter (of say 3x3) across the input image, which has dimensions of image width x image height. The filter is applied to the image matrix first, and then an element-wise multiplication between the filter and the overlapping area of the image is computed, followed by a summation to get a feature value.

- Pooling: The dimension of the feature map acquired from the convolution step is reduced using this method. We only keep the element with the highest value among the four elements of the feature map that are covered in this window by max pooling with a 2x2 window size. With



a pre-defined stride, we move this window across the feature map.

- Non-linear Activation: The convolution procedure is linear, and we need to introduce some non-linearity to make the neural network more powerful. On each element of the feature map, we can use an activation function like Rectifier Linear Unit (ReLU) for this.

The model is made up of three convolutional blocks (conv base), a flatten layer that converts a 2D matrix to a 1D array, and a fully connected layer that outputs the likelihood that a given image belongs to each of the available classes.

The class probabilities for each of the eight potential class labels are output by the neural network's last layer (using the softmax activation function).

The following is how you calculate the cross-entropy loss:

where M is the number of classes; yo,c is a binary indicator whose value is 1 if observation o belongs to class c and 0 otherwise; po,c is the model's predicted probability that observation o belongs to class c; and po,c is the model's anticipated probability that observation o belongs to class c. This loss is used to backprogram the error, compute the gradients, and update the network's weights. This iterative approach is repeated until the loss reaches a minimal.

Convolutional Recurrent Neural Network

We train a Convolutional Recurrent Neural Network, which is a hybrid of convolutional and recurrent neural networks, to assess the performance improvement that CNNs may produce.

Recurrent Neural Networks

Recurrent nets are a form of artificial neural network that recognises patterns in data sequences like as text, genomes, music, video, or numerical time series data from sensors, stock exchanges, and government agencies. These algorithms have a temporal component since they consider time and sequence. Even photos, which can be broken into a number of patches and handled as a sequence, can be used with RNNs.

Recurrent networks differ from feedforward networks in that they have a feedback loop connected to previous judgments, consuming their own outputs as input moment after moment. Recurrent networks are frequently described as having memory. The addition of memory to neural networks serves a purpose: the sequence itself contains information, which recurrent nets employ to complete tasks that feedforward networks cannot.

Long-term memory networks, or "LSTMs," are a type of RNN that can learn long-term dependencies. Hochreiter & Schmidhuber were the ones who presented them (1997).

CRNN Model Details

The CNN-RNN model, also known as the CRNN model, consists of three one-dimensional convolution layers, an RNN's LSTM layer, and a fully linked dense layer, which serves as the output layer. To keep the evaluation and comparison fair, the batch size utilised in this model is 32 and the number of epochs is kept at 30. ReLU was used as the activation function. Dropout of 0.2 has been included in the hidden layers to avoid overfitting of the data.

CNN-RNN Parallel Model

This model runs the CNN and RNN models in tandem while maintaining the same metrics and regularisation factors as the prior models. The goal is to examine and compare the performance measures of a simple CNN model with these robust, complicated models.

Implementation Details

The spectrogram images are 150 × 150 pixels in size. A 512-unit hidden layer is implemented for the feed-forward network connected to the conv base. In neural networks, over-fitting is a typical problem. To avoid this, we implemented the following strategy:

Dropout [21]: During training, we randomly turn off part of the neurons (set their weights to zero) as a regularisation method. We forecast the final output with a different combination of neurons in each iteration, effectively randomising the training cycles. A 0.2 dropout rate is employed, which means that a given weight is set to zero with a probability of 0.2 during an iteration.

The dataset is divided into three groups: training (80%), validation (10%), and testing (10%). (10 percent). For all of the comparisons, the same split is used..

Tensorflow is used to implement the neural networks in Python. With a batch size of 64, all models were trained for 30 epochs. The ADAM optimizer was used to optimise these neural networks. One iteration of the full training set is referred to as an epoch.

Feature Extraction

This section summarises the features extracted from previous models that were compared to the proposed model. Time domain and frequency domain features are two types of features that can be grouped together. Librosa, a Python library, was used to extract features

Time Domain Features

These are the features which were extracted from the raw audio signal.

- Central moments: This consists of the mean, standard deviation, skewness and kurtosis of the amplitude of the signal.
- Zero Crossing Rate (ZCR): The signal's sign shifts from positive to negative at this moment. The amount of zero-crossings present in each frame is determined when the complete 30-second signal is broken into smaller frames. As typical features, the average and standard deviation of the ZCR over all frames were chosen.
- Root Mean Square Energy (RMSE): The energy signal in a signal is calculated as
- RMSE is calculated frame by frame and then the average and standard deviation across all frames is taken.
- Tempo: The tempo of a piece of music refers to how quick or slow it is. The tempo is measured in BPM (beats per minute) (BPM). We use the Tempo's aggregate mean, which varies from time to time.

Frequency Domain Features

The Fourier Transform is used to convert the audio signal into the frequency domain. The following characteristics are then extracted.

- Mel-Frequency Cepstral Coefficients (MFCC): Davis and Mermelstein introduced MFCCs in the early 1990s, and they've shown to be quite beneficial for applications like speech recognition..
- Chroma Features: This is a vector that represents the signal's total energy in each of the 12 pitch classes. (C, C#, D, D#, E, F, F#, G, G#, A, A#, B, C, C#, D#, D#, D#, D#, D#, D#, D#, D#, D#, D#, D#, D The mean and standard deviation are calculated using the sum of the chroma vectors.
- Spectral Centroid: This is the frequency at which the majority of the energy is centred. It's a magnitude-weighted frequency, which is determined as follows:

  where S(k) is the spectral magnitude of frequency bin k and f(k) is the frequency corresponding to bin k.
- Spectral Contrast: Each frame is separated into a number of frequency bands that are pre-determined. The spectral contrast is calculated as the difference between the maximum and minimum magnitudes within each frequency range.
- Spectral Roll-off: This feature corresponds to the value of frequency below which 85% of the total energy in the spectrum lies.

The mean and standard deviation of the values taken across frames is regarded the representative final feature that is provided to the model for each of the spectral features discussed above.

Classifier

This section provides a brief overview of the machine learning classifier adopted in this study.

- **Logistic Regression (LR)**: For binary classification tasks, this linear classifier is commonly employed. The LR is implemented as a one-vs-rest approach for this multi-class classification assignment. That is, 8 binary classifiers are trained separately. During testing, the predicted class is picked from among the 8 classifiers with the highest probability.
- **Simple Artificial Neural Network (ANN)**: An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks. Because a neural network alters - or learns, in a sense - based on the input and output, the information that goes through the network modifies the structure of the ANN. ANNs are nonlinear statistical data modelling tools that are used to model or find patterns in complex interactions between inputs and outputs. This model takes a csv file containing handcrafted characteristics taken from audio clips using the librosa package and outputs a function that is similar to the Logistic Regression methodology mentioned above.

4. EVALUATION

Metrics

In order to evaluate the performance of the models, the following metric will be used.

Accuracy: Refers to the percentage of test samples that are correctly classified. This statistic assesses how close the model's prediction is to the actual data.

5. RESULTS AND DISCUSSION

The model to evaluate the performance of our multiframe ap- proach has been trained with the train partition of our hand- made dataset and evaluated using the test partition. The genre prediction has been made using the recurrent layers fine-tuned CRNN model with a total accuracy of 77.89%. Furthermore, the confusion matrix between the real genre and the predicted genre has been built for two different scenarios: using the predicted tag of each frame of the test database, and using the predicted tag of the average score obtained for each song of the test database. Thus allowing an evaluation of the average stage improvement..

Therefore, we can conclude that the resulting model works well in almost all of the 10 genres. The weaker results have been obtained in disco, metal and reggae, which is rea- sonable as they are the less distinguished genres. Metal can be confused by rock in some songs, the same with reggae and hiphop, and finally disco is a genre that can be understood as a mix of other genres

With the exception of metal, all genres have seen an increase in diagonal parts after implementing the average stage. Despite this, the average accuracy, calculated as the mean of the diagonal elements, has grown from 77.89 to 82 percent. As a result, we can deduce that using the average stage outperforms using merely one frame every song.

6. CONCLUSION

We explore the application of CNN and CRNN for the task of music genre classification focusing in the case of a low- computational and data budget.

The results have shown that this kind of networks need large quantities of data to be trained from scratch. In the sce- nario of having a small dataset and a task to perform, trans- fer learning can be used to fine-tune models that have been trained on large datasets and for other different purposes.

In the ex- periments, a homemade dataset compounded by songs longer than our frame duration has been used. These songs belong to 10 different genres and the experiments have revealed that the average stage achieves better results in 9 of these 10 gen- res and a higher total accuracy. Therefore, using the average stage we are able to remove the non-representative frames de- pendence.

ACKNOWLEDGEMENT

We would like to give our heartfelt thanks to our guides Prof. Manish Gangawane for helping us choose the domain of our project and being constantly supportive & encouraging us throughout the journey of this project.

REFERENCES

[1] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, Xavier Bresson. FMA: A Dataset For Music Analysis. Sound; Information Retrieval. arXiv:1612.01840v3, 2017.

[2] Tom LH Li, Antoni B Chan, and A Chun. Automatic musical pattern feature extraction using convolutional neural network. In Proc. Int. Conf. Data Mining and Applications, 2010.

[3] Hareesh Bahuleyan, Music Genre Classification using Machine Learning Techniques, University of Waterloo, 2018

[4] Loris Nanni, Yandre MG Costa, Alessandra Lumini,Moo Young Kim, and Seung Ryul Baek. Combining visual and acoustic features for music genre classification. Expert Systems with Applications 45:108–117, 2016.

[5] Thomas Lidy and Alexander Schindler. Parallel convolutional neural networks for music genre and mood classification. MIREX2016, 2016.

[6] Chathuranga, Y. M. ., & Jayaratne, K. L. Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches. GSTF International Journal of Computing, 3(2), 2013.

[7] Matthew D Zeiler and Rob Fergus, "Visualizing and un- derstanding convolutional networks," in European Con- ference on Computer Vision. Springer, 2014, pp. 818– 833.

[8] Sander Dieleman and Benjamin Schrauwen, "Multi- scale approaches to music audio feature learning," in 14th International Society for Music Information Re- trieval Conference (ISMIR-2013). Pontif´ıcia Universi- dade Cato´lica do Parana´, 2013, pp. 116–121.

[9] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An end-to-end neural network for polyphonic piano music transcription," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 5, pp. 927–939, 2016

[10] Keunwoo Choi, George Fazekas, and Mark Sandler, "Explaining deep convolutional neural networks on mu- sic classification," arXiv preprint arXiv:1607.02444, 2016.